

# Web Site Adaptation: Recommendation and Automatic Generation of Navigation Menus \*

Vera Hollink, Maarten van Someren and Stephan ten Hagen\*

Dept. of Social Science Informatics, University of Amsterdam

{maarten,vhollink}@swi.psy.uva.nl

\*Faculty of Science, University of Amsterdam

stephanh@science.uva.nl

Keywords: Adaptive navigation support, adaptive web sites, recommender systems

## Abstract

In recommender systems the interests of users are typically represented as unordered sets of pages. However, on web sites where the pages are not independent of each other, the order in which the pages are visited is important. A recommender should not only recommend the right pages, but also recommend them in the right order. A widely used form of recommending is the context dependent navigation menu. In this short paper we compare content and usage based methods to build navigation menus from scratch. In our experiments the navigation traces of users clearly showed stages, which indicates that users prefer to view pages in a specific order. Content based methods appear to be adequate for creating menus that group of pages with similar topics, but do not provide an ordering. Usage based methods are necessary to recommend pages that match a user's current navigation stage.

## 1 Introduction

In the past few decades web sites have evolved from small electronic leaflets to continually changing highly complex information systems. This development has urged the need to provide the users of web sites with adaptive navigation assistance to prevent them from drowning in the available information. Recommender systems can provide such assistance by dynamically adding links to web pages. A recommender estimates which information a user wants and adds links that can reduce the effort needed to reach this information.

Many recommender systems make clusters of pages with similar topics in such a way that users who are interested in some of the pages from a cluster have a high probability of also being interested in the other pages from the same cluster. Some systems, including [Lieberman, 1997] and [Schwab and Pohl, 1999] group pages with a similar content while others group pages which are often visited in the same session (e.g. [Perkowitz and Etzioni, 2000], [Mobasher *et al.*, 2002]). The clusters do not indicate the order in which the user likes to view the pages. In cases where a user knows which pages he wants to see, the order in which the pages are shown is probably not a major issue. In this situation recommendations function as short-

cuts, which allow the user to reach his goal without passing through series of less interesting pages.

Users do not always know exactly what they are looking for and what the site has to offer, especially when they are visiting the site for the first time. Sometimes users search the site to find a solution to a problem, without knowing which solutions are available. In these cases pages and links include navigation information. They do not provide an answer to the user's question but rather help to articulate the question or to tell her where to look further. In this case the order in which the pages are viewed is relevant: the users first need to get information about the available solutions before they can benefit from pages describing a specific solution. A recommender should be aware of this and offer links which match the current stage of a user's search. Another way to look at this, is that the relations between the pages impose a partial ordering over all pages. The goal of a recommender system is to find out which sequence of pages within this ordering answers the user's information need. In this short paper we present the outlines of a method to generate recommendations which match both the topic and the stage of a user's search. The methods are evaluated in the context of navigation menus, which can be viewed as a special kind of recommendations.

This paper is structured as follows. Section 2 presents navigation menus as recommendations and introduces our approach. In section 3 the setting we use for evaluation is described. Section 4 contains the details of the evaluated methods and section 5 provides preliminary results of our experiments. In the last section we discuss the results and present our plans for the future.

## 2 Menu generation

Nowadays many sites help the users in their navigation by the use of hierarchical menus. These navigation menus are present on every page of the site and contain a structured list of links to (some of) the other pages of the site. In most cases the menus are context dependent in that they only show the specialized pages that are closely related to the current page, while links to more distant pages are hidden under one or more header links. The visible links in these context dependent menus can be viewed as recommendations.

As a first step in building recommender systems we construct navigation menus from scratch. We represent relations between pages by a reward function which associates the transition from each page to each other page with a real number. The number indicates how useful (a link to) the second page is after the first one has been seen. The menus are generated by placing after each link to a page  $p_1$  links

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

to those pages that have a high value for coming after  $p_1$ . If the created menus are able to decrease the search effort of the users, then the functions are apparently able to represent the important relations between pages. Therefore these functions can also be used to generate useful recommendations in other recommendation settings.

We distinguished two types of reward functions. Content based functions assign a high value to two pages if they are similar in content. The overlap in words between pages is used to determine how similar pages are. A disadvantage of these functions is that they can *only* find a relation between two pages if their contents are not too dissimilar, which is not necessarily the case for a page describing a problem and a page describing a solution. Instead of the content of the pages usage based functions use the information stored in log files. They group pages which are often viewed consecutively in the same session. In theory usage based methods can find more subtle relations between pages, but one has to be careful, as the discovered navigation patterns can be a result of the initial structure of the site rather than the interests of the users as we showed in [Someren van *et al.*, 2003].

Content based reward functions are usually symmetric: the value of seeing a page  $p_1$  after a page  $p_2$  is equal to the value of seeing  $p_2$  after  $p_1$ . Symmetric functions can specify groups of pages with a similar topic, but do not provide a preferred viewing order within these groups. Since many people can navigate from page  $p_1$  to  $p_2$  without going in the other direction, usage based functions are typically not symmetric and can specify a partial order.

### 3 Experimental setting

We evaluate our methods on the navigation menu of the SeniorGezond site<sup>1</sup>. The SeniorGezond site is a Dutch site developed by domain specialists from the Netherlands Organization for Applied Scientific Research TNO. It contains information for elderly people about the prevention of falling accidents. The pages of the SeniorGezond site are divided into three types: pages describing a problematic situation, pages describing a solution and pages describing a specific product or service. The navigation menu of the SeniorGezond site consists of two parts. The lower part of the menu structures all solutions, products and services per topic. People who are familiar with the contents of the site can use this part of the menu to navigate quickly to the specific information they need. The upper part of the menu associates each problem with a possible solution and each solution with some products and services. This 'process' part is meant to help people who have a problem, but do not know the solutions for their problem. Figure 1 shows a screenshot of the SeniorGezond site with the menu open on one of the items in the upper part of the menu.

The menus that are generated by the reward functions are evaluated by comparing them with the original menu on the SeniorGezond site, viewing menus that are more similar to the original menu as better ones. This is an intuitive metric, as the site and its menu are created by experts in the field, so that we can safely assume that the original menu is a good one. However, it is possible that there are more ways to structure the pages that are equally useful to the users. Since menus based on these alternative structures do not resemble the original menu, they receive an unfair low score. A more reliable method for measuring the use-



Figure 1: A screenshot of the SeniorGezond site with the menu on the left side of the page. The menu in this figure is open on one of the items in the process part of the menu.

fulness of a structure is using pre- and post-tests. First we have users search for certain information on the site while the menu is replaced by a menu with no structure at all (a flat list of links) and we measure the time and number of clicks they need to find the information. Then we replace the menu by a generated menu and measure again the effort needed to find the information. The menu reducing the effort most has the best structure.

### 4 Content based and usage based functions

For the content based functions the headers and the menu were removed from the pages and the words in the remaining texts were used to compare the pages. We compute the similarity in content between two texts by counting the number of overlapping words and weighting each overlapping word with its discriminative power. In the fields of (*web*) *document clustering* and *information retrieval* much time has been investigated in defining good word weighting schemes. We test the use of the TF.IDF term weighting scheme [Salton and McGill, 1983] and stop word removal and stemming for normalization. We use Hierarchical Agglomerative Clustering (HAC) for the actual clustering of the pages, as this technique is very simple and intuitive and has led to good results (e.g. [Dubey and Jain, 1988]).

To test the usage based functions we created user logs. We removed all links from the text and replaced the context dependent menu by a static structure which listed all links alphabetically. In this way we could be sure that the observed patterns in the navigation were not a result of the initial structure of the menu. We asked 29 subjects to perform ten tasks on the modified site. The subjects were all students who had never seen the SeniorGezond site before, so that their choices were solely based on the link anchors in the new menu and not on any knowledge about the structure of the original site. In each task the subjects had to play the role of an elderly person in a problematic situation who searched the SeniorGezond site for a solution to his or her problem. The sequences of pages that the subjects viewed were recorded in a log file and used to create the usage based reward functions.

On the basis of the logs the pages are clustered per stage and per topic. We detect the relevant stages of the search process and assign each page to one or more stages by determining in which parts of the sessions the page is visited most often. Initially pages are assigned to stages on the basis of their average positions in all visits to the site. We improve the initial classification through bootstrapping. If

<sup>1</sup><http://www.SeniorGezond.nl/>

a page which is initially assigned to stage  $s_1$  occurs in the log files more times between two pages from stage  $s_2$  than between pages from  $s_1$ , its stage is changed to stage  $s_2$ . The pages are grouped per topic by computing how often they occur in the same session. In the navigation menu we place under each link to a page  $p_1$  a link to those pages that (1) have the same topic as  $p_1$  and (2) are from the stage following the stage of  $p_1$ .

## 5 Results

We compared the generated menus with the original menu by computing the overlap between the clusters in the original menu and the generated menu and comparing the number of clusters in both menus. The menus generated by the content based functions were to a high degree similar to the topic based part of the original menu. The process part of the menu on the other hand could not be reproduced. Pages were split on a high level into the categories problems, solutions and products and services, so that hardly any connections were found between problems and the corresponding solutions and products and services.

These poor results are probably caused by the fact that on the SeniorGezond site the references to the associated pages are stated very briefly. The solution pages contained a long description of the general solution while the list of links to specific products was very compact. In the other direction, the product pages only described the product and did not contain a reference to the solutions. For a human reader this means he does not have to read the same information twice, but for a content based recommender it makes the cross relations almost impossible to find.

Preliminary results of experiments with the usage based functions show that users indeed tend to follow the pattern *problem pages - solution pages - product and service pages*. On the basis of the logs from the experiment the stage discovery algorithm assigned 99% of the pages to the right stage. This result suggests that navigation stages play an important role in user navigation. A navigation structure or recommender which makes use of the discovered stages can potentially outperform a system that uses only topic information. We are currently testing this hypothesis.

## 6 Discussion

In this short paper we discussed the need of a richer representation of interests than unordered sets of pages for automatic generation of navigation menus and recommender systems. Besides pure content pages, many web sites contain pages with a navigation function. These pages give information about the contents of the site and the relations between the contents of the pages. Users who do not know exactly what they are looking for benefit from first viewing one or more of these navigation pages, before going to more specific content. To provide optimal assistance to a user a recommender system should be aware of this and recommend pages in the right order. Sometimes navigation pages need to be recommended before more specific pages are recommended, although the specific pages are the ones that will eventually satisfy the user's information need.

We compared two types of methods to automatically generate navigation menus for web sites, one based on content and one based on usage. In an experiment we demonstrated that the content based methods could generate useful menus which reflect the topics of pages, but do not provide an order in which the pages should be viewed. Although we did not investigate all possible ways to make

use of content, we believe this is an inherent limitation of content based methods; to generate a preferred ordering usage based approaches will be necessary. We presented a usage based method to detect stages in navigation from the order of the pages in log files and to incorporate these stages in navigation menus and recommender systems. The underlying assumption is that users will most likely navigate through a site in a particular order. Preliminary results show that navigation stages can be restored from log files with very high precision. More empirical evaluation is needed to find out to what extent the discovered stages can benefit the recommendation process.

We are currently implementing and evaluating the use of navigation stages in automatic menu generation. We are planning to perform a series of post-tests on the SeniorGezond site which will show whether the use of stages in menus decreases the effort needed to find information. The reward functions that give the most promising results will be incorporated in a recommender system which will work on the full online version on the SeniorGezond site. This allows us to evaluate the methods in a real life situation with more noisy data. However these results will still be valid only for the SeniorGezond site; in later stages we will have to see whether they are transferable to other domains.

The methods described here either use the content or the usage of the pages. However, studies have shown that a combination of content and usage information can improve recommending (e.g. [Joachims *et al.*, 1997]). Moreover, in hybrid systems content can be used to solve the cold start problem and recommend newly added pages. In the future we will investigate the combination of stages deduced from usage and topics deduced from content.

## References

- [Dubes and Jain, 1988] R.C. Dubes and A.K. Jain. *Algorithms for clustering data*. Prentice Hall, 1988.
- [Joachims *et al.*, 1997] T. Joachims, D. Freitag, and T. Mitchell. Webwatcher: A tour guide for the world wide web. In *Proceedings of The 15th International Joint Conference on Artificial Intelligence*, Nagoya, Japan, 1997.
- [Lieberman, 1997] H. Lieberman. Autonomous interface agents. In *Proceedings of The ACM Conference on Computers and Human Interaction*, Atlanta, USA, 1997.
- [Mobasher *et al.*, 2002] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Discovery and evaluation of aggregate usage profiles for web personalisation. *Data Mining and Knowledge Discovery*, 6:61–82, 2002.
- [Perkowitz and Etzioni, 2000] M. Perkowitz and O. Etzioni. Towards adaptive web sites: Conceptual framework and case study. *Artificial Intelligence*, 118:245–275, 2000.
- [Salton and McGill, 1983] G. Salton and M.J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, 1983.
- [Schwab and Pohl, 1999] I. Schwab and W. Pohl. Learning user profiles from positive examples. In *Proceedings of the ACAI'99 Workshop on Machine learning in user modeling*, Chania, Greece, 1999.
- [Someren van *et al.*, 2003] M. Someren van, S. Hagen ten, and V. Hollink. Greedy recommending is not always optimal. In *Proceedings of the 1st European Web Mining Forum*, Cavtat-Dubrovnik, Croatia, 2003.