Journal of Universal Computer Science, vol. 11, no. 8 (2005), 1383-1396
submitted: 10/3/05, accepted: 5/5/05, appeared: 28/8/05 © J.UCS

Semantic Preprocessing of Web Request Streams for Web Usage Mining

Jason J. Jung
(School of Computer and Information Engineering,
Inha University, Korea
j2jung@intelligent.pe.kr)

Abstract: Efficient data preparation needs to discover the underlying knowledge from complicated Web usage data. In this paper, we have focused on two main tasks, semantic outlier detection from online Web request streams and segmentation (or sessionization) of them. We thereby exploit semantic technologies to infer the relationships among Web requests. Web ontologies such as taxonomies and directories can label each Web request as all the corresponding hierarchical topic paths. Our algorithm consists of two steps. The first step is the nested repetition of top-down partitioning for establishing a set of candidates of session boundaries, and the next step is evaluation process of bottom-up merging for reconstructing segmented sequences. In addition, we propose the hybrid approach of this method, as combining with the existing heuristics. Using synthesized dataset and real-world dataset of the access log files of *IRCache*, we conducted experiments and showed that semantic preprocessing method improves the performance of rule discovery algorithms. It means that we can conceptually track the behavior of users tending to easily change their intentions and interests, or simultaneously try to search various kinds of information on the Web.

Key Words: Web usage mining, semantic analysis, browsing patterns
Category: H.3.3, I.5.3

1 Introduction

As the very large amount of Web log and request data have been generated on the Web, the concerns for searching relevant information from the Web have been exponentially increasing. Thus, users have lost their way in the Web space, as starting to enter a specific URL, clicking on hyperlinks to the other Web nodes, and going "backward" and "forward" along a stack of already visited nodes [Bielecki et al. 2002]. This phenomena *information overloading* of Web has caused navigational problem to users. However, we suppose that Web mining can support them through recognizing what they are interested in and predicting which requests will be occurred next. The sequence of Web requests can be regarded as the evidence implicitly including user intention generated during "tedious browsing tasks." In fact, many applications have been focusing on the analysis of Web requests as well as Web logs, in order to recognize the client usage patterns and user preferences and to discover additional meaningful patterns [Cooley et al. 1997]. For example, on-line newspaper on the Web [Batista and Silva 2001], and Web caching [Bonchi et al. 2001] can be told as the domains applicable to analyzing Web data. Especially, for supporting user's Web browsing, "Letizia" [Lieberman 1995] is the well-known personal agent-based system concurrently anticipating Webpages of user interests and recommending some candidates of them. Moreover, [Berendt and Spiliopoulou 2000] have shown the WUM (Web Usage Miner) discovering navigation patterns from "SchulWeb," by constructing concept hierarchies for integrating multiple information systems.

In this paper, we want to deal with the preprocessing task of online data streams. For mining personal Web usages such as user profiling and browsing pattern discovery, data preparation process is very important [Cooley et al. 1999], [Pierrakos 2003]. Also, the reliability of data mining depends on the quality of these data, which may be noisy, erroneous, and incomplete [Spiliopoulou 2003]. Due to the domain-specific characteristics of Web log data, particularly, session identification methods should be considered to efficiently segment streaming Web requests. Because the Web requests implicitly indicate what the user is looking for, they should be sessionized through semantic enrichment process with the topics extracted from Web contents in order to find out more potential and meaningful information like a user's preference and intention. More importantly, Web caching (or proxy server) systems have to track multiple clients by analyzing massively streaming Web requests, because they have to increase the predictability for prefetching Web contents possibly requested in next time. In the following Sect. 1.1, two simple heuristics-based sessionization methods are introduced. However, knowledge discovery from sessions identified by these heuristics is limited to the only simple patterns like frequent and sequential patterns represented by Web requests. More seriously, they are impossible to deal with the complicated browsing patterns such as multiple activities and multiple navigation, which mean that users usually request more than one Web request simultaneously by using more than a Web browser.

1.1 Two heuristic-based sessionization methods

There are mainly two kinds of heuristic methods for partitioning each user's activities into sequences of entries [Spiliopoulou 2003]. First, *time-oriented* heuristics consider temporal boundaries such as a maximum session length (time window) or maximum time allowable for each pageview [Berendt et al. 2002]. There have been several empirical studies to calculate the time spent inside a Website. Catledge and Pitkow proved that the mean inactivity time within a certain site was a value of 9.3 minutes [Catledge and Pitkow 1995]. Cooley et al. derived a 25.5 minute cutoff for the duration of a visit by adding 1.5 standard deviations [Cooley et al. 1999]. Approximately this has been rounded to a half of an hour and has been used in many applications as a rule of thumb for maximal session length [Spiliopoulou and Faulstich 1999]. While visiting a Website, users need to take some time to read and recognize the contents in Webpages. The duration of this time between one request and the next is a reasonable implication of the fact that page-stay time is affected by the information content of a page, by the time needed to load the components of the page and by the transfer speed of the communication line.

Second, *navigation-oriented* heuristics take the linkage between Webpages into account [Chen et al. 2002]. This heuristics exploit behavioral patterns associated with Web navigation. In [Cooley et al. 2000], [Cooley et al. 1999], a requested Webpage that is not reachable from previously visited pages should be assigned to a different session, and vice versa. This heuristic also accounts for the fact that Web requests need not be accessible from the page immediately accessed before it. The topological graph structure of Website is another important issue related to navigation oriented heuristics. [Cooley et al. 1999] introduced simple topology heuristics based on the referral information extracted from the extended log file format and [Berendt et al. 2001] proposed the extension of this heuristic as handling some drawbacks caused by empty referral. For example, maximum forward referencing and reference length establishing are the most representative link analysis methods. Moreover, Webpage ranking algorithms such as HITS, PageRank, and DirectHit have developed various approaches to recognize the relationship between Webpages.

2 Topic Distillation for Semantic Labeling

Web requests can be regarded as the user intention and interests that they are trying to search. Thereby, we have to extract features from the Web requests such as term frequency, hyperlinks, and URLs. We employ a Web directory as the supervisor for semantic labeling based on simple URL information and assume that a label is represented as the corresponding paths on topic hierarchy. For example, Springer-Verlag (http://www.springer.de/) requested by a user can be categorized to “Germany > Publishers > Springer-Verlag” directory.

Web directories, however, are not possible to label all Websites. When semantically labeling Web requests from users, we are considering two ways to extract the topics related to them. First, some URLs registered in a Web directory are directly labeled. Second way is the link analysis-based indirect labeling for Websites unregistered in the Web directory. Furthermore, some practical drawbacks of Web directories will be described, and then, we propose how to deal with these problems in this paper.

2.1 Web Directories and Web Requests

An ontology, the so-called semantic categorizer, is an explicit specification of a conceptualization [Gruber 1993]. It means that the ontology can play a role of enriching semantic or structural information to unlabeled data. We have regarded the Web directory as topic-specific ontology. Web directories like Yahoo (http://www.yahoo.com/) and Cora (http://cora.whizbang.com/) can be used to describing the content of a document in a standard and universal way as ontology [Labrou and Finin 1999]. Besides, a Web directory organized as a topic hierarchical structure is an efficient way to organize, view, and explore large quantities of information that would, otherwise, be cumbersome [McCallum et al. 1999].

In this paper we assume that Web requests from users can be labeled by a well-organized Web directory. There are, however, some practical obstacles to simple URL based labeling, because most of Web directories are forced to manage a non-generic tree structure in order to avoid a waste of memory space caused by redundant information [Jung 2005]. We briefly note that problems with categorizing an URL with Web directory as an ontology are the following:

- **The multi-attributes of a Website.** A Website can be involved in more than a topic. The causal relationships between categories makes their hierarchical structure more complicated.
- **The relationship between categories; subordination and redundancy.** A category can have more than a path from root node and be a subcategory of more than one parent category. Furthermore, some categories can be semantically identical, even if they have different labels.

In characterizing the structure and content of a Webpage, it is necessary to establish precise data model of the Web requests. A Web request is a message issued by a Web client such as Web browser, and it can be described as explicit and implicit manifestations [W3C 1999]. Implicit requests initiated transparently by the Web clients without user interventions have been discarded in this paper.

2.2 Two-way Labeling Based on Web Directory

For labeling of Web requests, we extract URL information from "Host" feature of Web request and conduct the labeling process. There are two kinds of labeling, which are direct and indirect labeling. It depends on whether this Website is registered on Web directory. Direct labeling is simple querying process looking up the corresponding URL from Web directory. In order to deal with the drawbacks of Web directory, we have to acquire a set of labels including all possible paths as the result. On the other hand, indirect labeling is needed for unregistered Websites. It is based on link analysis for searching "authoritative" pages about a certain topic on the hyperlinked space like Web [Kleinberg 1999],

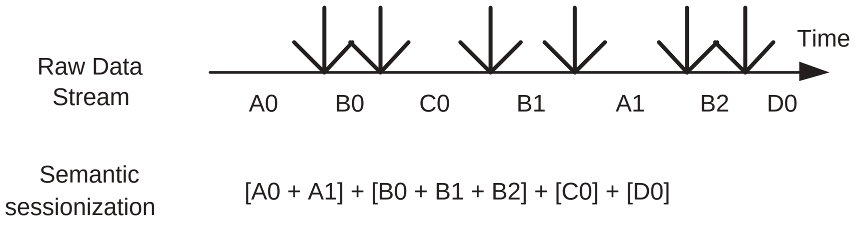

Figure 1: Two steps of semantic sessionization

[Ding et al. 2002]. We propose modified HITS algorithm searching the most similar data from already labeled dataset. Let a Website M requested by clients not to be registered yet on Web directory. ¿From links of M, we can reach the Website X, the nearest neighbor category registered on Web directory. The hyperlinked Webpages organize a directed graph $G = (V, E)$, where V is the set of nodes representing Websites, and E is the set of hyperlinks between v_i and v_j. In order to search the most authoritative node of a particular Website, we focus on ongoing links of that Website. Outgoing and incoming links of graph G can be formulated as the asymmetric adjacency matrix $O(M)^{(d)}$, where $O(M)_{ij} = 1$ if $p_i \rightarrow p_j$ and $O(M)_{ij} = 0$, otherwise. Also, the variable d is the number of iterated expansion, which means the distance from the node M. Therefore, we can reach some labeled nodes, as repeating this iteration along outgoing links. If there are more than one labeled nodes at the same distance, we have to evaluate the incoming degree of those nodes by using the following equation

$$O(M)_X = \max_{j^\star \in j} \left[\sum_k O(M)^{(d)}_{kj^\star} \right] \tag{1}$$

where $j^\star$-th Websites are labeled. It means that the Websites can be regarded as more authoritative one, as they are referred by more other Websites.

3 Semantic sessionization for data preparation

In order to segment the sequence of labeled data for supporting efficient data mining tasks, we establish a novel approach based on conceptualization of ontology. We introduce semantic factors for quantifying the relationships between the labeled Web requests. Then, detecting semantic outliers from the sequence of Web requests will be described, with respect to static and dynamic cases.

During top-down partitioning step, we can find out the session identifiers (boundaries) as a set of semantic outliers. Then, bottom-up merging establishes more sophisticated sessions through connecting the sequences generated by similar semantics. As shown Fig. 1, vertical arrows mean semantic outliers detected from streaming dataset. Even if six semantic outliers can be detected through top-down partitioning, each fragment should be merged into a common session by bottom-up merging step. Eventually, the raw data stream is semantically sessionized to only four segments A, B, C, and D.

3.1 Preliminary notations and definitions

We define the semantic factors measuring the relationship between two Web requests. After labeling two arbitrary Web requests by referring to the Web directory, we can obtain both sets of all possible categorical and ordered paths for the corresponding requested URLs. Firstly, the semantic distance is formulated for measuring how semantically different these URLs are between each other. Let a URL url_i categorized to the sets

$$\{path_i | path_i^m \in Category(url_i), m \in [1, \ldots, M]\} \tag{2}$$

where M is the number of all possible categorical paths. As simply extending *Levenshtein* edit distance [Levenshtein 1966], the semantic distance $\mathit{\Delta}^\diamond$ between URLs url_i and url_j is given by

$$\mathit{\Delta}^\diamond[url_i, url_j] = \arg \min_{m=1,n=1}^{M,N} \frac{\min((L_i^m - L_C^{(m,n)}), (L_j^m - L_C^{(m,n)}))}{\exp(L_C^{m,n})} \tag{3}$$

where L_i^m, L_j^n, and $L_C^{(m,n)}$ are the lengths of $path_i^m$, $path_j^n$, and common part of both, respectively. As marking paths representing the labeled URLs on trees, we can easily get this common part overlapping each other. The semantic distance $\mathit{\Delta}^\diamond$ compares all combinations of two sets ($|path_i| \times |path_j|$) and returns the minimum among these values in the interval $[0, 1]$, where 0 means complete matching. Exponent function in denominator is used in order to increase the effect of $L_C^{(m,n)}$. Second factor is to aggregate a series of adjacent URLs during a given time interval. Thereby, semantic distance matrix $D_{\mathit{\Delta}^\diamond}$ is given by

$$D_{\mathit{\Delta}^\diamond}(i, j) = \begin{bmatrix} \ldots & \ldots & \ldots \\ \ldots & \mathit{\Delta}^\diamond[url_i, url_j] & \ldots \\ \ldots & \ldots & \ldots \end{bmatrix} \tag{4}$$

where the size of this matrix is the predefined time interval T and the diagonal elements are all zero. Based on $D_{\mathit{\Delta}^\diamond}$, the semantic mean $\mu^\diamond$ is given by

$$\mu^\diamond(t_1, \ldots, t_T) = \frac{2 \sum_{i=1}^{T} \sum_{j=i}^{T} D_{\mathit{\Delta}^\diamond}(i, j)}{T(T-1)} \tag{5}$$

where $D_{\Delta^\diamond}(i,j)$ is the (i,j)-th element of distance matrix. This is the mean value of upper triangular elements except diagonals. Then, with respect to the given time interval T, the semantic deviation $\sigma^\diamond$ is derived as shown by

$$\sigma^\diamond(t_1,\ldots,t_T) = \sqrt{\frac{2\sum_{i=1}^{T}\sum_{j=i}^{T}\left(D_{\Delta^\diamond}(i,j) - \mu^\diamond(t_1,\ldots,t_T)\right)^2}{T(T-1)}} \tag{6}$$

These factors are exploited to quantify the semantic distance between two random logs and statistically discriminate semantic outliers such as the most distinct or the N distinct data from the rest in the range of over preset threshold, with respect to given time interval.

3.2 Semantic outlier detection from static Web requests

When we try to segment the Web requests dataset, these sequential entries are generally time-varying, more properly, streaming. In this section, we simply assume that a given dataset is time-invariant and its size is fixed. Furthermore, for handling this streaming dataset, we have to consider to compute not only the semantic factors in a given interval but also the distribution of the semantic mean $\mu^\diamond$ by sliding windows method, and this case will be discussed in the Sect. 3.3.

The semantic outliers are evaluated through bottom-up merging process establishing the most optimal partitioning of given dataset. We want to obtain the best combination of semantic outliers, which is making the sum of partial semantic deviation $\mu^\diamond$ for each session minimized. Thereby, the principle session identifiers

$$PSI = \{psi_a | a \in [1,\ldots,S-1], psi_a \in [1,\ldots,T-1]\} \tag{7}$$

is defined as the set of boundary positions, where the variables S and T are the required number of sessions and the time interval, respectively.

The semantic outlier analysis for sessionizing static logs SOA_S as objective function with respect to PSI is given by

$$SOA_S(PSI) = \sum_{i=1}^{S} \mu_i^\diamond \tag{8}$$

where $\mu_i^\diamond$ means partial semantic deviation of i^{th} segment. In order to minimize this objective function, we scan the most distinct pairs, in other words, the largest value in the semantic distance matrix $D_{\Delta^\diamond}$, as follows:

$$\Delta^\diamond_{MAX}[T_a,T_b] = \arg\max_{i=1,j=1}^{T} D_{\Delta^\diamond}(i,j) \tag{9}$$

where $\arg\max_{i=1}^{T}$ is the function returning the maximum values during a given time interval $[T_a,T_b]$. When we obtain $D_{\Delta^\diamond}(p,q)$ as the maximum semantic distance, we assume there must be at least a principle session identifier between

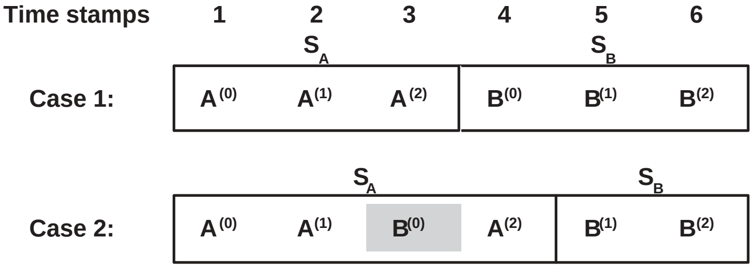

Figure 2: Example for top-down approaching sessionization.

p-th and q-th URLs. Then, the initial time interval $[T_a, T_b]$ is replaced by $[T_p, T_q]$, and the maximum semantic distance in reduced time interval is scanned, recursively. Finally, when two adjacent elements are acquired, we evaluate this candidate psi by using $SOA_S(psi)$. If this value is less than $\sigma^\diamond$, this candidate psi is inserted in PSI. Otherwise, this partition by this candidate psi is cancelled. This sessionization process is top-down approaching, until the required number of sessions S is found. Furthermore, we can also be notified the over-sessionization, which is a failure caused by overfitting sessionization, detected by the evaluation process $SOA_S(PSI)$. For example, let a URL entry composed of two sessions S_A and S_B in two cases, as shown in Fig. 2. We assume that the semantic distances between A$^{(i)}$s (or B$^{(i)}$s) is much less than between each other. The four largest semantic distances $\Delta^\diamond[A^{(1)}, B^{(1)}]$, $\Delta^\diamond[A^{(2)}, B^{(2)}]$, $\Delta^\diamond[A^{(2)}, B^{(0)}]$, and $\Delta^\diamond[A^{(2)}, B^{(1)}]$ are 0.86, 0.85, 0.81, and 0.79, respectively. We want to segment them into two sessions. In Case 1, due to the maximum distance $\Delta^\diamond[A^{(1)}, B^{(1)}]$ in the initial time interval $[1, 6]$, time interval is reduced to $[2, 5]$, and then, $\Delta^\diamond[A^{(2)}, B^{(0)}]$ in updated time interval determines that psi_3 can be a candidate. Finally, the evaluation $\sigma^\diamond[1, 3] + \sigma^\diamond[4, 6] < \sigma^\diamond[1, 6]$ makes a candidate psi_3 inserted to PSI. This case is clear to find the candidate psi and prove this sessionization to be validate. More complicatedly, in Case 2, a heterogeneous request $B^{(0)}$ is located in the session S_A. The first candidate psi_3 is generated by $\Delta^\diamond[B^{(0)}, A^{(2)}]$ in time interval firstly refined by $\Delta^\diamond[A^{(1)}, B^{(1)}]$. By the evaluation $\sigma^\diamond[1, 3] + \sigma^\diamond[4, 6] \geq \sigma^\diamond[1, 6]$, however, this candidate psi_3 is removed. Finally, because the second candidate psi_4 by $\Delta^\diamond[A^{(2)}, B^{(1)}]$ meets the evaluation $\sigma^\diamond[1, 4] + \sigma^\diamond[5, 6] < \sigma^\diamond[1, 6]$, a candidate psi_4 can be into PSI.

3.3 Session identification from Web requests

Actually, on-line Web logs are continuously changing. It is impossible to consider not only the existing whole data but also streaming data. We define the time window W as the pre-determined size of considerable entry from the most recent one. Every time new URL is requested, this time window have to be shifted. In order to semantic outlier analysis of streaming logs, we focus on not only basic semantic factors but also the distribution of the semantic mean with respect to time window, $\mu^{\diamond}(W^{(T)})$.

As extending SOA_S, the objective function for analyzing semantic outlier of dynamic logs SOA_D is given by

$$SOA_D^{W^{(i)}}(PSI) = \sum_{k=1}^{S} \mu_k^{\diamond}|_{W^{(i)}} \tag{10}$$

where the $W^{(i)}$ means that the time window from i^{th} URL is applied. We want to minimize this $SOA_D(PSI)$ by finding the most proper set of principle session identifiers. The candidate psi_i is estimated by the difference between the semantic means of contiguous time windows and predefined threshold ε, as shown by

$$\left|\mu^{\diamond}(W^{(i)}) - \mu^{\diamond}(W^{(i-\tau)})\right| \geq \varepsilon \tag{11}$$

where τ is the distance between both time windows and assumed to be less than the size of time window $|W|$. Similar to the evaluation process of SOA_S, once a candidate psi_i is obtained, we evaluate it by comparing $SOA_D^{W^{(i)}}$ and $SOA_D^{W^{(i-1)}}$. Finally, we can retrieve PSI to sessionize streaming Web logs. In case of streaming logs, more particularly, a candidate psi meeting the evaluation process can be appended into unlimited size of PSI.

4 Mining user interests from sessionized Web requests

A sequence is an ordered list of elements, a set of items appearing together in a transaction. The general goal of sequence mining is to discover the sequences of maximal length with predefined support, from given a collection of transactions ordered in time. Basically, in [Agrawal and Srikant 1995], frequent patterns are built incrementally by discovering frequent patterns as extending them stepwise to patterns of lager sizes. However, disadvantages for applying this algorithm to Web usage mining are mentioned, in [Wang 1997], [Schechter et al. 1998], [Berendt and Spiliopoulou 2000]. For example, simple frequency based algorithm is hard to discovery relationship between linked Webpages.

In this study, we have defined a browsing pattern for recognizing user interests as two kinds of aspects; i) the co-occurrences among Websites, and ii) the sequential accessing of several Websites. We therefore exploit two kinds of pattern mining algorithms, which are association rule mining (*Apriori* algorithm) [Agrawal et al. 1993], [Agrawal and Srikant 1994] and generalized sequential pattern mining (GSP algorithm) [Srikant and Agrawal 1996] in order to discover browsing patterns. Each session can be easily identified by selecting a PSI. It consists of a sequence of Web requests ordered by timestamps. We can mine browsing pattern such as "$url_a \rightarrow url_b$" with 80% support and 100% confidence and "$(url_a, url_c) \rightarrow url_b$" with 40% support and 40% confidence, by using *Apriori* algorithm without considering temporal information. On the other hand, a generalized sequential pattern can be discovered like "$url_b \rightarrow url_h \rightarrow url_e$" based on GSP algorithm.

Now, we can easily retrieve user interests. Because URL information applied to the extracted browsing patterns are already labeled to specific topics, we therefore can explicitly infer a set of topics that a user is interested in. This set of topics is represented as hierarchical topic paths. As overlapping each path of them on the topic hierarchy, some topics can be regarded as the major interest of users. Additionally, we can visualize user interests in this way.

5 Experimental results

We have proved the effect of semantic labeling based on ontologies for sessionization and the applicability to Web applications. The ODP (Open Directory Project) was employed as ontology. The first experiment is to verify the effects of semantic sessionization through measuring *precision* and *recall*. We organized the testing bed consisting of 2530 Webpages labeled from ODP, and two user groups ($U_{Semantic}$ ans U_{Time}) whose profiles were already established by questionnaire, before started experiments. Web requests from these user groups were preprocessed by semantic sessionization and time-oriented sessionization, respectively. The main topics can be extracted from given a set of sessions, because Webpages as testing bed are already labeled. Thus, we were able to match them with newly extracted topics that users are likely to be interested in, for evaluating the performance of extraction of user interests. The *recall* of our proposed method is remarkably higher than those of the others. This measure is for the coverage meaning how much the proper answers are retrieved. It proves that semantic sessionization is much more reliable to find the boundaries of each sessions. This method can make Web requests related to the same topics in a same session. Not only the recall but also the precision showed 77.0% and 44.3% improvements, compared with the other simple heuristics.

As the second experiment, we have tried to predict the next Web requests that client will access to for Web content prefetching, one of the most important functions of personalized Web browser and Web proxy server. For the experiment using real-world data, we collected the sanitized access logs from *sv.us.ircache.net*, one of Web cache servers of *IRCache*. These raw files, generated from 20 March 2003 to 26 March 2003, consist of eleven attributes and about 9193000 entries. During data cleansing, log data whose URL field is empty and ambiguous (wrong

Table 1: The number of sessions by time-oriented heuristics and semantic sessionization (static and dynamic logs) from logs for seven days (20-26 March 2003)

	1	2	3	4	5	6	7
Time-oriented	1563	1359	1116	877	1467	1424	1384
Semantic sessionization	907	923	692	421	807	783	844
(Static requests, SOA_S)	(58%)	(68%)	(62%)	(48%)	(55%)	(55%)	(61%)
Semantic sessionization	983	1051	939	683	1118	827	1105
(Dynamic requests, SOA_D)	(63%)	(77%)	(84%)	(78%)	(76%)	(58%)	(80%)
Common session boundary	47%	51%	49%	48%	57%	32%	74%

spelling or IP address) are removed. We compared two sessionizations based on time heuristics and semantics, with respect to the number of segmented sessions and the reasonability of association rules extracted from them. In case of semantic sessionization, the fields related with time such as “Timestamp” and “Elapsed Time” were filtered. Time-oriented heuristics simply sessionized the log entries between two sequential requests whose difference of field “Timestamp” is more than 20 milliseconds with respect to the same IP address. On the other hand, for ontology-oriented heuristics, the size of time window W was predefined as 50. The numbers of sessions generated in both cases are shown in Table 1.

Time-oriented heuristics estimate denser sessionization than two ontology-oriented approaches. It means that generally ontology-oriented heuristics based on SOA_S or SOA_D can make URLs requested over time gap semantically connected each other. They, SOA_S or SOA_D, decreased the number of sessions to, overall, 58.14% and 73.71%, respectively, compared to time-oriented heuristics. Even though ontology-oriented heuristics searched fewer sessions, the rate of common session boundaries (the number of common sessions matched with time-oriented heuristics over the number of sessions of SOA_D) is average 51.1%. It shows that more than 48% of sessions not segmented by time-oriented heuristics can be detected by semantic outlier analysis. While time oriented sessionization is impossible to recognize patterns of users who is easily changing their preferences or simultaneously trying to search various kinds of information on the Web, ontology-oriented method can discriminate these complicated patterns.

We also evaluated the reasonability of the rules extracted from three kinds of session sequences. According to the standard “least recently used (LRU),” we organized the expected set of URLs, which means the set of objects that cache server has to prefetch [Schechter et al. 1998]. The size of this set is constantly 100. As shown in Table 2, we measured the two hit ratios by both of their sessionizations for seven days.

Table 2: Evaluation of the reasonability of the extracted ruleset

	1	2	3	4	5	6	7
Time-oriented	0.06	0.32	0.46	0.41	0.51	0.52	0.49
Static requests, SOA_S	0.05	0.45	0.66	0.72	0.76	0.74	0.75
Dynamic requests, SOA_D	0.05	0.46	0.52	0.67	0.70	0.75	0.72

The maximum hit ratios in three sequences were obtained 0.52, 0.76, and 0.75, respectively. Semantic sessionization SOA_S acquired about 24.5% improvement of prefetching performance, compared with time-oriented. Moreover, we want to note that the difference between SOA_S and SOA_D. For the first three days, the hit ratio of SOA_S was higher than that of SOA_D by over 5%. Because of streaming data, SOA_D showed the difficulty in initializing the ruleset. After initialization step, however, the performances of SOA_S and SOA_D were converged into a same level.

6 Conclusions and future work

In order to mine useful and significant patterns from Web requests while browsing, many kinds of well-known association discovering methods have been developed. Due to the domain specific properties of streaming Web requests, sessionization process of the sequential entries is the most important in a whole step of discovering processes. We have proposed ontology-oriented heuristics for sessionizing Web requests. In order to provide each requested Website with the corresponding semantic information, Web directory as ontology have been applied to label this URL. Especially, we mentioned three practical problems for using real non-generic tree structured Web directories like Yahoo. After labeling URLs, we measured the semantic distance matrix indicating the relationships between URLs within the predefined time interval. Additionally, the factors like semantic mean and semantic deviation were formulated for dealing with on-line streaming Web requests. Therefore, two semantic outlier analysis approaches SOA_S and SOA_D were introduced based on semantic factors. Through the evaluation process, the detected candidate semantic outliers were tested whether their sessionization is reasonable or not. According to results of our experiments, investigating semantic relationships between Web requests is very important to sessionize them. Classifying semantic sessions, 48% of total sessions, brought about 25% higher prefetching performance, compared with time-oriented sessionization. Complex Web usage patterns seemed to be meaninglessly mixed along with "time" could be analyzed by ontology. Now, we note several important benefits, compared with simple heuristic-based sessionizations. Recognizing latent relationships between Websites makes the configuration of sessions more exact and reliable. It means the probability that Web requests during same intentions nearly occur in a same session is very high. Furthermore, a user's complicated browsing patterns can be efficiently discriminated, as shown in results of two experiments.

In future work, we want to evaluate our method by using the other measures proposed in [Spiliopoulou 2003]. We are considering the optimization scheme in order to minimize the error of sessionization. As exploiting semantic sessionization proposed in this paper to the Web proxy server, more practically, we will study association rule mining on various Web caching architecture, in order to improve the predictability of content prefetching.

References

[Agrawal et al. 1993] Agrawal, R., Imielinski, T., Swami, A.: "Mining association rules between sets of items in large databases"; Proc. ACM SIGMOD Int. Conf. Management of Data, Washington DC (May 1993), 207-216.

[Agrawal and Srikant 1994] Agrawal, R., Srikant, R.: "Fast algorithms for mining association rules"; Proc. 20^{th} Int. Conf. Very Large Data Bases, Santiago, Chile (Sept 1994), 487-499.

[Agrawal and Srikant 1995] Agrawal, R., Srikant, R.: "Mining sequential patterns"; Proc. 11^{th} Int. Conf. Data Engineering, Taipei, Taiwan (Mar 1995), 3-14.

[Batista and Silva 2001] Batista, P., Silva, M.J.: "Web access mining from an on-line newspaper logs"; Proc. 12^{th} Int. Meeting Euro Working Group on Decision Support Systems (2001)

[Berendt et al. 2002] Berendt, B., Mobasher, B., Nakagawa, M., Spiliopoulou, M.: "The impact of site structure and user environment on session reconstruction in web usage analysis"; Proc. 4^{th} WebKDD Workshop at ACM SIGKDD Conf. Knowledge Discovery in Databases, Edmonton, Canada (July 2002)

[Berendt et al. 2001] Berendt, B., Mobasher, B., Spiliopoulou, M., Wilsshire, J.: "Measuring the Accuracy of Sessionizers for Web Usage Analysis"; Proc. Workshop on Web Mining at The First SIAM Int. Conf. Data Mining, Chicago, USA (Apr 2001), 7-14.

[Berendt and Spiliopoulou 2000] Berendt, B., Spiliopoulou, M.: "Analysis of navigation behaviour in web sites integrating multiple information systems"; VLDB Journal, 9, 1 (Mar 2000), 56-75.

[Bielecki et al. 2002] Bielecki, M., Hidders, J., Paredaens, J., Tyszkiewicz, J., Bussche, J.; "Navigating with a browser"; Proc. 29^{th} Int. Colloquium on Automata, Languages, and Programming (2002), 764-775.

[Bonchi et al. 2001] Bonchi, F., Giannotti, F., Gozzi, C., Manco, G., Nanni, M., Pedreschi, D., Renso, C., Ruggieri, S.: "Web log data warehousing and mining for intelligent web caching"; Data and Knowledge Engineering, 39, 2 (Nov 2001) 165-189.

[Catledge and Pitkow 1995] Catledge, L., Pitkow, J.: "Characterizing browsing strategies on the world wide web"; Computer Networks and ISDN Systems, 27, 6 (1995), 1065-1073.

[Chen et al. 2002] Chen, Z., Tao, L., Wang, J., Wenyin, L., Ma, W.-Y.: "A unified framework for web link analysis"; Proc. 3^{rd} Int. Conf. Web Information Systems Engineering, Singapore (Dec 2002), 63-72.

[Cooley et al. 1999] Cooley, R., Mobasher, B., Srivastava, J.: "Data preparation for mining world wide web browsing patterns"; Knowledge and Information Systems, 1, 1 (Feb 1999), 5-32.

[Cooley et al. 1997] Cooley, R., Srivastava, J., Mobasher, B.: "Web mining: information and pattern discovery on the world wide web"; Proc. 9^{th} IEEE Int. Conf. Tools with Artificial Intelligence, Newport Beach, USA (Nov 1997), 558-567.

[Cooley et al. 2000] Cooley, R., Tan, P., Srivastava, J.: "Discovery of interesting usage patterns from Web data"; Lect. Notes in Artificial Intelligence, 1836, Springer, Berlin (August 2000), 163-182.

[Ding et al. 2002] Ding, C., He, X., Husbands, P., Zha, H., Simon, H.: "PageRank, HITS and a Unified Framework for Link Analysis"; Technical Report, 49372, Lawrence Berkeley National Laboratory (2002).

[Gruber 1993] Gruber, T.R.: "A translation approach to portable ontologies"; Knowledge Acquisition, 5, 2 (1993), 199-220.

[Jung 2005] Jung, J.J.: "Collaborative web browsing based on semantic extraction of user interests with bookmarks"; Journal of Universal Computer Sciences, 11, 2 (Feb 2005).

[Kleinberg 1999] Kleinberg, J.M.: "Authoritative sources in a hyperlinked environment"; Journal of the ACM, 46, 5 (1999), 604-632.

[Labrou and Finin 1999] Labrou Y., Finin, T.: "Yahoo! as an ontology: using Yahoo! categories to describe documents"; Proc. 8^{th} Int. Conf. Information Knowledge Management, Kansas, USA (Nov 1999), 180-187.

[Levenshtein 1966] Levenshtein, I.V.: "Binary codes capable of correcting deletions, insertions, and reversals"; Cybernetics and Control Theory, 10, 8 (1996), 707-710.

[Lieberman 1995] Lieberman, H.: "Letizia: an agent that assists web browsing"; Proc. 14^{th} Int. J. Conf. Artificial Intelligence (IJCAI-95), Montreal, Canada (Aug 1995), 924-929.

[McCallum et al. 1999] McCallum, A., Nigam, K., Rennie, J., Seymore, K.: "Building domain-specific search engines with machine learning techniques"; Proc. AAAI-99 Spring Symposium on Intelligent Agents in Cyberspace (1999).

[Mobasher 2000] Mobasher, B., Cooley, R., Srivastava, J.: "Automatic personalization based on web usage mining"; Communications of the ACM, 43, 8 (Aug 2000), 142-151.

[Pierrakos 2003] Pierrakos, D., Paliouras, G., Papatheodorou, C., Spyropoulos, C.D.: Web usage mining as a tool for personalization: a survey"; User Modeling and User-Adapted Interaction, 13, 4 (Nov 2003), 311-372.

[Schechter et al. 1998] Schechter, S., Krishnan, M., Smith, M.D.: "Using path profiles to predict HTTP requests"; Proc. 7^{th} Int. World Wide Web Conf., Brisbane, Australia (Apr 1998), 457-467.

[Spiliopoulou and Faulstich 1999] Spiliopoulou, M., Faulstich, L.C.: "WUM: a tool for web utilization analysis"; Lect. Notes in Comp. Sci. 1590, Springer, Berlin (1999), 184-203.

[Spiliopoulou 2003] Spiliopoulou, M., Mobasher, B., Berendt, B., Nakagawa, M.: "A framework for the evaluation of session reconstruction heuristics in web-usage analysis"; INFORMS Journal on Computing, 15, 2 (2003), 171-190.

[Srikant and Agrawal 1996] Srikant, R., Agrawal, R.: "Mining sequential patterns: generalization and performance improvements"; Proc. 5^{th} Int. Conf. Extending Database Technology, Avignon, France (Mar 1996), 3-17.

[W3C 1999] WWW Consortium: Web characterization terminology and definitions sheet working draft. (1999) http://www.w3.org/1999/05/WCA-terms/.

[Wang 1997] Wang, K.: "Discovering patterns from large and dynamic sequential data"; Journal of Intelligent Information Systems, 9, 1 (1997), 33-56.