Adaptive Weblog Post Filtering Based on User Browsing History

Ali Farahmand Nejad¹, Sadegh Kharazmi¹, Shahabedin Bayati², Hassan Abolhassani³, Koosha Golmohammadi⁴

¹Payame Noor University, ²University of Tarbiat Modares, ³Web Intelligence Lab, ⁴University of Alberta

1Department of Software Engineering, Payame Noor University, Tehran, Iran
2Department of Information Technology, University of Tarbiat Modares, Tehran, Iran
3Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
4Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada

afarahmand@pnu.ac.ir, kharazmi@pnu.ac.ir, sh.bayati@modares.ac.ir, abolhassani@sharif.edu, koosha@ece.ualberta.ca

Abstract

One of the most important Web-based services that established the foundations of the Web 2.0 is the weblog. Weblogs are evolving to be topic based systems that can lead to more revenue for companies. Therefore many companies provide free weblog hosting. Weblog popularity is an effective factor to gain more revenue. Weblogs have posts and topics that are arranged chronologically with the most recent post first. Some weblogs have so many posts that it makes finding a specific post very difficult. On the other hand irrelevant ordering of the posts makes it worse. Weblogs that do not have the posts in a proper order may result in decreasing the popularity of weblogs. Our experiments on Farsi weblogs have shown that many viewers close weblog windows before they are completely loaded in their Web browsers. This is due to a large number of posts on the weblogs.

Adaptive filtering of the posts on the weblogs can provide the readers with information that interests them. This paper introduces a new approach for filtering and reordering weblog posts based on user’s browsing history. Our experimental results show that our filtering approach can improve weblog popularity and increases the number of weblog viewers.

Introduction

The web is growing exponentially. Cuil search engine has indexed over 20 billion webpages [1]. Weblogs, wikis, multimedia sharing services, content syndication, podcasting and content tagging services are Web-based services that demonstrate the foundation of the Web 2.0 concept [2]. The concept “weblog” (blog) is a relatively well established Web-based service. The word “weblog” was coined by Jorn Barger, when he was the editor of the influential early weblog, Robot Wisdom [3]. A weblog is a website or page that is (generally) the product of an individual or of a non-commercial origin that uses a date-limited or diary format. Weblogs are updated either daily or at least regularly with new information about a subject, range of subjects, or personal details called posts arranged chronologically with the most recent post first [4]. The option for readers to leave comments in an interactive form is an important part of many weblogs [5]. Weblogs are indispensable for WWW as they have become extremely popular among Web users. The blog search engine BlogScope tracked over 30.88 million blogs with 544.66 million posts on December 2008 [6]. These figures make many companies consider weblogs as a good source of revenue. Many companies provide free weblog hosting and publishing tools with various features such as simple customization, multi-lingual posting, and video and photo blogging.

One of the main reasons that many companies focus on weblogs is making a profit from the popularity of blogs. Blogs are an increasingly attractive advertising platform. Many marketing managers believe that bloggers are creating high quality content and increasingly attract dependable audience. For example Technorati included the annual estimated revenue from advertising on their registered blog in its annual report [7]. They reported that the top 10 percent of blogger respondents earned an average of $19,000 annually.

Although weblog posts are in a chronological order, finding specific posts in a weblog is not trivial as there might be many posts on the startup page. This problem especially influences blog viewers with limited bandwidth and slow internet connection. Weblog posts may be ordered in other forms but still may not be effective for many blog viewers.

Our experiments show many blog viewers close weblog windows before they are completely loaded in their Web browsers. This is mainly due to a large number of posts on the weblogs.

We studied weblog accesses on PersianBlog which is the most widely used Persian weblog hosting service, with 582554 registered blogs [8]. We selected 160 popular weblogs randomly and asked 50 first year students to participate in this experiment as blog readers. Then students fill a set of predefined result forms. In this experiment 45% of students close 30% of weblog windows
before the pages are completely loaded into their browsers. 72.2% of students stated limited bandwidth and large number of blog posts as the reason for closing the blogs before they fully load. They prefer to use a simple website with the same content rather than a weblog. We defined the user’s tolerance threshold as the average time the users waited before closing the weblogs. We found out that the blogs with a loading time higher that the tolerance threshold, are the blogs that were closed the most by the users (see Figure 1.).

![Figure 1. Logarithmic view of weblog loading time in comparison with user’s tolerance threshold](image)

The popularity of a weblog can be improved by decreasing the number of posts in the main page of the blog without losing the general appeal of the contents. Furthermore reordering the blog posts may attract more blog viewers and increase the popularity of a blog due to quicker loading times and easier to navigate content.

We believe that adaptive filtering of posts can highly improve popularity of weblogs. Adaptive filtering is based on the user’s interests. We identify the user interests from his browsing history (similar to Google web history). We analyze the user’s browsing history to extract his/her preferred topics. Then we use this information for filtering the weblog posts. In this paper we introduce our method for filtering and reordering weblog posts based on a user’s browsing history on WWW.

### Related Work

Most research works on weblog filtering mechanisms focus on spam blogs (splog) detection. Kolari et al. [9, 10] described experimental results of blog identification using SVM and proposed a new splog mechanism. In another work Kolari et al. [11] characterized splogs by comparing them against trusted blogs. Salvetti et al. [12] proposed an effective and simple splog detection approach based on analyzing URLs. Their technique reached an accuracy of 78% versus human filtering with 76%. There are many research works about Web content filtering such as, artificial neural networks for content filtering [13]; Early Decision algorithm [14]; text classification approach for classification of webpages to desirable and undesirable ones [15].

### Filtering-Reordering Method

The proposed Filtering-Reordering method includes two main parts: first part is called miner unit. It logs, analyzes, and extracts user preferences from user browsing history. The second part is called filtering-reordering unit which performs weblog post filtering and reordering based on the user preferences identified in the miner unit. Each time the user requests a weblog the filtering unit activates.

Suppose that $W$ is a directed hypertext graph as $W = (P, L)$ in which nodes are physically distributed. $W$ is the simplified form of WWW that contains two pairs, where $P$ is the finite set of webpages, and $L$ is set of directed links between two webpages. Also suppose that there exists the hierarchy like $C$ that includes $n$ multi-leveled webpage types (classes). $C$ is a predetermined tree-shaped hierarchical topic directory such as Netscape’s ODP or Yahoo directory. For each node like $c_i$ in the hierarchy and a webpage like $p$ in $P$, we have:

$$\forall p \in P \exists c_i \in C \; p \in c_i \; 0 \leq i \leq n$$

where the $p \in C$ is always true as there is no webpage that is not member of $C$. So, the hierarchy induces a hierarchical categorization of the Web documents. Blogosphere like $B$ is the set of all weblogs in $W$ that define as: $B = \{wb|wb \in B, B \subset W\}$.

Each weblog like $wb$ contains many posts. We define a post as quintuplet $Post = (T, C, D, W, M)$. The entries in quintuplet are $T$ as the post title, $C$ as the post content, $D$ as the post date and time, $W$ as the post writer or blogger, and $M$ as the post comments. Therefore we can define each blog like $wb$ as:

$$wb = (URL, P, Feed, \bigcup_{i=1}^{n} Posts)$$

where $URL$ is the address of weblog, $P$ represents the collection of weblog owner profiles, $Feed$ is the Web feed such as RSS, and fourth parameter specifies union of weblog posts. We can think of each post as an individual webpage because each weblog post usually has a permalink to enable direct access to entry. So we have:

$$\forall Post \in wb \exists C_i \in H \; Post \in C_i$$

Our filtering mechanism works based on these properties. We define user preferences as all classes of webpages that user views more than a threshold.

$$Pref_{user} = \{c_p \in C | (\forall p \in W) \land c_p \in c_i [|c_p| > \delta]\}$$

where $c_i$ is a collection of all webpage classes that a user visited. $c_p$ is a subset of $c_i$ that is visited more than a threshold like $\delta$. We can define a user’s preferences more precisely by defining it as a combination of webpage classes that the user visits and classes of queries during browsing the Web. Query classes can be determined with the classification of results. In this work we simply use only the first factor as preferences function. Note that,
ordered list of the user preferences:
We used a heuristic for reordering weblog posts utilizing
filtered based on his/her preferences.
URL, filtering mechanism is activated and posts are
user’s preferences each time the user requests a weblog
(browse that collects data during the user Web browsing.
Collected data is analyzed to extract user preferences
(classes of preferences). After obtaining and analyzing the
user’s preferences each time the user requests a weblog
post, filtering component is activated and posts are
filtered based on his/her preferences.
We used a heuristic for reordering weblog posts utilizing
ordered list of the user preferences:

\[(\forall c_i, c_j \in \text{Pref}_{\text{user}})[(c_i >_{\text{pref}} c_j) \leftrightarrow (|c_i| > |c_j| > \delta)]\]

This relation reveals that user preferences class with more
view has more precedence. So if \( \text{Pref}_{\text{user}} = \{c_1, c_2, ..., c_k\} \),
we can arrange its elements based on order property.
As previously mentioned, we think of webpage (also
weblog post) classes as a predetermined tree-shaped
hierarchy. This means that a class can be subset of other
classes in the higher levels in the hierarchy and, also, a
class can be a super-class of some subclasses. Weblog
servers can use this property to show the nearest relevant
weblog posts in case there is no class identical to the user
preferences for a weblog post. Each time the user requests
a weblog, inversely the server requests user preference
classes from the client. By sending these classes from the
client, the server processes user preference classes and
current weblog posts classes to one by one mapping
between them. Server filtering component must traverse
upward and downward in the Web class hierarchy to find
the nearest class if there is a class in user preferences that
does not have an identical instance in current weblog post
classes. This upward/downward traversing must be
performed no more than a threshold number of times.

On the other hand each weblog is dependent to a general
class in the hierarchy, and each weblog post is an element
of a weblog class. Therefore the filtering component must
not filter any of the weblog posts if the user has a
preference identical to the weblog class. This problem
occurs in case of ineffective specification of webpage
classes. A comprehensive class hierarchy and an effective
method for classification of weblog posts are necessary to
address this issue.

Simulation and Experimental Results
This section describes our experimental results on weblog
post filtering using user browsing history. We simulated a
system to collect user preferences, filter, and reorder
weblog posts.

Collecting User Preferences
We developed a simple Web browser to collect the user’s
browsing history. Each time the user requests a URL,
clicking a hyperlink, or posting information to the server,
the browser stores the information about webpages and
user interactions into a local repository. This information is
used to identify user preference classes. Each class is
derived based on the webpage content, surrounding text
near to each clicked hyperlink, hyperlink text, surrounding
text near to the forms that user posts to a server.

Creating Webpage Class Hierarchy
We can exploit Yahoo Directory or ODP hierarchy to
model hierarchy of webpage classes. We used a Web-
crawler to harvest the Web and set crawler seed with ODP
starter webpage. This needs some modifications in the
structure of web-crawler. ODP webpage class hierarchy
cannot be used in our experiments as we ran this method
on Farsi weblogs. Instead we created a hierarchy based on
multilevel clustering of weblog posts in our dataset. Each
cluster is named by expert users.

Test Dataset
We tested the proposed filtering method on a collection of
1000 Farsi weblogs extracted from the most popular Farsi
blogs that are registered under PersianBlog. We collected
the last twenty weblog posts (each post is in the form of a
webpage) for each blog. We collected 20000 weblog posts
in total for our experiments. We consider each post as a
bag of words. A given post is labeled with the class name
that appeared the most in that post.

Experimental Results
We asked 30 students to use our simple Web browser for a
month. After a month we analyzed the browsing histories
and collected user preferences for each user. We then
simulated weblog requesting and displaying as we select a
user and a weblog randomly and our software filters and
reorders posts based on the user preference classes. Table 1
shows five users in our experiment that we chose randomly
as examples to describe the results using our method. Each
cell represents filtering and non-filtering percentages and
some examples justifying non-filtered posts.

Conclusion
These days many companies provide free weblog hosting
services and publishing tools. Weblogs are evolving to be
topic-based systems and a good source of revenue for
companies. The large number of weblog posts and static
form of ordering them are two issues that may affect
weblogs’ popularity. In this paper we introduced a new
filtering and reordering method based on user browsing
history. We developed a simple Web browser to identify
the user preferences as they browse webpages. Our
experimental results confirm that the proposed method is
effective in filtering the weblog posts based on interesting topics for the user.

Table 1. Experimental results for five randomly selected users in our experiment with their preferences used to filter posts on four weblogs

<table>
<thead>
<tr>
<th>Random users</th>
<th>User Preferences</th>
<th>Weblog 1</th>
<th>Weblog 2</th>
<th>Weblog 3</th>
<th>Weblog 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Filtered</td>
<td>Non-filtered</td>
<td>Filtered</td>
<td>Non-filtered</td>
</tr>
<tr>
<td>user 1</td>
<td>Java Programming, Persian music, Multithread programming, Movie, Online games, free eBooks, …</td>
<td>95%</td>
<td>5%</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>user 2</td>
<td>Java Programming, News, HTML Help, Traveling, Software Engineering</td>
<td>85%</td>
<td>15% (contains 3 poems about traveling)</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>user 3</td>
<td>News, Movie, games, Furniture, Rap Music, …</td>
<td>80%</td>
<td>20% (2 rap poems)</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>user 4</td>
<td>Java Programming, Rock music, Wallpaper Pictures, Software Download, HTML, …</td>
<td>75%</td>
<td>25% (3 traveling poems and 2 rap poems)</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>user 5</td>
<td>Java Programming, Poem, Comic Arts, Fashion pictures, Football News, …</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
<td>100%</td>
</tr>
</tbody>
</table>

References