# Separating Interleaved HTTP Sessions Using a Stochastic Model

Marko Poženel, Viljan Mahnič and Matjaž Kukar
Faculty of Computer and Information Science
University of Ljubljana
Tržaška cesta 25, SI-1000 Ljubljana, Slovenia
E-mail: {marko.pozenel, viljan.mahnic, matjaz.kukar} @fri.uni-lj.si
Corresponding author: matjaz.kukar@fri.uni-lj.si

**Keywords:** Markov model, clickstream analysis, Web user behavior

**Received:** October 30, 2009

*We describe a novel method for interleaved HTTP session reconstruction based on first order Markov model. Interleaved session is generated by a user who is concurrently browsing a web site in two or more web sessions (browser windows). In order to assure data quality for subsequent phases in analyzing user's browsing behavior, such sessions need to be separated in advance. We propose a separating process based on trained first order Markov chains. We develop a testing method based on various measures of reconstructed sessions similarity to original ones. We evaluate the developed method on two real world clickstream data sources: a web shop and a university student records information system. Preliminary results show that the proposed method performs well.*

*Povzetek: V članku predstavljamo metodo za razpletanje prepletenih HTTP sej s pomočjo markovskega modela.*

## 1 Introduction

In the past decades World Wide Web (WWW) has become one of the main sources of information. It has enabled unprecedented exchange of data between different parties. Companies need web sites to reach customers and sell their products, institutions furnish information about their services, individuals can effectively access various services over Internet. With the growing number of web pages and documents, web sites are coping with stronger competition. It is difficult to attract new customers and retain the existing ones. Under such circumstances only the web sites that understand the needs of their customers will prevail. Analysing users' behavior has become an important part of web page data analysis. *Clickstream* data represent the main data source for the analysis of user behavior (11). A sequence of clicks that a user makes while browsing through a website is called a clickstream. Analysis of web data such as clickstreams entails certain problems with availability and quality of data (7).

Data about behaviour of web site visitors have become one of the most important sources of information in most web-aware companies. They play an important part in daily transactions and important business decisions. It is essential to get reliable data analyses, which require both appropriate methods and data. The quality of the the patterns discovered in data analysis depends on the quality of the data on which data mining is performed. A *user session* is represented by one visit of a user to a web site. For better web usage mining results we need reliable sessions. Clickstream data from a normal website are noisy, page events are often not explicitly linked to page requests. The preprocessing phase is therefore prone to errors. Although many methods for sessions reconstruction have been devised (1; 13), reliable session reconstruction still remains a challenge.

Especially really interested and capable users often browse the same web site with multiple browser windows opened. In each web browser they perform actions to complete a certain task. Typically, users switch between browsing tasks so that they work on a task only for a certain time period. Even if only one user is currently active, we actually have concurrent sessions, each for one web browser window (i.e. task). In a web server log file all concurrent sessions will be seen as a single long session. We call such sessions *interleaved sessions*. They cannot be easily separated without some kind of context help. Such sessions have negative effect on data quality so we have to deal with the issue. We have three choices: (i) neglect the problem, (ii) simply abandon such sessions, (iii) try to separate them. The first choice is bad for data quality since such sessions can affect web usage analysis results. If we abandon such sessions we also abandon useful knowledge about web site usage. Such sessions are usually generated by advanced users whose behaviour colud be potentially extremly valuable to us. Therefore we decided to develop a method for separating interleaved sessions.

We present a novel approach for session separation using a trained first-order Markov model to facilitate session separation. To the very best of our knowledge, the Markov approach has not been used for this purpose before. Actually, the interleaved session problem has been largely neglected

in Web mining, with the only exception being Viermetz et al. (14) who use an entirely different approach based on building a clicktree. This clicktree contains all possible paths a user could have taken through a website map. While their approach is dedicated to better understanding of actual user behavior, our approach is focused on separation process. Based on training first-order Markov model on validated (clean) sessions, our approach is very effective in deinterleaving process (with linear complexity). We introduce a special purpose methodology for evaluation of separation process, evaluate our method on clickstreams from different sources, and present preliminary results.

# 2 Methods

## 2.1 Clickstream

In order to attract more visitors to our web site we have to know who our visitors are, what they do on our site, and what they would like to be changed. A great aid in achieving this goal is clickstream data. Clickstream is a sequence of clicks or pages visited as a visitor explores a particular Web site. Clickstream data are often large, inadequately structured, and show incomplete picture of users' activity. For example, server side log data do not involve browser and e.g., network caching ('Back' browser actions or requesting pages in intermediate server's cache) (7).

Clickstream data needs to be gathered, preprocessed and cleaned prior to the analysis. This step depends on the type and the quality of data. Work done in this phase affects the quality of results of further analyses.

The basic form of clickstream data from a Web server is stateless – no session identifier is logged. This is the consequence of the fact that the HTTP protocol is stateless. Each line in the log file shows an isolated resource retrieval event, but does not provide a link to other events in a user session. Since we are interested in all user actions in a certain period of time, we have to gather all individual events in a user session. The process is called *sessionization*. Without some context help it is hard or impossible to reliably identify complete user session. Berendt et al. (1) report that these sessionization tools are based on heuristic rules and assumptions about the site's usage and are therefore prone to errors.

## 2.2 Discrete Markov models for clickstream analysis

Markov chain is defined as follows. We have *a set of states* $S = \{s_1, s_2, ..., s_N\}$, where $N$ denotes the number of states. The process starts in one of the states and moves forward from one state to another at regularly spaced discrete times. For example, the chain is currently in the state $s_i$ and it moves next to $s_j$ with the *transition probability* $p_{ij}$. The starting state is defined by a probability distribution. We denote the steps in which the process changes states as $t = 1, 2, ...n$ and the state at time $t$ as $q_t$. Associated with each state is a set of transition probabilities $p_{ij}$, where

$$p_{ij} = P(s_i \to s_j) = P(q_t = s_j | q_{t-1} = s_i) \quad (1)$$

that is, given the present state, the future and the past states are independant. This paper focuses on time-homogenous Markov chains, in which

$$\forall t : P(q_{t+1} = s_i | q_t = s_j) = P(q_t = s_i | q_{t-1} = s_j) \quad (2)$$

for all $t$, meaning that the transition probabilities do not change with time. We restrict our discussion to Markov chains defined on a finite state-space. The probability of transition between states in a single step can be written as *transition probability matrix* $T$:

$$T = \begin{bmatrix} p_{11} & \cdots & p_{1N} \\ \vdots & \ddots & \vdots \\ p_{N1} & \cdots & p_{NN} \end{bmatrix}, \ \sum_j p_{ij} = 1 \quad (3)$$

The final parameter of a Markov chain is the *starting state*, which can either be a predefined fixed state or can be chosen from a probability distribution on a set of states given in the form of a probability vector $\pi$,

$$\pi = (\pi_1, \pi_2, \pi_3, \ldots, \pi_N) \quad (4)$$

where $\pi_i$ denotes the probability that state $s_i$ is initial and $N$ denotes number of states.

$$\pi_i = P(q_1 = s_i), 1 \leq i \leq N \quad (5)$$

Given a sequence of states $(q_1, q_2, \ldots, q_k)$ we can calculate the probability of the sequence by multiplying the probability of the initial state $P(q_1)$ with the probability of transitions to the successive state as follows:

$$P(q_1, q_2, \ldots, q_k) = P(q_1) \cdot \prod_{i=2}^{k} P(q_{i-1} \to q_i) \quad (6)$$

In the first-order Markov chain the next step depends only on current state. If the step depends on the current and the previous state, we get a somewhat more complicated second-order Markov model. Its states correspond to all possible pairs of actions that can be performed in a sequence. We can generalize this approach to the Kth-order Markov model, which computes the predictions by looking at the last $K$ actions performed by the user, leading to a state-space that contains all possible sequences of $K$ actions (6).

## 2.3 Related work

Data pre-processing is important part of web usage analysis since it requires large amount of time and affects the results of analyses. This problem motivated researchers to develop new methods for pre-processing.

Colley et al. (3) proposes a series of steps for data pre-processing for web usage mining. These include data cleaning, user identification, session identification and data formatting. Zhang et al. (13) improved statistical-based time oriented heuristics for the reconstruction of user sessions. They used statistical analysis and usage mining techniques to improve time-oriented heuristics. Ting et al. (11) developed the Pattern Restore Method (PRM) algorithm, which attempts to reconstruct missing server-side clickstream data based on referring site information and the Website's link structure. Berendt et al. (1) used the web site structure to reconstruct incomplete sessions.

Markov models have also been used in the clickstream analysis area. Many approaches have been proposed. In (2) the authors primarily focus on visualization aspects of website navigation patterns. Model-based clustering (using finite mixtures of Markov models) is used to assign users to clusters. Sarukkai (10) employs Markov chains to both predict the most likely sites that a user will visit next and generate tours (sequences of websites) that a user might be interested in according to his or her current browsing history. The model can be continuously updated with data provided by new users of the websites it covers.

In (6) authors look at the ways of reducing the state-space complexity of higher order Markov models, while retaining their high coverage. This is achieved by first building a full model from some of the training data, then pruning it with the rest. The results show that these methods can greatly reduce the state-space complexity while generally improving its accuracy. Ypma and Haskes (12) expanded the work done by Cadez and Heckerman by using mixtures of Hidden Markov models. This enabled them to process the dataset without first grouping actual URI requests into page categories. Their work shows that even without artificially categorized webpages, a mixture of HMMs will generate classes of pages with similar characteristics.

## 2.4 Separating interleaved sessions with Markov model

The process of separating interleaved sessions is one of the phases in data pre-processing. First, clickstream data has to be cleaned and sessionized. We refer to sessions, that have been restored without deficiencies, as *clean* sessions. Durring the sessionization process we detect interleaved sessions which we cannot separate at that time either by using some background knowledge, or by applying a pre-trained Markov model (MM). Interleaved sessions are separated from clean sessions and are additionally processed. The separation process is based on stochastic methods which have been used to solve some other issues related to cliskstream. Because of generality and simplicity we decided to use first-order Markov model. We build a Markov model and train it with data from clean sessions. Training proceeds as follows. If there is a transition $s_i \rightarrow s_j$ in training data, the frequency counter $n_{ij}$ is incremented by one. We can use last pre-processing clean sessions or clean sessions from last few pre-processings. Trained markov model is then used to separate interleaved sessions. In case of more than two interleaved sessions only the first one is considered as clean, and the second one is submitted to further separation. This results in more reliable pre-processed user behavior data. The last step in a analysis is evaluation of separated sessions with several methods.

For separating interleaved sessions we use a trained first-order MM. We utilize site map data as background knowledge. Site map consists of links between pages that are explicitly connected with hyperlinks. A link between pages $S_1$ and $S_2$ in a site map means higher prior probability of transition between these two pages than if there were no link in a site map. When we train the MM we also use the web site map. Based on links between page sites we calculate initial transition probability between pages $p_{ij}^{(0)}$, where $i, j$ denotes source and target state. Formula for calculating $p_{ij}^{(0)}$:

$$p_{ij}^{(0)} = \frac{1 - P_A^{(ij)}(N - n_t)}{n_t}, n_t \geq 1 \tag{7}$$

where $j$ denotes all states that are connected to state $i$, $N$ denotes number of states, $n_t$ number of outgoing links from state $i$[1] and $P_A^{(ij)} = 1/N^2$ an uninformed probability of transition between any two states. If there is no connection between $i$ and $j$, probability $P_A^{(ij)}$ is assigned. Parameter $P_A^{(ij)}$ determines the prior probability of transition between arbitrary two pages in the site map.

Let each session be represented as sequence of pages $S = \{q_1, q_2, \ldots q_n\}$ where $n$ denotes length of session. $q_1$ denotes the entry page and $q_n$ the last page the user visited in this session. For a transition from $q_{i-1} = s_j$ to $q_i = s_k$, training data site map data can be combined with *m-estimate* (5):

$$P(s_j \rightarrow s_k) = p_{jk} = \frac{(n_{jk} + m p_{jk}^{(0)})}{n_j + m} \tag{8}$$

where $n_{jk}$ denotes number of transitions from state $j$ to $k$, which we got from training data. $n_j$ is number of visits of state $j$. $m$ denotes the weight which presents the ratio between prior (web site map) and posterior knowledge. $p_{jk}^0$ denotes transition probability based on web site map. Parameter $m$ represents the importance rate of prior knowledge. The higher the $m$ is, the more important the prior knowledge is. If $m = 0$, then we completly neglect the meaning of prior knowledge. In that case m-estimate converts to relative frequency $p_{jk} = n_{jk}/n_j$.

## 2.5 The separation process

Separating interleaved session is based on a fact that a transition between sites $q_i \rightarrow q_{i+1}$ is more likely to belong to one of the consisting sessions. If we have interleaved session $S_p = [q_1, q_2, \ldots, q_n]$ that consists of two clean sessions length $n_1$ and $n_2$, where $n_1 + n_2 = n$. The number of

[1] We assume that there is always the reflective transition from $s_i$ to $s_i$, so $n_t$ is always greater than 0.

possible different separations is $C = \binom{n_1+n_2}{n_1} = \binom{n_1+n_2}{n_2}$. Let us say that the last page of the first session that we already managed to separate is $S_{1i}$. Similarly for the second session we denote the last page as $S_{2i}$. For each page $S_i$ in an unprocessed interleaved session, we check what is the transition probability from last page of separated session to current page $S_i$. If $P(S_{1i} \rightarrow S_i) > P(S_{2i} \rightarrow S_i)$ we add page $S_i$ to the first separated session, otherwise to the second one. Until both of the separated sessions get the first element (entry page), we have to check whether $S_i$ is an entry page for second session. Separating process can be seen on Figure 1.

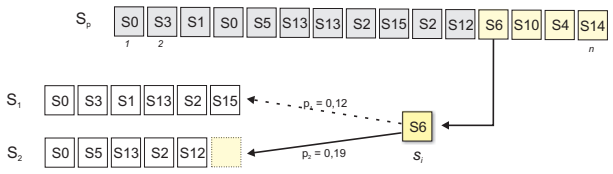


Figure 1: Figure shows simple process of separating interleaved session.

## 2.6 Evaluation of separating process

Separated sessions needs to be evaluated to see how successful our method was. Each session is represented as a sequence of pages. Evaluating quality of separated sessions can be viewed as evaluating their similarity. Determining the similarity between sequences is one of the basic tasks in machine translation as well as in computational biology (8). Basically, two sequences are more similar if they have more symbols in common and the symbols' order is similar. There are many methods of measuring similarity between two sequences. We use several more or less strict methods based on: perfect match, Levenshtein distance, longest common subsequence (LCS) and weighted longest common subsequence (9).

*Perfect match* is a simple method where only sequences that perfectly match contribute to the end result.

Alternative approach to measure sequence similarity is based on sequence distance, named *edit distance*. The distance between two sequences is defined as the smallest sum of edit operations' costs that transforms one sequence to another. If we have only three edit operations: inserting, deleting and swapping symbols, and all have the cost of 1, we get *Levenshtein distance*.

A sequence $Z = [z_1, z_2, ..., z_n]$ is a subsequence of another sequence of sequence $X = [x_1, x_2, ...x_m]$ if there exists a strict increasing sequence $i_1, i_2, ...i_k$ in $X$ such that for all $j = 1, 2, ..., k$ we have $x_{ij} = z_j$ (4). If we have sequences $X$ and $Y$, the longest common subsequence of $X$ and $Y$ is a common subsequence with the maximum length. The longer the common subsequence, the more two sessions are similar to each other. One advantage of LCS is that it does not require consecutive matches but in-sequence matches that reflect level element order as n-grams. Deficiency of LCS is that it only counts the main in-sequence elements. Other common subsequences are not reflected in a result (8). We estimated these methods are apropriate for evaluation of separating process.

We can improve LCS method to differentiate LCS in relation to other elements in the sequence. Chin et al. (9) called this method *weighted LCS* (WLCS). They also propose the use of *F-measure* to estimate the similarity between two sequences $X$ of length $m$ and $Y$ of length $n$. We decided to use F-measure for presenting end results.

# 3 Materials

## 3.1 Synthetic data

First we created a test environment that is similar to real one but is not as complex. We checked what is the average HTTP session length on a local web server. For testing we fixed the number of Web pages to 30. We created an artificial web site map that represented links with higher probability. According to the site map we generated a number of sessions that were used for MM training data, and some of them for creating interleaved sessions. After training MM, we applied the process for separating interleaved sessions and verified the results. About 48% of interleaved sessions were separated 100% correctly, which encouraged us to proceed to real data.

## 3.2 Real-world data

We applied the interleaved session separating process on two real clickstream sources. The first clickstream originates from log files of university student records information system. It has been used by 16 member institutions. It has approximately 300 different pages. Each state in MM corresponds to an individual page. Typical user paths are well defined. Users have to be logged on in order to use the system. Sometimes they are logged on with different user roles at the same time, and this creates interleaved sessions. Since users have to be logged on we can always determine the session entry point. The Web server log files use the basic CLF format. Clickstream data was taken for 4 months of use, which resulted in 150.000 user sessions.

The second clickstream source is taken from a web shop, which is considerably different from the student records information system. Users do not have to sign in (except for buying items), it has many more users and many more pages. We had to cut down number of states of Markov model in order to efficiently use it. Every state of our Markov model represents a group of pages, not an individual page. We transformed the web shop pages to 900 states. Session entry point can be almost any page, which makes separating interleaved sessions harder. The Web shop site map has plenty of links between pages. In fact only few pages are not linked with all others. The web shop generates about 10.000 user sessions a day.

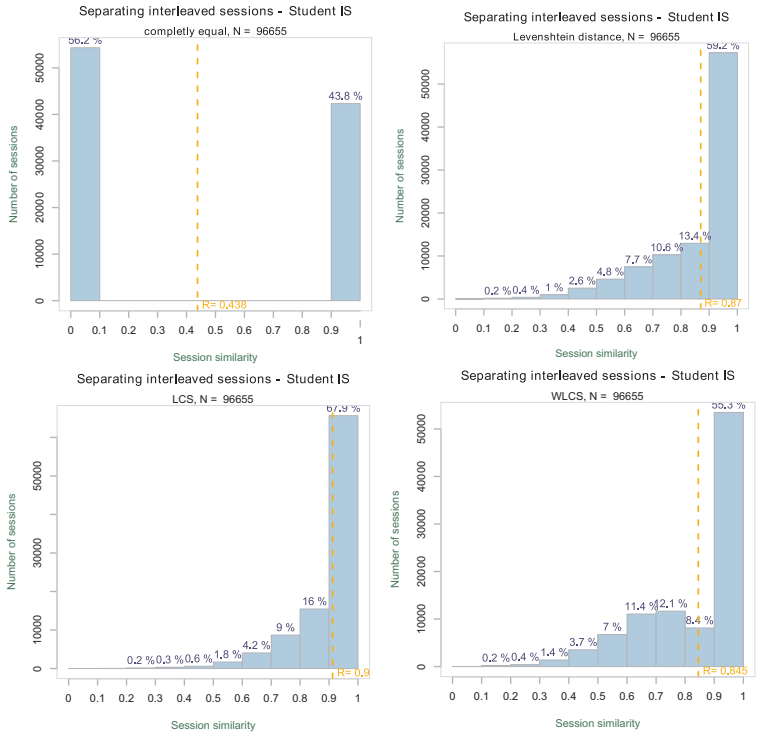


Figure 2: Results of separating for Student records IS clickstream. R denotes weighted average of session similarities.

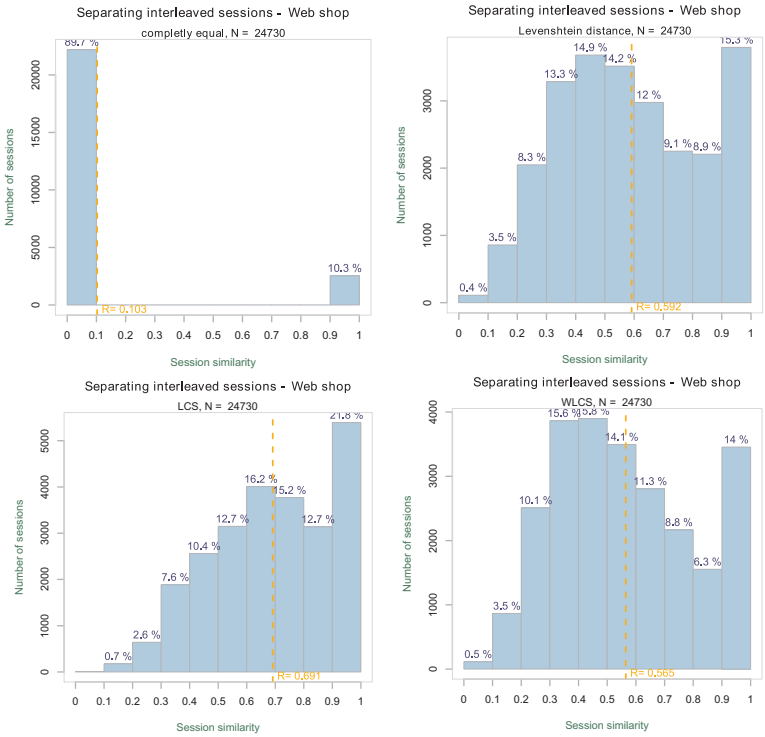


Figure 3: Results of separating for Web shop clickstream. R denotes weighted average of session similarities.

For both clickstreams we took the same steps as with artificially generated data. Initial clean sessions, used for learning, were generated during the sessionization process of clickstream data. During the sessionization we applied all the neccessary steps in order to remove noisy data. We analysed what a typical user session looks like and removed all sessions that did not meet the rules (e.g. too short or too long sessions). 70% of clean sessions were used as a training set for MM, and the rest were used to generate interleaved sessions in order to evaluate the separation process. After separating interleaved sessions we evaluated results with evaluation methods that we presented earlier.

# 4 Results

In Figures 2 and 3 we can see graphs for evaluation methods and source of clickstream. Each graph corresponds to one evaluation method. The $X$ axis shows intervals for F-measure based similarity and the $Y$ axis shows number of sessions that fall in that interval. Figure 2 reports results for student IS clickstream. $96655$ interleaved sessions have been created and separated. On the first graph we see that 43% sessions have been separated 100% correctly (session sequence similarity = 1). This result is much better in comparison with Web shop. Other three graphs on at Figure 2 depict how well the sessions have been separated according to evaluation method. LCS and WLCS graphs show that majority of sessions are more than 50% similar to the original ones.

If we look at Figure 3 we see results for Web shop. $24730$ interleaved sessions have been created and separated. Looking at the first graph in that Figure, one sees how many sessions have been separated 100% correctly. For web shop this percentage is a little more than $10\%$, which is quite low. However even 10% is better than throwing away all interleaved sessions. One of the reasons is that grouping pages together affects the results. Since the site map is larger, there may be numerous user paths, what also affects the results. User can enter the web shop at almost any page, so it is harder to detect where the second session in interleaved session starts. Results on a graph that show LCS seem better, since LCS is a less strict method of evaluation than WLCS.

# 5 Conclusion

We propose a new method for improving the quality of clickstream data in pre-processing phase that is based on a first-order Markov model. To the very best of our knowledge, the Markov approach has not been used for this purpose before. Proposed method is very effective in deinterleaving sessions (linear complexity). We present the motivation that led us to implementation and have applied method on two real data clickstreams. The presented results show that in certain cases method gives promising results. We analysed the domain and detected possible causes of worse results. In order to minimize method deficiencies we plan to work on the issues we presented. First we have to improve the method for detecting interleaved session starting pages. We are also planning to use second-order Markov model and Hidden Markov Model (HMM) for separating process.

# References

[1] Bettina Berendt, Bamshad Mobasher, Miki Nakagawa, and Myra Spiliopoulou. The impact of site structure and user environment on session reconstruction in web usage analysis. In *WEBKDD - KDD Workshop on Web Mining and Web Usage Analysis*, pages 159–179, 2002.

[2] Igor Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. Model-based clustering and visualization of navigation patterns on a web site. *Data Min. Knowl. Discov.*, 7(4):399–424, 2003.

[3] Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1:5–32, 1999.

[4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press and McGraw-Hill Book Company, 1989.

[5] S. Džerovski, B. Cestnik, and I. Petrovski. Using the m-estimate in rule induction. *J. Comput. Inf. Technol.*, 1(1):37–46, 1993.

[6] Mukund Deshpande and George Karypis. Selective markov models for predicting web page accesses. *ACM Trans. Interet Technol.*, 4(2):163–184, 2004.

[7] Ron Kohavi. Mining e-commerce data: The good, the bad, and the ugly. In Foster Provost and Ramakrishnan Srikant, editors, *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 8–13, 2001.

[8] G. Leusch, N. Ueffing, and H. Ney. A novel string-to-string distance measurewith applications to machine translation evaluation. In *In Proceedings of MT Summit IX*, pages 240–247, 2003.

[9] C-Y. Lin and F. J. Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 605, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[10] R. R. Sarukkai. Link prediction and path analysis using markov chains. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking*, pages 377–386, Amsterdam, The Netherlands, The Netherlands, 2000. North-Holland Publishing Co.

[11] I-Hsien Ting, Chris Kimble, and Daniel Kudenko. A pattern restore method for restoring missing patterns in server side clickstream data. *Lecture Notes in Computer Science*, 3399:501–512, March 2005.

[12] Alexander Ypma and Tom Heskes. Automatic categorization of web pages and user clustering with mixtures of hidden markov models. In *WEBKDD*, pages 35–49, 2002.

[13] J. Zhang and A.A. Ghorbani. The reconstruction of user sessions from a server log using improved time-oriented heuristics. In *Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on*, pages 315–322, May 2004.

[14] M. Viermetz, C. Stolz, C. Gedov, and M. Skubacz. Relevance and Impact of Tabbed Browsing Behavior on Web Usage Mining. In *Web Intelligence, 2006. WI 2006, 2006. Proceedings. IEEE/WIC/ACM International Conference on*, pages 262–269, Dec 2006.