Semantic Outlier Analysis for Sessionizing Web Logs

Jason J. Jung[1] and Geun-Sik Jo[1]

Intelligent E-Commerce Systems Laboratory,
School of Computer Engineering, Inha University,
253 Yonghyun-dong, Incheon, Korea 402-751
j2jung@intelligent.pe.kr, gsjo@inha.ac.kr

Abstract. As the web usage patterns from clients are getting more complex, simple sessionizations based on time and navigation-oriented heuristics have been restricted to exploit various kinds of rule discovering methods. In this paper, we present semantic session reconstruction based on semantic outliers from web log data. Above all, web directory service such as Yahoo is applied to enrich semantics to web logs, as categorizing them to all possible hierarchical paths. In order to detect the candidate set of session identifiers, semantic factors like semantic mean, deviation, and distance matrix are established. Eventually, each semantic session is obtained based on nested repetition of top-down partitioning and evaluation process. For experiment, we applied this ontology-oriented heuristics to sessionize the access log files for one week from IRCache. Compared with time-oriented heuristics, more than 48% of sessions were additionally detected by semantic outlier analysis. 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.

1 Introduction

As the concern for searching relevant information from the web has been exponentially increasing, the very large amount of log data have been generated in web servers. There are several types of web logs such as server access logs, referrer logs, agent logs, and client-side cookies. Thus, many applications have been focusing on various ways to analyze them in order to recognize the usage patterns of users and discover other meaningful patterns [2]. For example, on-line newspaper on the web [6], web caching [7], and supporting user web browsing [8], [20] can be told as the domains relevant to analyzing web log data.

In this paper, among the whole steps of web user profiling mentioned in [3], we have taken the session identification for segmenting web log data in consideration. There are mainly two kinds of sessionization heuristics for partitioning each user activity into sequences of entries corresponding to each user visit. First, time-oriented heuristics consider temporal boundaries such as a maximum session length or maximum time allowable for each pageview [4]. Second,

navigation-oriented heuristics such as HITS, PageRank, and DirectHit take the linkage between pages into account [5].

However, knowledge that is extractable from sessions identified by those heuristics is limited like frequent and sequential patterns represented by URLs. It means that web logs has to be sessionized with semantic enrichment based on ontology in order to find out more potential and meaningful information like a user's preference and intention. As shown in Fig. 1, blocks are requests from a user and their shapes are means the concepts related with requested URLs.

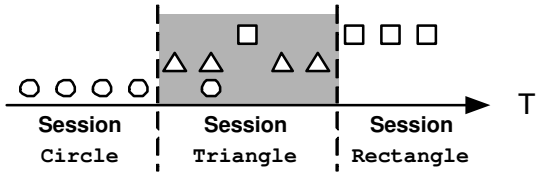

Fig. 1. Streaming Logs

More importantly, web caching(or proxy) servers have to track streaming URL requests from multiple clients, because they have to increase predictability for prefetching web content that is expected in next request.

Enriching web logs with their corresponding semantic information has been attempted in some studies [9], [20]. Typically, there are two kinds of approaches which are the *information extraction* and *information dimensions* mapping URLs to set of concepts as a feature vector and a specific value, respectively. While both of them apply keywords extracted from an URL's web page to perform semantic enrichment, we present conceptualizing an URL information itself by using web directory and introduce representing conceptualized URLs as tree-like information.

In this paper, we describe the conceptualization of web logs based on web directory, and then sessionize them by detecting semantic outliers. Thereby, in the following section, general types of web logs and their data models will be introduced. Sect. 3 will address how to handle url conceptualization against problems of web directory (Sect. 3.1) and how to determine semantic relationships between URLs for detecting semantic outliers (Sect. 3.3). We show semantic sessionization of the streaming web logs and verify our ontology-oriented heuristics approach in the Sect. 4 and Sect. 5, respectively. Finally, Sect. 6 makes a conclusion of this study and mention future work.

2 Data Model of Web Log and Problem Statement

There are several standard data models of web logs such as NCSA, W3C extended, and IRCache[1] format. In this paper we have focused on the IRCache format, which is the NLANR web caching project [1].

Table 1. The Data Format of IRCache Log Files

Fields	Description
Timestamp	The time when the client socket is closed. (millisecond)
Elapsed Time	The elapsed time of the request, in milliseconds.
Client Address	A random IP address identifying the client.
Log Tag	The Log Tag describing how the request was treated locally (hit, miss, etc)
HTTP Code	The HTTP status code
Size	The number of bytes written to the client.
Request Method	The HTTP request method.
URL	The requested URL.
User Ident	Always '-' for the IRCache logs.
Hierarchy Data and Hostname	A description of how and where the requested and Hostname object was fetched.
Content Type	The Content-type field from the HTTP reply.

Each IRCache log file consists of ten basic fields, as shown in the Table 1, and for instance, some records of a log file are as follows.

1048118411.784 214655 165.246.31.128 TCP_MISS/503 1568 GET http://www.intelligent.pe.kr/a.html - NONE/- text/html

1048118421.159 238 218.53.200.251 TCP_MISS/304 261 GET http://www.antara.co.id/images/spacer.gif - DIRECT/202.155.27.190 -

There are, generally, some problems to analyze these web logs such as their anonymity, rotating IP addresses connections through dynamic assignment of ISPs, missing references due to caching, and inability of servers to distinguish among different visits. Therefore, we note the problem statements concentrated for semantic sessionization in this paper, as follows.

- **Weakness of IP address field as session identifier.** The same IP address field in a web logs (within the time window or not) can not guarantee that those requests are caused by only one user, and reversely, requests from the different IPs can be generated by a particular user. As independent of temporal difference, the sequential or contiguous logs whose IP addresses are equivalent can be more partitioned or more agglomerated.
- **Simultaneous user requests based on multiple intention.** An user can request more than an URL "at the same time" by using more than a

[1] This project was administered by the University of California San Diego, in cooperation with the San Diego Supercomputer Center and the National Laboratory for Applied Network Research.

web browser. It means we have to consider multiple intention of users by classifying mixed logs according to the corresponding semantics.

Each request consists of timestamp, IP address, and URL fields, as shown in Table 2. URL field is divided into base url and reminder, which are the host name of web server and the rest part of full URL, respectively. Then, we assume that each URL is semantically characterized by its base URL.

Table 2. Example

Timestamp	IP Address	URL(BaseURL + Reminder)
$<t_1, t_2, t_3, t_4>$	ip_1	$<b\_url_1+r_1, b\_url_1+r_2, b\_url_1+r_3, b\_url_2+r_4>$
$<t_5>$	ip_2	$<b\_url_1+r_5>$
$<t_6, t_7, t_8>$	ip_1	$<b\_url_2+r_6, b\_url_1+r_7, b\_url_3+r_7>$

For example, we are given a web log composed of eight requests ordered by timestamps from t_1 to t_8. We denote the URL set of sequential requests by $< \ldots, b\_url_i+r_j, \ldots >$ mapped to the timestamps $< \ldots, t_i, \ldots >$. These logs are partitioned with respect to an IP address ip_i. After partitioning, we compare semantic distance between base URLs in a set of requests, because we regard a semantic session as the sequence of URL having similar semantics. In other words, we investigate if an user's intention is retained or not.

3 Ontology-Oriented Heuristics for Sessionization

In order to semantically recognize what an URL is about, we try to categorize a requested URL based on ontology. Such ontologies are applied with web directories. We therefore can retrieve the tree-like conceptualized URLs, and then measure the similarity between them.

Outlier, generally, is a data object which are glossly different from or inconsistent with the remaining set of data [15]. In order to detect outliers, cluster-based and distance-based algorithms have been introduced [17], [16], [18], [19].

Meanwhile, semantic outlier of streaming web log data in this paper means a particular URL request whose semantic distance with previous sequence logs is over prefined semantic threshold. Thereby, we define these semantic measurement such as semantic mean and semantic distance. Based on semantic outliers detected by these quantified values, semantic sessionization is conducted.

3.1 URL Conceptualization Based on Web Directories

An ontology, a so-called semantic categorizer, is an explicit specification of a conceptualization [10]. It means that ontologies can play a role of enriching semantic or structural information to unlabeled data. Web directories like Yahoo[2]

[2] Yahoo. `http://www.yahoo.com`

and Cora[3] can be used to describe the content of a document in a standard and universal way as ontology [11]. Besides, web directory is organized as a topic hierarchical structure which is an efficient way to organize, view, and explore large quantities of information that would, otherwise, be cumbersome [13].

In this paper we assume that all URLs can be categorized by a well-organized web directory service. There are, however, some practical obstacles to do that, 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 [12]. For example, some websites for the category Computer Science:Artificial Intelligence:Constraint Satisfaction:Laboratory can also be in the category Education:Universities:Korea:Inha University:Laboratory. We briefly note that problems with categorizing an URL with web directory as an ontology are the following:

- **The multi-attributes of an URL.** An URL can be involved in more than a category. The causal relationships between categories makes their hierarchical structure more complicated. As shown in Fig. 2 (1), an URL can be included in some other categories, named as A or B.

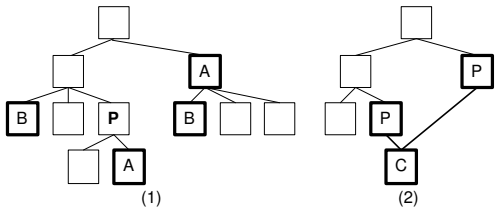

Fig. 2. (1) The multi-attribute of URLs; (2) The subordinate relationship between two categories

- **The relationship between categories.** A category can have more than a path from root node. As shown in Fig. 2 (2), the category C can be a subcategory of more than one like P. Furthermore, some categories can be semantically identical, even if they have different labels.
  - Redundancy between semantically identical categories
  - Subordination between semantically dependent categories

In order to simply handle these problems, we categorize each URL to all possible categories causally related with itself. Therefore, an URL url_i is categorized to a category set $Category(url_i)$, and the size of this category set depend on the web directory. Each element of a category set is represented as a path from the root to the corresponding category on web directory. In Table 2, let the base URLs {b_url$_1$, b_url$_2$, b_url$_3$} semantically enriched to {<a:b:d, a:b:f:k>,

[3] Cora. `http://cora.whizbang.com`

<a:c:h>, <a:b:f:j>}. The leftmost concept "a" is indicating the root of web directory and these base URLs are categorized to <d, k>, <h>, and <j>, respectively. In particular, due to multi-attribute of base URL b_url$_1$, $Category$(b_url$_1$) is composed of two different concepts.

3.2 Preliminary Notations and Definitions

We define semantic factors measuring the relationship between two log data. All possible categorical and ordered paths for the requested URL, above all, are obtained, after conceptualizing this URL by web directory. Firstly, the semantic distance is formulated for measuring the semantic difference between two URLs. Let an URL url_i categorized to the sets $\{path_i | path_i^m \in Category(url_i), m \in [1, \ldots, M]\}$ where M is the number of total categorical paths. As simply extending Levenshtein edit distance [21], the semantic distance $\Delta^\diamond$ between two URLs url_i and url_j is given by

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

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

$$D_{\Delta^\diamond}(i,j) = \begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & \Delta^\diamond[url_{t_i}, url_{t_j}] & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} \tag{2}$$

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

$$\mu^\diamond(t_1, \ldots, t_T) = \frac{2\sum_{i=1}^{T}\sum_{j=i}^{T} D_{\Delta^\diamond}(i,j)}{T(T-1)} \tag{3}$$

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

$$\sigma^\diamond(t_1, \ldots, t_T) = \sqrt{\frac{2\sum_{i=1}^{T}\sum_{j=i}^{T} \left(D_{\Delta^\diamond}(i,j) - \mu^\diamond(t_1, \ldots, t_T)\right)^2}{T(T-1)}} \tag{4}$$

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 pre-fixed threshold, with respect to given time interval.

3.3 Semantic Outlier Analysis for Sessionization

When we try to segment web log dataset, log entries are generally time-varying, more properly, streaming. In case of streaming dataset, not only semantic factors in a given interval but also the distribution of the semantic mean $\mu^{\diamond}$ is needed for sessionization. This will be described in the Sect. 4. We, hence, simply assume that a given dataset is time-invariant and its size is fixed in this section.

In order to analyze semantic outlier for sessionization, we regard the minimize the sum of partial semantic deviation $\mu^{\diamond}$ for each session as the most optimal partitioning of given dataset. Thereby, the principle session identifiers $PSI = \{psi_a | a \in [1, \ldots, S-1], psi_a \in [1, \ldots, T-1]\}$ 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

$$SOA_S(PSI) = \sum_{i=1}^{S} \mu_i^{\diamond} \tag{5}$$

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:

$$\Delta_{MAX}^{\diamond}[T_a, T_b] = \arg \max_{i=1,j=1}^{T} D_{\Delta^{\diamond}}(i, j) \tag{6}$$

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 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 cadidate 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 oversessionization, which is a failure caused by overfitting sessionization, detected by the evaluation process $SOA_S(PSI)$.

As an example, let a URL entry composed of two sessions S_A and S_B in two cases, as shown in Fig. 3. We assume that the semantic distances between $\mathrm{A}^{(i)}$s (or $\mathrm{B}^{(i)}$s) is much less than between each other.

In the first case (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 cadidate psi and prove this sessionization to be validate.

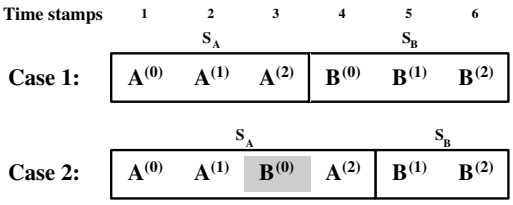

Fig. 3. Top-down approaching sessionization. The four largest semantic distances $\Delta^{\diamond}[A^{(1)}, B^{(1)}]$, $\Delta^{\diamond}[A^{(2)}, B^{(2)}]$, $\Delta^{\diamond}[A^{(2)}, B^{(0)}]$, and $\Delta^{\diamond}[A^{(2)}, B^{(1)}]$ are 0.86, 0.85, 0.81, and 0.79, respectively. We want to segement them into two sessions.

More complicatedly, in second case (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.

According to the required number of sessions, this recursion process can be executed.

4 Session Identification from Streaming Web Logs

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

$$SOA_D^{W^{(i)}}(PSI) = \sum_{k=1}^{S} \mu_k^{\diamond}|_{W^{(i)}} \tag{7}$$

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

$$\left|\mu^{\diamond}(W^{(i)}) - \mu^{\diamond}(W^{(i-\tau)})\right| \geq \varepsilon \tag{8}$$

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.

5 Experiments and Discussion

For experiments, we collected the sanitized access logs from sv.us.ircache.net, one of web cache servers of IRCache. These files were generated for seven days from 20 March 2003 to 26 March 2003. Raw data consist of 11 attributes, as shown in the Table 1, and about 9193000 entries. The total size of them is about 1.2 GB.

We verified sessionizing process proposed in this paper on a PC with a 1.2 GHz CPU clock rate, 256 MB main memory, and running FreeBSD 5.0. During data cleansing, logs whose URL field is ambiguous (wrong spelling or IP address) are removed, as referring to web directory.

We compared two sessionizations based on time oriented and ontology oriented heuristics, with respect to the number of segmented sessions and the reasonability of association rules extracted from them. In case of ontology-oriented sessionization, fields related with time such as "Timestamp" and "Elapsed Time" were filtered. Time-oriented heuristics simply sessionized 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.

Table 3. The number of sessions by time-oriented heuristics and ontology-oriented heuristics (static and dynamic logs) from logs for seven days (20-36 March 2003).

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

The numbers of sessions generated in both cases are shown in Table 3. Time-oriented heuristics esimate denser sessionization than two ontology-oriented approaches. It means that ontology-oriented heuristics based on SOA_S or SOA_D, generally, 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 extraced 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. The size of this set is constantly 100.

Table 4. Evaluation of the reasonability of the extracted ruleset (hit ratio (%))

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

As shown in Table 4, we measured the two hit ratios by both of their sessionizations for seven days. The maximum hit ratios in three sequences were obtained 0.52, 0.76, and 0.75, respectively. Ontology-oriented 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 association rules from web logs, many kinds of well-known association discoverying methods have been developed. Due to the domain specific properties of web logs, sessionization process of log entries is the most important in a whole step. We have proposed ontology-oriented heuristics for sessionizing web logs. In order to provide each requested URL with the corresponding semantics, web directory service as ontology have been applied to categorize this URL. Especially, we mentioned three practical problems for using real non-generic tree structured web directories like Yahoo. After conceptualizing URLs, we measured the semantic distance matrix indicating the relationships between URLs within the predefined time interval. Additionally, factors like semantic mean and semantic deviation were formulated for easier computation.

We considered two kinds of web logs which are stationary and streaming. Therefore, two semantic outlier analysis approaches SOA_S and SOA_D were introduced based on semantic factors. Through the evaluation process, the detected cadidate semantic outliers were tested whether their sessionization is reasonable or not. According to results of our expements, investigating semantic relationships between web logs is very important to sessionize them. Classifying semantic sessions, 48% of total sessions, brought about 25% higher prefetching performance, campared with time-oriented sessionization. Complex web usage patterns seemed to be meaninglessly mixed along with "time" can be analyzed by ontology.

In future work, we have to consider bottom-up approaching sessionization to cluster sessions divided by top-down approaching. 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 [14], in order to improve the predicability of content prefection.

Acknowledgement. This work was supported by INHA UNIVERSITY Research Grant. (INHA-30305)

References

1. IRCache Users Guide. http://www.ircache.net/
2. Cooley, R., Srivastava, J., Mobasher, B.: Web Mining: Information and Pattern Discovery on the World Wide Web. Proc. of the 9th IEEE Int. Conf. on Tools with Artificial Intelligence (1997)
3. Mobasher, B., Cooley, R., Srivastava, J.: Automatic personalization based on Web usage mining. Communications of the ACM **43**(8) (2000)
4. Berendt, B., Mobasher, B., Nakagawa, M., Spiliopoulou, M.: The Impact of Site Structure and User Environment on Session Reconstruction in Web Usage Analysis. Proc. of the 4th WebKDD Workshop at the ACM-SIGKDD Conf. on Knowledge Discovery in Databases (2002)
5. Chen, Z., Tao, L., Wang, J., Wenyin, L., Ma, W.-Y.: A Unified Framework for Web Link Analysis. Proc. of the 3rd Int. Conf. on Web Information Systems Engineering (2002) 63–72
6. Batista, P., Silva, M.J.: Web Access Mining from an On-line Newspaper Logs. Proc. 12th Int. Meeting of the Euro Working Group on Decision Support Systems (2001)
7. 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) (2001) 165–189
8. Cooley, R., Mobasher, B., Srivastava, J.: Data Preparation for Mining World Wide Web Browsing Patterns. Knowledge and Information Systems **1**(1) (1999) 5–32
9. Dai, H., Mobasher, B.: Using ontologies to discover domain-level web usage profiles. Proc. of the 2nd Semantic Web Mining Workshop at the PKDD 2002 (2002)
10. Gruber, T.: What is an Ontology? http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

11. Labrou, Y., Finin, T.: Yahoo! as an Ontology: Using Yahoo! Categories to Describe Documents. Proc. of the 8th Int. Conf. on Information Knowledge Management (1999) 180–187
12. Jung, J.J., Yoon, J.-S., Jo, G.-S.: Collaborative Information Filtering by Using Categorized Bookmarks on the Web. Proc. of the 14th Int. Conf. on Applications of Prolog (2001) 343–357
13. McCallum, A., Nigam, K., Rennie, J., Seymore, K.: Building Domain-Specific Search Engines with Machine Learning Techniques. AAAI Spring Symposium (1999)
14. Aggarwal, C., Wolf, J.L., Yu, P.S.: Caching on the World Wide Web. IEEE Tran. on Knowldge and Data Engineering **11**(1) (1999) 94–107
15. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers (2001)
16. Arning, A., Agrawal, R., Raghavan, P.: A Linear Model for Deviation Detection in Large Databases. Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, (1996) 164–169
17. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient Algorithms for Mining Outliers from Large Data Sets. Proc. of the ACM SIGMOD Conf. on Management of Data (2000) 427–438
18. Menasalvas, E., Millan, S., Pena, J.M., Hadjimichael, M., Marban, O.: Subsessions: a granular approach to click path analysis. Proc. of the IEEE Int. Conf. on Fuzzy Systems (2002) 878–883
19. Pei, J., Han, J., Mortazavi-Asl, B., Zhu, H.: Mining Access Patterns Efficiently from Web Logs. Proc. of Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'00) (2000)
20. Berendt, B., Spiliopoulou, M.: Analysing navigation behaviour in web sites integrating multiple information systems. The VLDB Journal **9**(1) (2000) 56–75
21. Levenshtein, I.V.: Binary Codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory, **10**(8) (1966) 707–710