Context-based Navigational Support in Hypermedia

Sebastian Stober and Andreas Nürnberger

Institut für Wissens- und Sprachverarbeitung,
Fakultät für Informatik,
Otto-von-Guericke-Universität Magdeburg, D-39106 Magdeburg, Germany
{stober,nuernb}@iws.cs.uni-magdeburg.de

Abstract. In this paper, we present the system "DAWN" (direction anticipation in web navigation) that helps users to navigate through the world wide web. Firstly, the purpose of such a system and the approach taken are motivated. We then point out relations to other approaches, describe the system and outline the underlying prediction model. Evaluation on real world data gave promising results.

1   Introduction

Navigating through hypermedia can be a hard task, especially if the resource is as dynamic as the world wide web that is steadily growing and constantly changing. Users that browse the web often need to choose between multiple options on how to continue navigation. Depending on how well they choose the hyperlinks they follow, it will take them less or more time to finally get to the information they desire. Whilst some users might find this task quite easy, others may get lost, especially if they are facing unfamiliar web content. Although there is most likely no general rule on how to most efficiently navigate to the desired information, there may be certain navigational patterns that can be used as heuristics. Users may unconsciously develop such patterns. A system that watches users navigating the web could learn the users' navigational patterns. During a browsing session, the system could perform a lookahead crawl in the background and suggest the hyperlinks that best match the previously learned patterns.

   This paper introduces a prototype of such a system, named DAWN (direction anticipation in web navigation). In the following sections, differences to related work are pointed out and a brief system overview is given. In Sect. 4 we present some promising results of a first evaluation of the system with real-world data. Finally, we summarize our work and give an outlook on future developments.

2   Related Work

The general idea to support users browsing the world wide web is not new. Systems that accomplish this task by suggesting hyperlinks or web pages are e.g.

discussed in [1–5]. The system presented here has much in common with these systems: Like Webmate [3], Broadway [4] or Personal Webwatcher [5] it uses an HTTP-proxy to log user actions and to manipulate the requested web pages. Documents are, according to common practice, represented as term vectors with TF/iDF-weights. Letizia [1] was one of the first systems that used background lookahead crawling. However, Letizia was designed as a plug-in for the Netscape Navigator 3 and heavily relied on its API whereas the proxy-architecture chosen for DAWN allows the users to use their browsers of choice. Moreover, bandwidth and computational expensive tasks can be performed on the server which reduces the burden on the client machine. In contrast to DAWN, which uses a combination of navigational patterns and document similarities, Webmate, Personal Webwatcher and Letizia are purely content-based systems that rely on document-similarities to decide which web pages are interesting. Broadway relies on case-based reasoning to recommend web pages using a special similarity measure on ordered sequences of past accessed documents. For this, a special similarity measure on ordered sequences of past accessed documents had been developed, combining temporal constraints with similarities of URLs and page content represented by page title, HTML headers and keywords. Webwatcher [2] uses a collaboration-based approach by asking a user about the desired information and then suggesting links that users with similar information needs have followed previously. Furthermore, Webwatcher is a server-side application that is restricted to one specific website, whereas DAWN works on a client-side proxy and therefore has no such local restriction. DAWN stores the navigational patterns in a Markov Model. Such models have been, e.g., successfully used for prediction of HTTP-requests to optimize web-caches [6]. Recently, they have been proposed to model user navigational behavior in the context of adaptive websites [7–9] and web-usage mining [10]. However, these are solely server-side applications that are inherently locally confined. To our knowledge, there have not been any client-side systems that use Markov Models so far.

3 System Overview

An overview of the system is shown in Fig. 1. All HTTP-requests made by the user during a browsing session are recorded in a database. This information
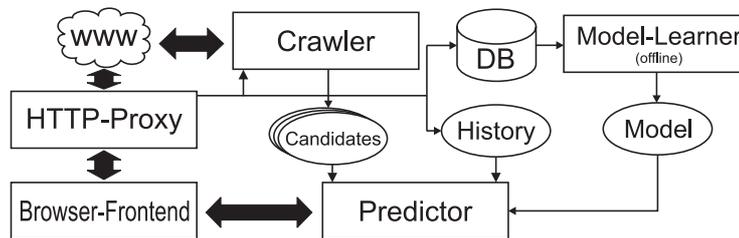

Fig. 1. DAWN: System Overview

is used by the model-learner that generates a model of navigational patterns using an algorithm that has been derived from the one presented by Borges and Levene in [10]. This algorithm has been originally designed to represent a collection of user navigation sessions of a website. It iteratively constructs an nth-order Markov Model by making use of the state cloning concept where a state is only duplicated if the nth-order probabilities diverge significantly from the corresponding 1st-order probabilities. Apart from several small modifications we introduced an additional clustering step prior to model induction. In this step similar pages are grouped into clusters according to standard TFiDF-similarity. Each cluster corresponds to a single context represented as a weighted term vector. This drastically reduces the size of the model's state space but more importantly it combines different browsing paths (consisting of web pages) into an abstract navigational pattern (consisting of contexts). The additional abstraction level accomplished by the introduction of the preliminary clustering step makes it possible to detach the original approach from its server-side application, i.e. the prediction model can be applied to arbitrary web sites.

It is possible to learn a separate model for each user as well as a global one. Based on an nth-order Markov Model candidate pages can be assessed given a user's current history of the last n accessed web pages. Mapping the history onto the model by identifying similar navigational paths, a probability distribution for the next state (i.e. a cluster of web pages) is derived. Having mapped the candidate pages onto the model's states in an analogous manner, the probability distribution can be used to rank the candidate pages. Eventually, title, URL and automatically generated thumbnails of the three most probable candidates are displayed in the browser-frontend shown in Fig. 2. The thumbnails allow a general visual impression of the page layout and provide visual information that can help the user in addition to the usual ranking and content information to assess a list of candidate pages collected by a lookahead crawler.

4 Evaluation

Aim of the system presented here is to support users to navigate through the world wide web. Consequently, the usefulness of the system can only be truly assessed by a user study. This however involves considerable effort and costs. We therefore decided to do a first evaluation of the prediction model on web server log files assuming that every link that a user followed led to a page satisfying the user's information need. Obviously, this assumption may not hold in all cases. Furthermore, predicting a link that a user did not follow does not necessarily implicate poor performance as recommending the link still might have been useful. Thus, the accuracy of the predictions can only be interpreted as an indicator for the system's usefulness.

As the content of the accessed web pages is required in the clustering step, only web server log files that did not contain outdated URLs could be utilized for the evaluation. We therefore used anonymized log files containing requests on web pages hosted by the University of Magdeburg recorded during 6 consecutive

4
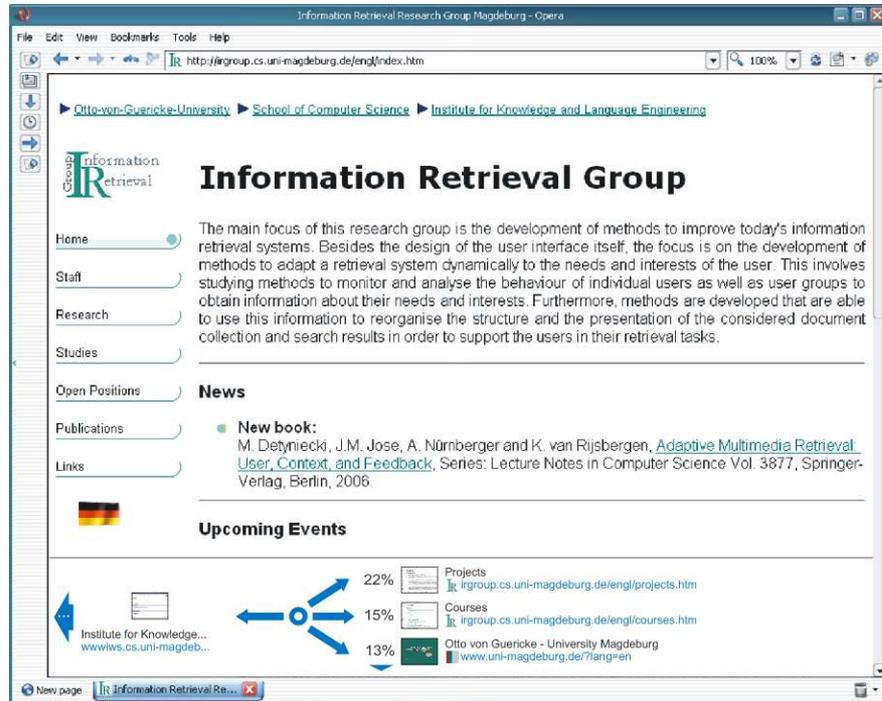
Fig. 2. Browser-Frontend. Suggestions for possible next links in the users navigation path are recommended at the bottom of a web page.

days. These logs were filtered as follows: Any requests with a response status code other than 200 (OK) and 304 (not modified) were removed as well as requests of URLs with non-HTML content. Additionally, several possible denial of service attacks on specific URLs were detected and removed. Afterwards, sessions were identified as, e.g., described in [11]. The prediction model was learned from the data of the first 5 days. For the evaluation of the model the data of the 6th day was used. Table 1, left shows the number of sessions, requests and unique URLs in the data.

The requested web pages in the training data were clustered into 250 clusters resulting in an average cluster size of about 100. From the 250 clusters and the sessions extracted from the training data, a 2nd-order Markov Model was induced. This model was used to estimate the probability of all outgoing links for each web page in the test sessions. Table 1, right shows the results. In about 30% of all test cases, the candidate that had actually been chosen by the user was amongst the 3 highest ranked candidates. In these cases it would have been displayed in the browser-frontend and could have helped the user.

	sessions	requests	URLs	rank	1st	2nd	3rd	1st–3rd	in 1st third
train	58314	237098	25771	absolute	6788	2113	3328	12229	22716
test	13437	60461	5657	relative	16%	5%	8%	30%	56%
total	71751	297559	27877						

Table 1. Left: Number of sessions (sequences of pages accessed by a user), requests (page accesses) and unique URLs in the data used for training and evaluation. Right: Evaluation results. The number of candidates (number of different outbound links in a page) ranged from 1 to 607 with a mean of 10.77.

5 Conclusions and Future Work

In this paper, we have presented a prototype system that provides navigation support for users browsing the world wide web. It combines ranking, content and visual information to make it easier for users to assess the relevance of suggested hyperlinks. A first evaluation of the prediction accuracy on real-world data has shown promising results. However, the evaluation results should only be interpreted as an indicator for the system's usefulness. As future work we plan an evaluation of the system in a user study.

References

1. Lieberman, H.: Letizia: An Agent That Assists Web Browsing. IJCAI, 1995.
2. Joachims, T., Freitag, D., Mitchell, T.: WebWatcher: A Tour Guide for the World Wide Web. In: IJCAI, 1997.
3. Chen, L., Sycara, K.: WebMate: A Personal Agent for Browsing and Searching. In: Proc. 2nd Intl. Conf. on Auton. Agents and Multi Agent Sys., AGENTS '98, 1998.
4. Jaczynski, M., Trousse, B.: Broadway, A Case-Based Browsing Advisor for the Web. In: ECDL '98: Proc. of the 2nd Europ. Conf. on Research and Adv. Technology for Digital Libraries, 1998.
5. Mladenic, D.: Machine learning used by Personal WebWatcher. In: Proc. of ACAI-99 Workshop on Machine Learning and Intelligent Agents, 1999.
6. Sarukkai, R.: Link Prediction and Path Analysis using Markov Chains. Computer Networks, Vol. 33, pp. 337–386, 2000.
7. Anderson, C., Domingos, P. Weld, D.: Relational Markov Models and their Application to adaptive Web Navigation. In: Proc. 8th ACM SIGKDD Intl. Conf. on Knowl. Discovery and Data Mining, 2004.
8. Zhu, J., Hong, J., Hughes, J.: Using Markov Chains for Link Prediction in Adaptive Web Sites. In: Soft-Ware 2002: Comp. in an Imperfect World: 1st Intl. Conf., 2002.
9. Cadez, I., Heckerman, D., Meek, C., Smyth D., White, S.: Model-Based Clustering and Visualization of Navigation Patterns on a Web Site. Data Min. Knowl. Discov., Vol. 7, pp. 399–424, 2003.
10. Borges, J., Levene, M.: A Clustering-Based Approach for Modelling User Navigation with Increased Accuracy. In: Proc. of the 2nd Intl. Workshop on Knowl. Discovery from Data Streams (IWKDDS) & PKDD, 2005.
11. Cooley, R., Mobasher, B., Srivastava, J.: Data Preparation for Mining World Wide Web Browsing Patterns. Knowl. and Information Sys., Vol. 1, No. 1, pp. 5–32, 1999.