Mining User Access Logs to Optimize Navigational Structure of Adaptive Web Sites

Željko Eremić*, Dragica Radosav*, Branko Markoski*
*University of Novi Sad, Technical faculty “Mihajlo Pupin”, Zrenjanin, Serbia
zeljko.eremic@gmail.com, radosav@tfzr.uns.ac.rs, markoni@uns.ac.rs

***Abstract*—Web sites may contain numerous documents. Using some web techniques, it’s possible to analyze users’ data about using resources, contents of those documents and structure of web sites. Adaptive web sites automatically change their structure and representation based on visitor’s behavior. Shortcutting is an approach that enables connecting two documents which has never been connected before. Most of existing approaches enables connecting the first and the last document in user’s navigation path, not considering the possibility that some of the documents within the navigation path might contain useful information for reaching intended document. These documents, which are positioned within user’s navigation path, are called wayposts, and they may contain useful information that can help users to get to the specific “target” document. The goal of this paperwork is to discuss about all the possibilities of identifying those waypost documents in users’ navigation paths and to propose an optimization of navigation structure of a web site based on users’ navigation paths, initial and target documents.**

I. Introduction

Inspiration for this approach comes from some earlier works like [1] where visitors have left "footprints" while navigating through website. Idea of adaptive web sites is presented in [2] and [3]. Adding and removing hyperlinks in adaptive web sites, as a way to change web site's organization, is described in [4]. Those works are used as a foundation for approach described in this article.

Recommended input data for this kind of optimization should be easily and simply obtained in some standardized data format, so that the processing would be easier and done on larger data set. The best given solution might be log files that are generated by web servers. These documents keep records that show the activities that have been occurred on the Web site. An example of a log file data is given:

„127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326“, [5].

According to [6] there are several steps performed before data preprocessing. These steps, that are performed sequentially, are shown in Fig. 1.

Merging is a step where the contents of various relevant log files are merged into one log file. Afterwards, sorting by access time is being performed. Data cleaning is an activity of removing all unnecessary data from log files. Data, that are being removed, have been collected by web robots, spiders and crawlers. Data can also be obtained by requests for images on web pages. Only http status codes are accepted which indicate to a success from 200 to 299, and request methods like GET and POST.

“Session identification is the process of segmenting the access log of each user into individual access sessions”, [7].

In cases where there is no user’s registration, the user is most often associated with the IP address in the record, although one should consider that different users can assess at various times from the same IP address. User’s session is a series of requests that the user makes within a specified time, and one user can also have multiple session.

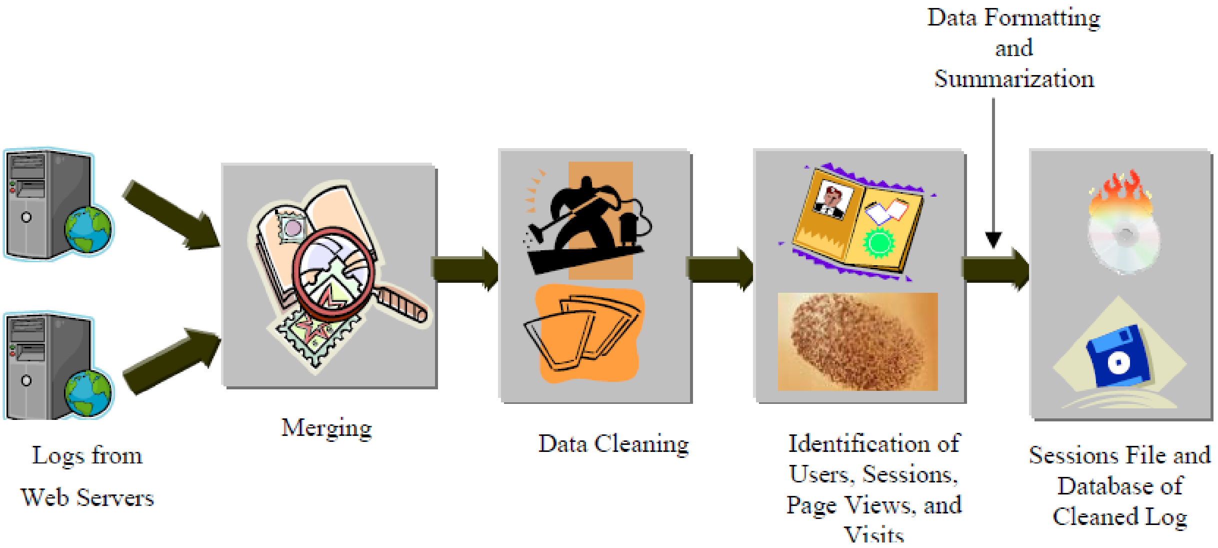

Data Formatting and Summarization
Logs from Web Servers
Merging
Data Cleaning
Identification of Users, Sessions, Page Views, and Visits
Sessions File and Database of Cleaned Log

Figure 1. Overview of Data Preprocessing according to [7].

978-1-4244-9280-0/10/$26.00 ©2010 IEEE

“The page view identification step determines which page file requests are part of the same page view and what content was served. This is necessary to provide meaningful results in the pattern analysis phase. If this step is not performed, the discovered patterns can be dominated by page files that make up a single popular page view. Page views are identified by using the time of the request. For requests made at the same time, only the first request (as ordered in the log file) is retained and the following ones are discarded.”, [7]. Data formatting and summarization represent the last step of data preprocessing where the file, that contains data about users and their sessions, is transformed into relational data base format which can be used later on for various statistical calculations.

II. TARGET DOCUMENTS AND PATHS

Users’ activities in a web site are motivated by the need to reach specific content. The result may be either successful – the user accessed specific document or not successful – the user failed to access specific document, gave up on further search and ended the session. Target documents are those documents that user has been successfully reached depending on the requirements. Series of documents which have been visited make user’s path.

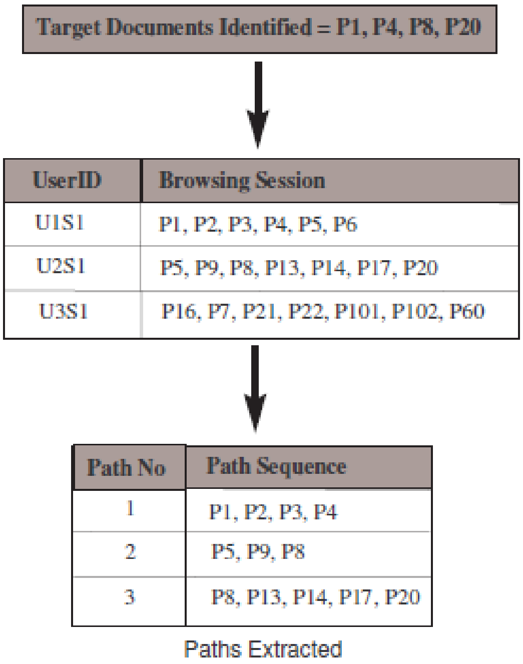
Approach described at [8], and used as a base for this paperwork separates waypost documents as often visited documents from user’s path. Navigational paths that are studied, begin with initial document and they represent series of previously visited documents with target document as an ultimate document. Some of the methods for identifying target documents are End Document Method, mentioned in [8], and Browsing Time Method, mentioned in [8].

“The sequence of documents visited along this journey indicate the navigational route taken by a user to reach his intended target document.”[8].

Users’ sessions are transformed in series which describe navigational paths by using previously obtained set of target documents. A session can contain no target documents, one or more target documents and therefore it’s possible to have more navigational paths in a session. An example of extracting paths from users’ session is given in Fig. 2.

III. CLUSTERING

“Clustering is a data mining technique used to group a set of items that share similar characteristic together. It is commonly used to discover clusters of document with similar content or identify groups of users with similar browsing behaviors in a website.”, [8]. For determination of similarities between two navigational paths different approaches can be used. One of the approaches is to determine the number of common documents between two paths in relation to total number of documents in those two paths.



Target Documents Identified = P1, P4, P8, P20

UserID	Browsing Session
U1S1	P1, P2, P3, P4, P5, P6
U2S1	P5, P9, P8, P13, P14, P17, P20
U3S1	P16, P7, P21, P22, P101, P102, P60

Path No	Path Sequence
1	P1, P2, P3, P4
2	P5, P9, P8
3	P8, P13, P14, P17, P20

Paths Extracted

Figure 2. Extracting Paths from Users’ Sessions Based on A Set of Target Documents, [8].

This approach is used in [8]. In this paperwork it is recommended to use Levenshtein distance algorithm.

“The Levenshtein algorithm (also called Edit-Distance) calculates the least number of edit operations that are necessary to modify one string to obtain another string.”, [9].

It is possible to map different documents in different character, and the two navigation paths can be represented by two strings, over which to apply the algorithm for the Levenshtein distance. In the example where two series with four documents are given, and the second and third document took a different sequence in the two series, the advantage of Levensthein algorithm, that considers the order of documents in the series, not just their presence or absence, could be clearly seen.

IV. IDENTIFICATION OF WAYPOSTS

Selection of initial document affects navigational path but it was observed that there are certain documents which are often accessed before target document has been accessed. If there are some documents which are often accessed in various paths (more often than in some other documents) before target document has been accessed, then those documents can be considered as some kind of road sign and they are candidates for a status of a waypost document.

„Providing a link (i.e.,shortcut) between these potential wayposts could assist users by reducing the number of clicks they have to make while browsing, pointing them in the right direction towards a specific target document.“, [8].

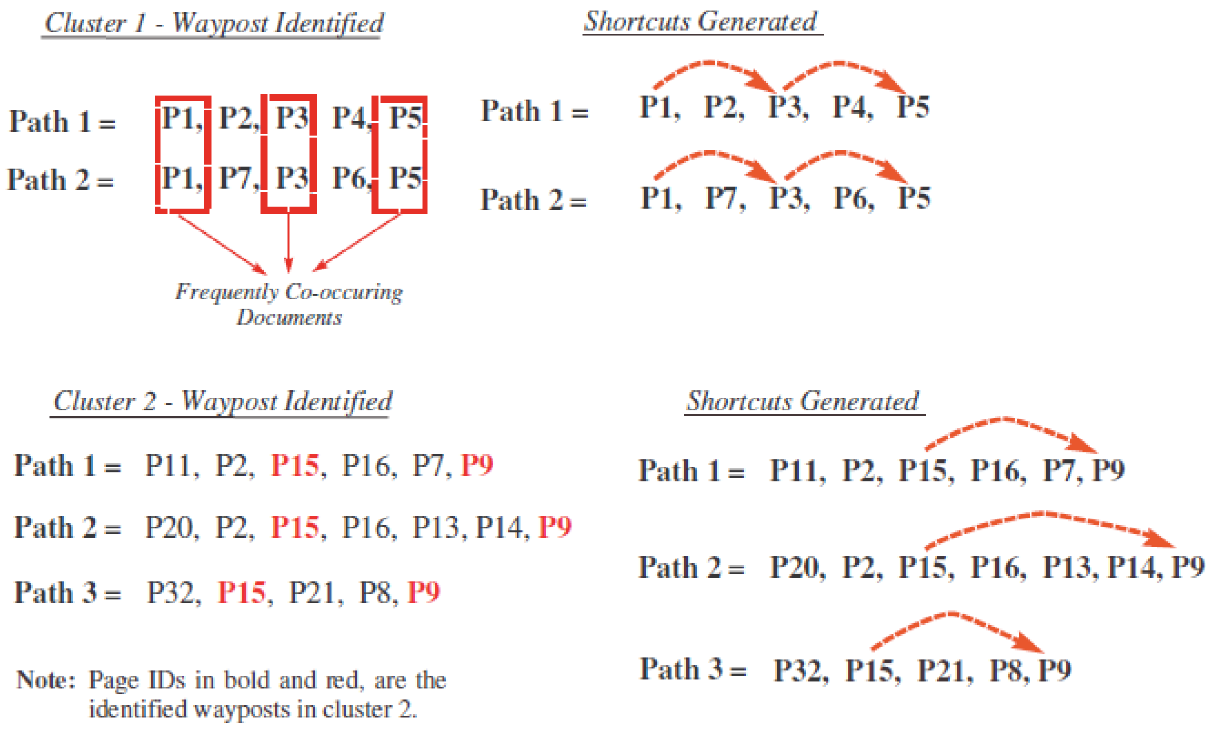

Cluster 1 - Waypost Identified
Path 1 = P1, P2, P3 P4, P5
Path 2 = P1, P7, P3 P6, P5
Frequently Co-occuring Documents
Shortcuts Generated
Path 1 = P1, P2, P3, P4, P5
Path 2 = P1, P7, P3, P6, P5
Cluster 2 - Waypost Identified
Path 1 = P11, P2, P15, P16, P7, P9
Path 2 = P20, P2, P15, P16, P13, P14, P9
Path 3 = P32, P15, P21, P8, P9
Note: Page IDs in bold and red, are the identified wayposts in cluster 2.
Shortcuts Generated
Path 1 = P11, P2, P15, P16, P7, P9
Path 2 = P20, P2, P15, P16, P13, P14, P9
Path 3 = P32, P15, P21, P8, P9

Figure 3. Identifying Wayposts and Generating Shortcuts

In order to be able to separate waypost documents from clusters that have similar paths, a cluster must contain at least two paths, and that there is at least one document different from the target document that appears in each of the cluster path. One example is shown in Fig.3.

V. Overcoming the Disadvantages and Limitations of Previous Approaches

Step forward in the area of waypost document carries with it certain potentials when it comes to an optimization of a Web site navigation. In order to maximize these potentials, certain changes should be established. One of the disadvantages, that has been pointed out, is a neglection of a sequence access to documents in the path. However, the condition, that in each path within the cluster must exist specific document in order to be named waypost, may be too strict for achieving results. An example would be a cluster with 100 path where a specific document appears in 99 paths and cannot be named a waypost only because it doesn't appear in one path which could be a potential loss of useful results. This paperwork promotes some more flexible approach where an element has the status to some extent instead of discrete evaluation where an element has a certain status or not. The boundaries of acceptance could be set to some initial value, but would be modified if such a need prior to practice. One of the aggravating factors in [8] is that it's never been completely tested in practice.

Fig. 4. shows a simple example of the optimization path. The example describes six paths from P1 to P6, where each path represents a series of visited documents.

For the purpose of illustration various documents are represented by different letters. The initial document for 5 paths is a document A, and one initial document is K. Target document is in all cases, G. "Waypost" documents that are common for paths are C and E, but it should be noted that the order of their appearance is different in some paths. For example, in the path P1 sequence of access to documents is A-B-C-D-E-F-G, and in the path P5 sequence of access to documents is A-S-E-T-C-U-G. It has been noticed that the order of appearance of waypost elements is not the same. According to the scheme, there are six shortcuts in this example that can be generated.

For example, the shortcut S [A-C] is generated because it has been noticed that there are three paths (P1, P2 and P4) between documents A and C. These three paths have led from A to some other documents (B,H and O) but eventually ended in C, so there is a presumption that the visitors needed to visit C after visiting A and that there is a room to suggest a shortcut between A and C – which now only exists as a proposal and if it proves as promising to be added to the document A as a suggested link in a later analysis. It should also be noted that some of the shortcuts reduce more paths than others. In this example, it can be seen that shortcuts S[C-E] and S[E-G] perform shortening of the four paths, shortcut S [A-C] perform shortening of the three paths, and the remaining two shortcuts perform shortening of only two paths. The shortcut that performs optimization of larger number of paths is definitely a better candidate for the application, compared to the one that performs the optimization of a small number of paths.

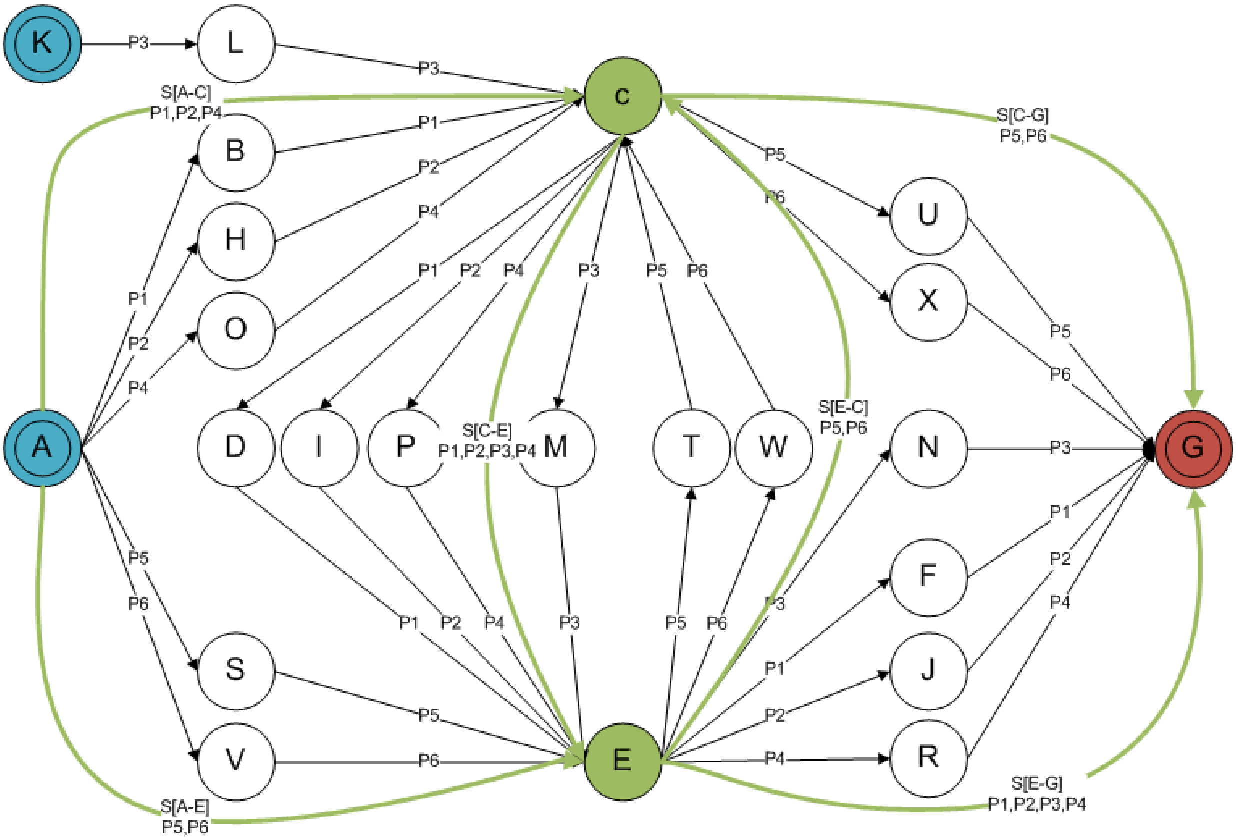


Figure 4. An example of optimization based on six navigation path

The number of documents that would be skipped by using the proposed shortcuts would also be a good parameter influencing the decision on the applicability of the proposed shortcut. Certainly, it is desirable to have a shortcut that significantly reduces the number of visited documents, and thereby saving time visitors and preventing them from committing a mistake and increasing the satisfaction in using the system.

The example in Fig. 5. contains four paths with identified wayposts C,D,G. Here is evident a more flexible approach for promoting waypost elements – the condition is that at least two paths pass through waypost element. Three candidates for the shortcut have been noticed while S[A-C] reduces three paths, and the others reduce two paths. On the other hand, it should be noticed that the shortcut S[G-E] although optimizes only two paths, significantly reduces the number of clicks, especially for the path P3.

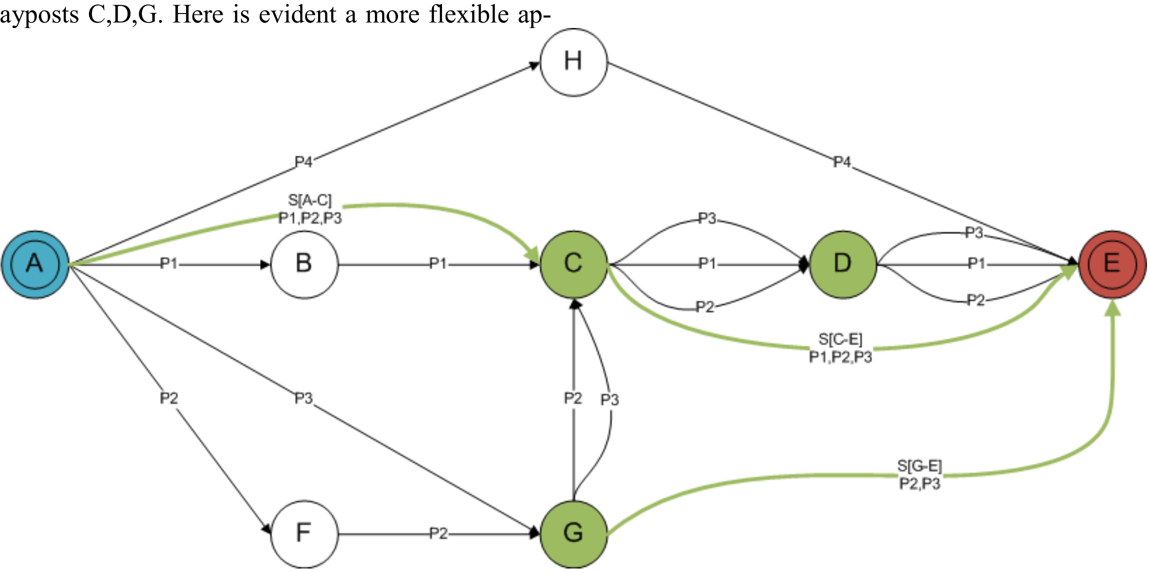


Figure 5. An example of optimization of navigation based on four paths

Number of paths that a shortcut optimizes compared to the total number of paths in the cluster and the number of documents that are skipped by using the shortcut, represent relevant information for a making a decision on acceptance of certain shortcut and about navigation system of a web site.

VI. Conclusion

Web mining techniques can have a useful role in helping visitors to Web sites in a faster way to reach the desired content, while reducing the probability of leaving the system, or wandering. In practice, the most commonly used approaches for the time being are direct connecting of documents with their potential target documents.

There is also an approach that considers the possibility that the documents contained within the path carry useful information and that they may affect the determination of access to a specific target document. In that case, it is advisable to eliminate the drawbacks of existing methods, which could bring significant improvement with the application of new ideas. This paperwork represents a step in that direction.

References

[1] A. Wexelblat and P. Maes, “Footprints History-Rich Tools for Information Foraging”, ACM Press, pp. 270—277, 1999.

[2] M. Perkowitz and O. Etzioni, "Adaptive Web Sites: an AI Challenge", In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, pp. 16-23, 1997.

[3] M. Perkowitz and O. Etzioni, "Adaptive Web sites", Communications of the ACM, Vol.43, No. 8, pp. 152-158, August, 2000.

[4] J. Lee and W. Shiu, “An adaptive website system to improve efficiency with web mining techniques”, Advanced Engineering Informatics , Vol 18, pp. 129-142, 2004.

[5] http://en.wikipedia.org/wiki/Common_Log_Format.

[6] R. Kohavi, and R. Perekh, “Ten supplementary analyses to improve e-commerce web sites”, Proceedings of the Fifth WEBKDD workshop, 2003.

[7] G. Raju, and P. Satyanarayana, “Knowledge Discovery from Web Usage Data: Complete Preprocessing Methodology”, IJCSNS International Journal of Computer Science and Network Security, Vol. 8, 2008.

[8] G. Bathumalai, “Self adapting websites mining user access logs”, The Robert Gordon University, 2008.

[9] http://www.levenshtein.net/index.html.