View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Scitech Research Journals

SCITECH
RESEARCH ORGANISATION|

Volume 2, Issue 1
February, 2015|

Journal of Information Sciences and Computing Technologies
www.scitecresearch.com

Pretreatment of web log files

Hanane EZZIKOURI[(1)], Mohammed ERRITALI[(2)], Mohamed OUKESSOU[(1)]
(1) LMACS laboratory,
(2)TIAD laboratory, Computer Sciences Department, Faculty of sciences and techniques
Sultan Moulay Slimane University
Beni-Mellal, BP: 523, Morocco
ezzikourihanane@gmail.com, m.erritali@usms.ma, ouk_mohamed@yahoo.fr

Abstract

The pretreatment of web data is often the most laborious and requires the most time, this due in particular to the lack of structuration and the large amount of noise present in the raw data. Pretreatment of Web log files is to clean and organize the data contained in these files to prepare them for future analysis. Web log files are often text type, an objective of the pretreatment step is to transfer the data in an easier to use environment (eg in a database).

In this paper we will start with the presentation of different formats of web log files, then we will present the different pretreatment methods that we used as cleaning of Web robots queries, removing queries relating to scripts (“.js”, “.css”, ”.swf”) , identifications of users, sessions and visits.

Keywords: Pretreatment; web log files; identifications of users; sessions; visits.

1. Introduction

The user behavior on a website triggers a sequence of queries that have a result which is the display of certain pages. The Information about these queries (including the names of the resources requested and responses from the Web server) are stored in a text file called a log file. These data are stored in a standardized manner so that it is possible to carry out analyzes.

The process of Web mining (figure 1), which was defined as the set of techniques designed to explore, process and analyze large masses of consecutive information activities on the Internet, has three main steps: data preprocessing, extraction of reasons of the use and the interpretation of results.

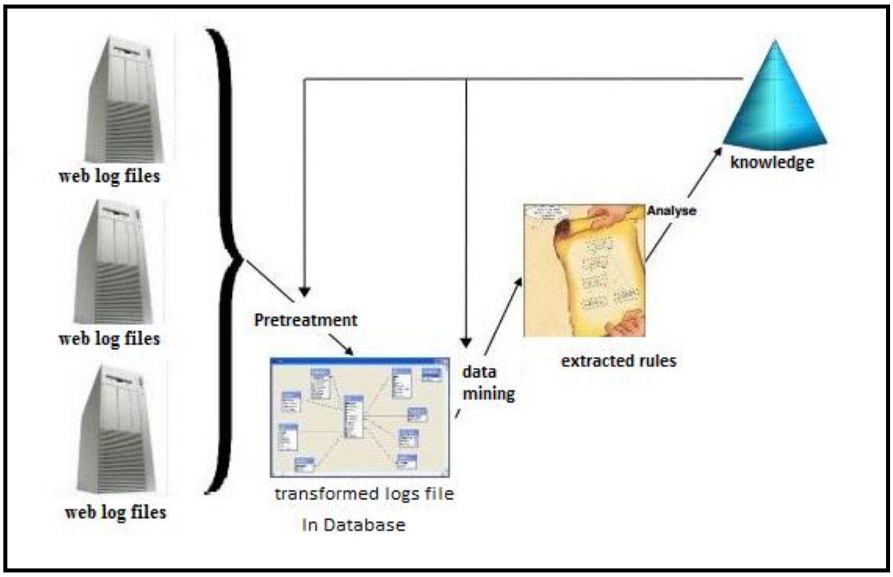
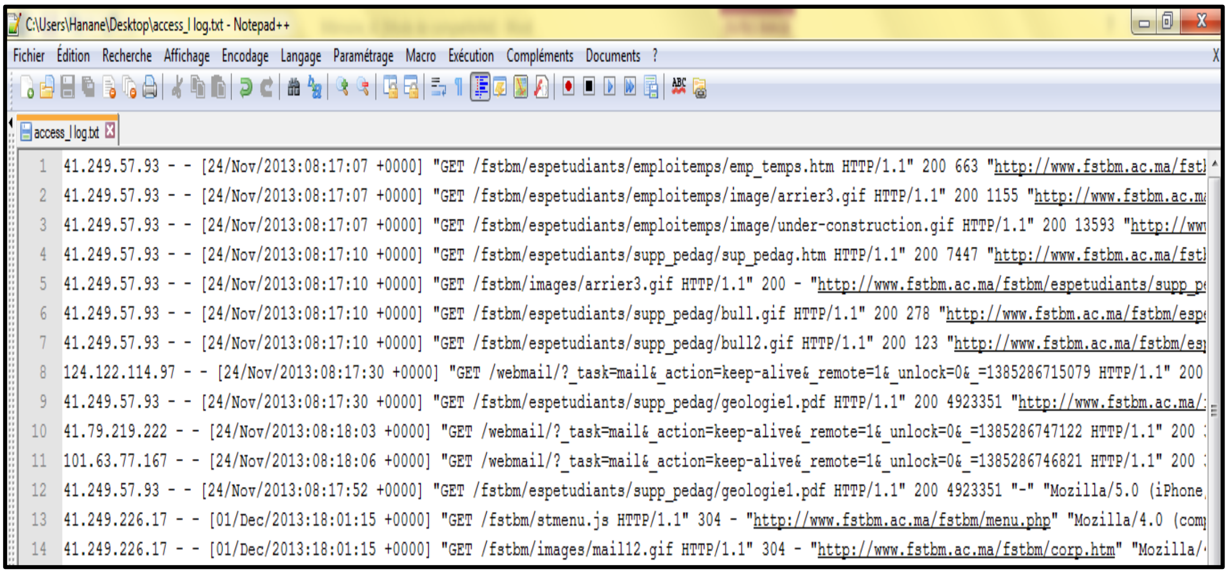


Fig. 1: Web usage mining process

The preprocessing of web data is often the most laborious and requires the most time, this due in particular to the lack of structuration and the large amount of noise present in the raw data. Pretreatment of Web log files is to clean and organize the data contained in these files to prepare them for future analysis. Web log files are often text type, an objective of the preprocessing step is to transfer the data in an easier to use environment (eg in a database). In this paper we will start with the presentation of different formats of web log files, then we will present the different pretreatment methods that we used as cleaning of Web robots queries, removing queries relating to scripts (".js", ".css", ".swf") , identifications of users, sessions and visits, as well as recording of this data in a relational database. our experiments have shown that our methodology allows a significant reduction (up to 70%) of the initial number of requests and offers richer structured logs for the next step of data mining.

2. Log file

A log file [8, 9, 10, 11] is a text file created and saved on the server of the website in an automatic way, it is a diary of connections in which a line is written in chronological order for each query and transaction processed by the server (change page, downloading a file ...), it is a source of data for global analyzes, the figure 2 below illustrates an example of a log file from the web server of the Faculty of sciences and techniques the Beni Mellal (FSTBM).



```
41.249.57.93 - - [24/Nov/2013:08:17:07 +0000] "GET /fstbm/espetudiants/emploitemps/emp_temps.htm HTTP/1.1" 200 663 "http://www.fstbm.ac.ma/fst
41.249.57.93 - - [24/Nov/2013:08:17:07 +0000] "GET /fstbm/espetudiants/emploitemps/image/arrier3.gif HTTP/1.1" 200 1155 "http://www.fstbm.ac.ma
41.249.57.93 - - [24/Nov/2013:08:17:07 +0000] "GET /fstbm/espetudiants/emploitemps/image/under-construction.gif HTTP/1.1" 200 13593 "http://ww
41.249.57.93 - - [24/Nov/2013:08:17:10 +0000] "GET /fstbm/espetudiants/supp_pedag/sup_pedag.htm HTTP/1.1" 200 7447 "http://www.fstbm.ac.ma/fst
41.249.57.93 - - [24/Nov/2013:08:17:10 +0000] "GET /fstbm/images/arrier3.gif HTTP/1.1" 200 - "http://www.fstbm.ac.ma/fstbm/espetudiants/supp_p
41.249.57.93 - - [24/Nov/2013:08:17:10 +0000] "GET /fstbm/espetudiants/supp_pedag/bull.gif HTTP/1.1" 200 278 "http://www.fstbm.ac.ma/fstbm/esp
41.249.57.93 - - [24/Nov/2013:08:17:10 +0000] "GET /fstbm/espetudiants/supp_pedag/bull2.gif HTTP/1.1" 200 123 "http://www.fstbm.ac.ma/fstbm/es
124.122.114.97 - - [24/Nov/2013:08:17:30 +0000] "GET /webmail/?_task=mail&_action=keep-alive&_remote=1&_unlock=0&_=1385286715079 HTTP/1.1" 200
41.249.57.93 - - [24/Nov/2013:08:17:30 +0000] "GET /fstbm/espetudiants/supp_pedag/geologie1.pdf HTTP/1.1" 200 4923351 "http://www.fstbm.ac.ma/
41.79.219.222 - - [24/Nov/2013:08:18:03 +0000] "GET /webmail/?_task=mail&_action=keep-alive&_remote=1&_unlock=0&_=1385286747122 HTTP/1.1" 200
101.63.77.167 - - [24/Nov/2013:08:18:06 +0000] "GET /webmail/?_task=mail&_action=keep-alive&_remote=1&_unlock=0&_=1385286746821 HTTP/1.1" 200
41.249.57.93 - - [24/Nov/2013:08:17:52 +0000] "GET /fstbm/espetudiants/supp_pedag/geologie1.pdf HTTP/1.1" 200 4923351 "-" "Mozilla/5.0 (iPhone
41.249.226.17 - - [01/Dec/2013:18:01:15 +0000] "GET /fstbm/stmenu.js HTTP/1.1" 304 - "http://www.fstbm.ac.ma/fstbm/menu.php" "Mozilla/4.0 (com
41.249.226.17 - - [01/Dec/2013:18:01:15 +0000] "GET /fstbm/images/mail12.gif HTTP/1.1" 304 - "http://www.fstbm.ac.ma/fstbm/corp.htm" "Mozilla/
```

Fig. 2: Excerpt from log file from the web server of the FSTBM

2.1 Location of the log file

The log files are located in three different places.

• Web Server: The log files provide information on the most accurate and complete data on the web server use. The log file does not save visited pages in cache. The log file data is sensitive information, thus the web server keeps them closed. [12]

• The client browser: The log file can reside in the client browser. HTTP cookies are used for the client browser, cookies are information generated by a Web server and stored in the computer of the user, for use in future access. [12]

2.2 Type of file Logs

All operations performed by the server are stored in log files that provide a detailed record of server activity. The logs can be stored in different file types [12]:

• Access logs: Data of all incoming requests, and information on the server clients. Access log files; record all requests that are processed by the server.

• Error logs: it keeps track of incidents in the dialogue with the server (eg wrong URL, interrupted transfer ...);

• Referential logs: it indicates the site and the page of origin and arrival;

• Agent logs: it caches information about user equipment (eg characteristics of the browser, operating system ... etc.).

The log file is a simple text file that stores information about each user. Log files can be in three different formats:

• W3C extended format (Extended log file format)

• NCSA common format

• IIS log file format

The data recorded in both formats of the NCSA log file and IIS are fixed while the W3C format allows the user to select the properties that you want to record for each request.

2.3 Format of log files

A. The format of the w3c log file

The format of the W3C log file is the format by default for IIS log file. This is most the format commonly used because it is flexible and allows you to store more information than other formats (that is to say that you can specify the information that you want to record). [12]

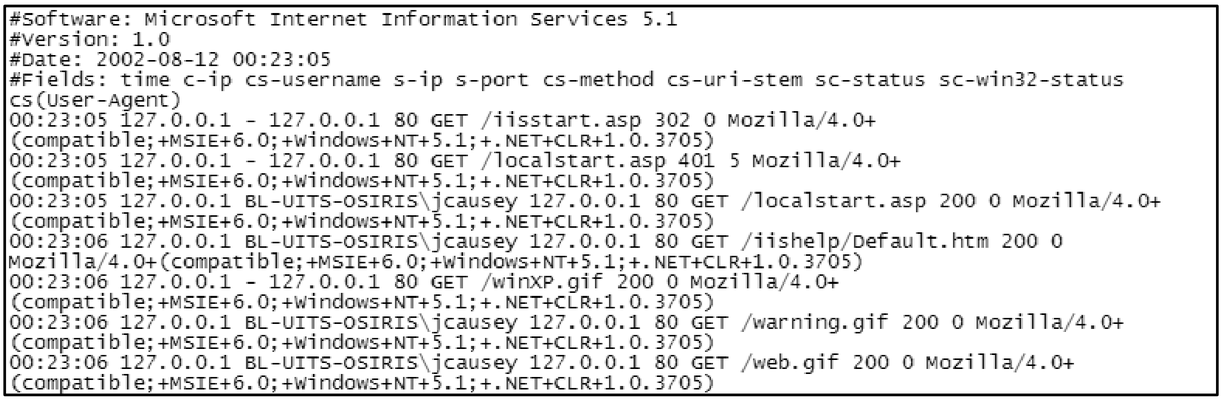
Figure 3 shows an excerpt of the log file that contains the following fields:

# Software - version of IIS running

# Version - the format of the log file

# Date - date and time of the first log record.

# Fields (explained in Table 1):: date time c-ip cs-username s-ip cs-method cs-uristem cs-uri-query sc-status sc-bytes cs-bytes time-taken csversion cs (User-Agent) cs (Cookie) cs (Referent).



```
#Software: Microsoft Internet Information Services 5.1
#Version: 1.0
#Date: 2002-08-12 00:23:05
#Fields: time c-ip cs-username s-ip s-port cs-method cs-uri-stem sc-status sc-win32-status
cs(User-Agent)
00:23:05 127.0.0.1 - 127.0.0.1 80 GET /iisstart.asp 302 0 Mozilla/4.0+
(compatible;+MSIE+6.0;+Windows+NT+5.1;+.NET+CLR+1.0.3705)
00:23:05 127.0.0.1 - 127.0.0.1 80 GET /localstart.asp 401 5 Mozilla/4.0+
(compatible;+MSIE+6.0;+Windows+NT+5.1;+.NET+CLR+1.0.3705)
00:23:05 127.0.0.1 BL-UITS-OSIRIS\jcausey 127.0.0.1 80 GET /localstart.asp 200 0 Mozilla/4.0+
(compatible;+MSIE+6.0;+Windows+NT+5.1;+.NET+CLR+1.0.3705)
00:23:06 127.0.0.1 BL-UITS-OSIRIS\jcausey 127.0.0.1 80 GET /iishelp/Default.htm 200 0
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+.NET+CLR+1.0.3705)
00:23:06 127.0.0.1 - 127.0.0.1 80 GET /winXP.gif 200 0 Mozilla/4.0+
(compatible;+MSIE+6.0;+Windows+NT+5.1;+.NET+CLR+1.0.3705)
00:23:06 127.0.0.1 BL-UITS-OSIRIS\jcausey 127.0.0.1 80 GET /warning.gif 200 0 Mozilla/4.0+
(compatible;+MSIE+6.0;+Windows+NT+5.1;+.NET+CLR+1.0.3705)
00:23:06 127.0.0.1 BL-UITS-OSIRIS\jcausey 127.0.0.1 80 GET /web.gif 200 0 Mozilla/4.0+
(compatible;+MSIE+6.0;+Windows+NT+5.1;+.NET+CLR+1.0.3705)
```

Fig. 3: Example of W3C log file format

B. The NCSA Common Log File Format

NCSA format [9] [10] [12] is based on a fixed ASCII text format, so you cannot customize it. This is a smaller version of W3C, In this type if no directive specifies a different format, access logs are recorded in CLF (Common Log Format). The NCSA Common format is available for websites and SMTP and NNTP services, but it is not available for FTP sites. The NCSA common log file format records the following data:

1. The domain name or Internet Protocol address (IP) of the calling machine

2. The name and the HTTP user login (in case of access with a password)

3. The date and time of the request,

4. The method used in the request (GET, POST,…) and the name of the requested resource (the URL of the requested page).

5. The status of the request ie the query result (success, failure, error,...)

6. The size in bytes of the requested page.

7. The browser and operating system used by the client.

A line from a log CLF is presented below:

```
116.203.228.15 - - [01/Dec/2013:18:02:24 +0000] "GET /webmail/?_task=mail&_action=keep-
alive&_remote=1&_unlock=0&_=1381425498988 HTTP/1.1" 200 33
"http://www.fstbm.ac.ma/webmail/?_task=mail&_id=1877860343529b4ac71ed12&_action=compose"
"Mozilla/5.0 (Windows NT 6.1; rv:24.0) Gecko/20100101 Firefox/24.0"
```

Fig. 4: example of CLF line

C. The IIS log file format

The IIS log file format is based on the fixed ASCII text format, so you cannot customize it. Because HTTP.sys manages the IIS log file format. The IIS log file format records the following data:

- The IP address of the client
- The User Name
- Date
- Time
- Service and instance
- Server Name
- Server IP Address
- The time taken
- Bytes sent by the client
- Bytes sent by the server
- State Service Code (A value of 200 indicates that the request was successfully executed.)
- Windows status code (A value of 0 indicates that the request was completed successfully.)
- Type of application
- Purpose of the transaction

A line of an IIS log is shown below:

```
172.16.255.255, anonymous, 03/20/01, 23:58:11, MSFTPSVC, SALES1, 172.16.255.255, 60, 275,
0, 0, 0, PASS, /Intro.htm
```

Fig. 5: example of IIS log file format

2.4 The fields of log files

The various fields in a log file used are described in Table 1; some fields may or may not exist according to the format:

Field	The name in the log file	Description
Date	Date	The date on which the activity occurred.
Heure	Time	The time at which the activity occurred.
IP address of the client	c-ip	The IP address of the client that accessed to the server.
Nom d'utilisateur	c-username	The name of the authenticated user who accessed the server. The hyphen represents anonymous users.
Nom du service	s-sitename	Internet service that runs on the client computer and the computer instance number.
Nom du serveur	s-computername	The name of the server on which IIS 5.0 generated the log entry.
Adresse IP du serveur	s-ip	The IP address of the server on which IIS 5.0 generated the log entry.
Port du serveur	s-port	The port number to which the client is connected.
Méthode	cs-method	The action that the client was trying to perform (for example, a GET method).
Ressource URI	cs-uri-stem	The resource that the server accessed (eg Default.htm).
Requête URI	cs-uri-query	The request that the client was trying to perform.
État du protocole	sc-status	The status of the action in terms of HTTP protocol.
État Win32	sc-win32-status	The status of the action in terms Win2K.
Octets envoyés	sc-bytes	The number of bytes sent by the server.
Octets reçus	cs-bytes	The number of bytes received by the server.
Durée écoulée	time-taken	The duration of the action.
Version du protocole	cs-version	The version of the protocol (HTTP, FTP) client used. For HTTP, this value is HTTP 1.1 and HTTP 1.0.
Hôte	cs-host	The name of the IIS 5.0 server.
Agent utilisateur	cs(User-Agent)	The browser used by the customer.
Cookie	cs(Cookie)	The content of any sent or received cookie.
Référant	cs(Referer)	The previous site that brought the user to the current site.

Table 1: Description of fields in a log file

Firstly it should be noted that the lines come in chronological order as different requests not grouped by visitor. Each line has a well defined format.

The line in Figure 4, Will serve as an example to comment on the various blocks of data.

- **116.203.228.15**: The first set of numbers is the Internet Protocol or IP address. This address is unique during a connection.
- **[01/Dec/2013:18:02:24 +0000]:** the date, time and time zone of the query.
- **GET / webmail /:** the query. Here the required page is webmail.php.

The requests generally used are [9] [10]: GET, HEAD, PUT, POST, TRACE and OPTIONS:

method	Explanation
GET	Information request. The server processes the request and returns the contents of the object.
HEAD	very similar to the GET method. However, the server returns only the header of the requested resource without the data. There is no message body.
PUT	allows to download a document whose name is specified in the URI, or delete a document, always if the server allows it.
POST	used to send data to the server.
TRACE	used for debugging. The server returns in the body of the response, the exact content it received from the client. This allows us to understand, in particular, what happens when the request passes through several servers.
OPTIONS	Allows to request the server, methods allowed for document reference

Table 2: Methods used in queries

- **HTTP / 1.1:** is the protocol used.
- **200:** data about status of requested page (200 to "available", 404 for "not found" ...).

Indeed, the status code, integer coded on three numbers, has a specific meaning in classes depends on the first digit:

- 1xx indicates only an informal message.
- 2xx indicates success.
- 3xx redirects the client to another URL.
- 4xx indicates a client-side error.
- 5xx indicates a server-side error.

- **33**: is the charged size.
- **http://www.fstbm.ac.ma/webmail/:** the reference page, the page from which the query is run.
- **Mozilla / 5.0 (Windows NT 6.1; rv: 24.0) Gecko / 20100101 Firefox / 24.0:** The last data block shows information about user configuration. Here the visitor uses the Mozilla browser on a Windows NT 5.0 environment.

Code	Description	Code	Description
101	switching protocol	409	conflicts
200	Success	410	Gone
201	Created	403	forbidden
202	Accepted	404	not found
203	Non-Authoritative Information	405	Method not allowed
204	No Content	406	not acceptable
205	Reset of Content	411	length required
206	partial contents	412	Condition failed
300	multiple choice	413	Excessive request entity
302	Found	414	Request-URI too long
307	temporary redirect	501	not implemented
400	invalid request	502	Incorrect Gateway
401	not allowed	503	Service Unavailable
408	Timeout expired		

Table 3: Description of status codes

The data stored in the web log file has a large amount of erroneous, misleading and incomplete information. Pretreatment which is one of the important and complex stages of Web usage mining WUM is necessary to convert a log in a data set that is adapted to the analyzes. In the next section we will present this step.

3. Preprocessing

The WUM three main steps: Preprocessing, extraction patterns (or rules) and analysis (interpretation) patterns.

The first stage of WUM process (figure 6), which is obviously Preprocessing, mainly consists of two types of tasks and occupies about 60% to 80% of the time involved in the whole process [9] [10].

- **Classical preprocessing tasks:**
- Fusion of web log files,
- Storage structured data in a database
- Cleaning and structuring of data.
- **Advanced preprocessing tasks:**
- The creation of a set of initial data, in which the algorithms of data mining can be applied.

At the end of the preprocessing step the initial data are cleaned and structured, in general, in a database.

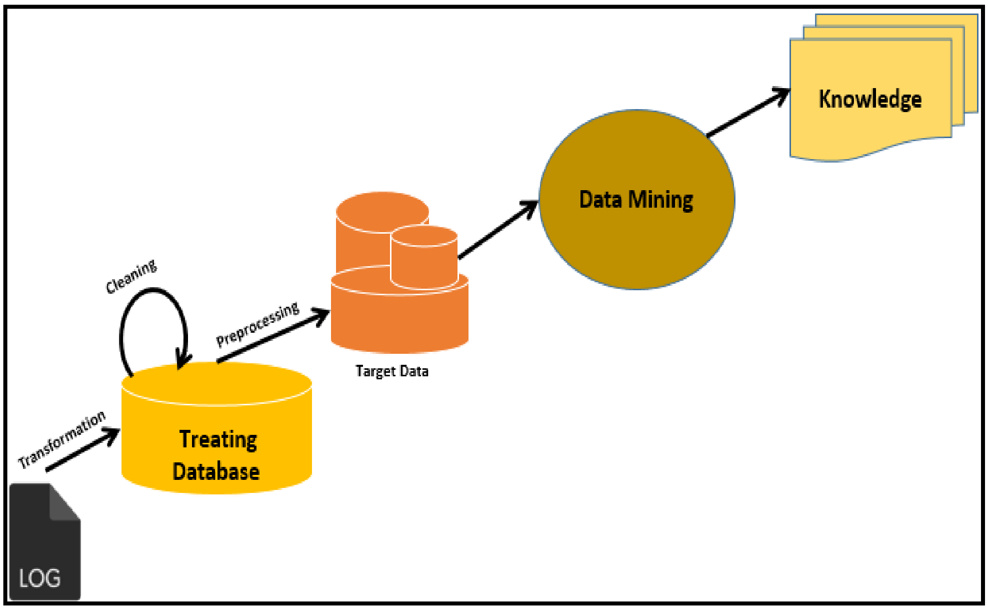


Fig. 6: Web Usage Mining process

3.1Problems specific to file data LOGS

Although the data provided by the Logs files are useful, it is important to consider the inherent limitations of these data in their analysis and interpretation. Some of the challenges that can occur [9] [10]:

❖ Useless requests

Each time the server receives a request, it records a line in the log file. Thus, to load a page, there will be as many lines in the file as the numbers of objects contained on this page (graphics). Pretreatment is essential to remove unnecessary queries.

❖ Firewalls

These network access protections mask the IP addresses of users. Any connection request from a server with such protection will have the same address, regardless of which user is. So it is impossible, in this case, identify and distinguish visitors from the network.

❖ Web Caching

To ease traffic on the Web, a copy of some pages is saved in the local browser of the user or at the proxy server in order to not download them every time a user requests them. In this case, a page can be accessed several times without any access to the server. As a result, the corresponding requests are not recorded in the log file [9] [10].

❖ Use of robots

Web yearbooks, known as search engines use robots that travel all the websites to update their search index. In doing so, they trigger queries that are stored in all log files for different sites, distorting their statistics.

❖ Users Identification

The identification of users from the log file is not a simple task. In fact, using the log file, the unique identifier available is the IP address and the "agent" of the user. This identifier has several limitations:

- Single IP Address / Multiple server sessions:

Same IP address can be assigned to multiple users accessing Web services through a single proxy server.

- Multiple IP addresses / single User:

A user can access the web from multiple machines.

- Several agents / single User:

A user, who uses more than one browser, even if the machine is unique, realized as multiple users.

❖ Sessions identification

All requests from a user identified constitute its session. The beginning of a session is defined by the provenance of user to the site. However, no signal indicates the disconnection from the site and consequently the end of a session.

❖ Lack of information

Log file does not bring anything about the behavior of the user between queries: Is it really reading the page displayed? In addition, the number of visits of a page does not necessarily reflect the interest of it. In fact a high number of visits may simply be attributed to the organization of a site and the forced passage of a visitor in others.

The method of preprocessing, we present consists of nine separate stages, six for cleaning in a so-called **classical preprocessing** and three for processing data in an **advanced preprocessing**.

3.2. Classical preprocessing

3.2.1. Merge log files

To reduce the load on a particular server, many servers are used; a user can come from multiple servers or Web application.

Even before starting the cleaning process, it must merge the different log files. [11] The fusion of log files of several web servers refers to the fusion of all the data they contain the queries of all log files were put together in one file.

3.2.2. Cleaning data

The format of web log files is unfit for direct analysis by various data mining techniques. Before being able applying these techniques we should clean the log file. [13]

Data cleaning for Web log files consists in remove any queries considered unnecessary. [9] [10]. The web server allows disposing of all types of resources: web page, multimedia item, and program. During a request corresponding to a page including other resources (usually images, animations), the client runs multiple requests to the server: one for the page (the container) and one for the various elements. Thus, for a requested page, multiple queries may lead to server [13].

Referring to the goal of our work, namely extraction valid models and knowledge of web services, we feel it wise to retain only web pages (called container) without the contents. We simplify the logs by removing anything that does not match to a web page.

For Web portals, and popular websites the size of Web log files is counted in gigabytes per hour. Even with the systems and software nowadays manipulate files with such dimensions, gets very complicated. By filtering data, we gain not only disk space, but at the same time, it makes more efficient the following tasks in the process of WUM, Some experiments have shown that this cleaning stage reduces the size of the original log file 40% to 70% [14].

3.2.3 Cleaning requests for images and multimedia files

Cleaning images is to delete files with extensions: .jpg, .gif, .png, etc ... And the media files with the extension: .wav, .wma, .wmv, etc ... However, it is not always possible to identify all the less interesting pictures when the site is large, in some cases, these images are not included in HTML files. There may be an image that requires clicking on a link to view it. In such cases we must maintain the request for this picture in the log file as it indicates a user action [11,13, 14].

3.2.4 Deleting requests from Web robots

This type of query, not only useless, but making noise in the data and do not reflect the behavior of the user [13].

Web robots (WRs) are software programs used to scan a Web site to extract its contents. They automatically follow all links on a Web page. Search engines like Google, bing, yahoo ... regularly send their robots to extract all the pages of a Web site to update their search index. The number of applications of a WR is usually greater than the number of requests for a normal user.

Usually a WR is identified by using the "User Agent" field in the log files. However, today it is almost impossible to know all the agents that represent a WR because every day appears new WRs and this makes this task very difficult.

We have used heuristics to identify queries from WRs, the first three were used by Tanasa et al. [4], considering that it is sufficient to check one of these heuristics to consider the corresponding query as being generated by a Web robot:

- Identify the IP and the "User-Agent" addresses known as web robots. This information is usually provided by the search engines,

- Identify the IP addresses that have made a request to the "\ robots.txt"

- Using a threshold for the browsing speed BS "Browsing Speed" equals the number of page views per second. Calculating Speed Browsing is only possible after the determination of the sessions.

- Identify the "User-Agent" with the following key words: "crawler", "spider" or "bot".

- Identify the requests made by aspirators of Web sites (HTTrack for example), or by some browsers modules for offline consultation, such as DigExt of Internet Explorer. For aspirators who hide their User-Agents, their identification is performed based on the duration of their applications, typically zero.

Once all requests from WRs have been identified, we can proceed to its removal [11,13,14].

2.3.5 Deleting queries scripts

Usually, downloading a user-requested page is accompanied by automatic download of scripts such as Java scripts (.js files), style sheets (.css files), flash animations (.swf file) .... These elements must be removed from the log file given that their appearance does not reflect the behavior of the user.

2.3.6 Removing invalid queries

In the learning phase we keep only the queries that resulted, ie corresponding to a valid resource. The return code in the logs allows us to filter these requests. Therefore, we keep only those codes between "200 and 399".

2.3.7 Suppression of methods different of "GET"

Since the WUM is interested in the study of the behavior of the user on the Web and therefore the resources it requires, we just keep the queries that the method is GET [11, 13, 14].

2.9. Resolution of several problems

A. Single Page interpreted differently by the shell of the server:

For extension pages ".php", it was necessary to solve the problem of the single page. So the idea was to change the name field of the requested resource (the URL of the requested page) so removed for a same page of his url beginning with '?' To the end the url.

For example:/webmail/||?task=mail&_action=keeplive&_remote=1&_unlock=0&_=1381425498988||.

B. Date & Time Format

We also had transformed the format of the date and time, as the format that the logfile uses to keep track of the user's input time is incompatible with that used in standard "date" and "datetime" databases.

For example: 01 / Dec / 2013: 18: 02: 24 0000

3.3. Advanced preprocessing: Data Transformation

In the data processing stage, we structured data stored in a persistent form, usually in BD [13], and we identified users, grouped queries sessions. And divided visits sessions (Figure 10) by choosing a Δt = 30min.

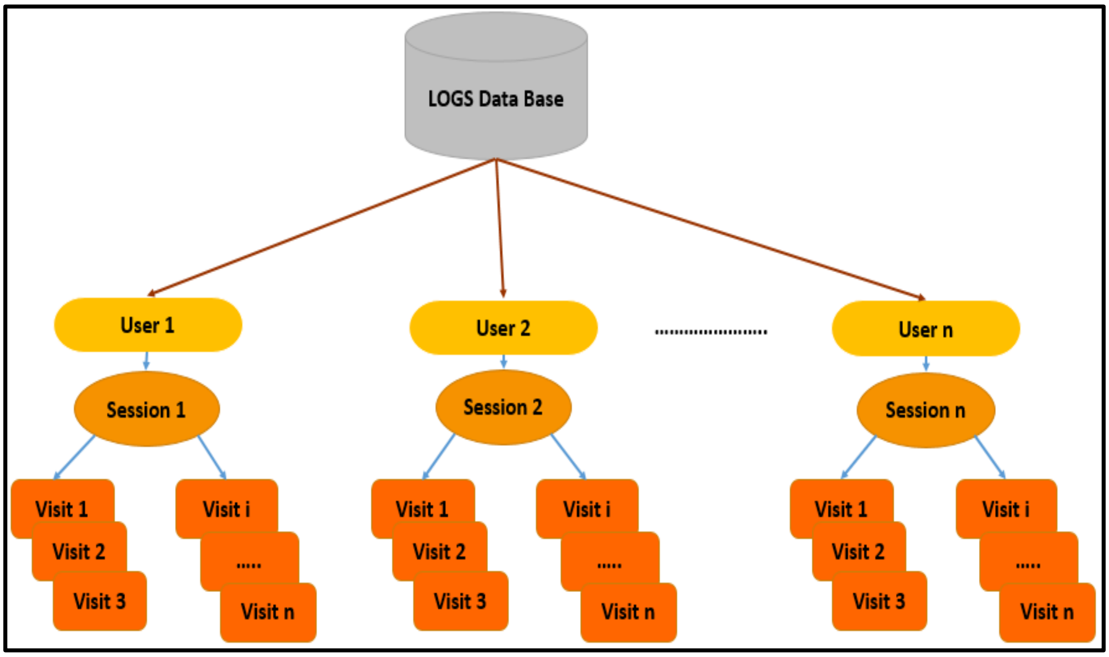


Fig. 7: Process of transformation data from log files

3.3.1 Identification of users:

For grouping requests, it is necessary to see which users issued them. The identification of users from the log file is not an easy task due to several factors such as: proxy servers, dynamic addresses, if users using the same computer (in a library, internet club, etc. .) or that of the same user using more than a Web browser, or more than one computer. In this case it is laborious to speak, user identification. But if the user has agreed to register and identify with a login, then the registration is immediate, but it concerns only a very small minority of visits. In fact, using the log file, there is only the IP address and the agent of the user. There are other methods that provide more information. The most used are: "cookies", web dynamic pages (with a session ID in the URL), registered users, modified browsers etc.

A. General form of user agents

This field contains information about the browser and operating system used by visitors when they visit a website [15,16]

User agents (UA) string is broken down into several components [17]:

- Browser: is the general sign that says the browser "X" is compatible, and is common to almost all modern browsers.

- Platform describes the native platform, the browser is running on (ex: Windows, Mac, Linux and Android) and whether or not it is a mobile phone. For example, the Firefox OS phones simply say "Mobile"; the web is the platform.

- Name Browser / browser version indicates the browser name browser (eg Firefox, Netscape ... etc.) And provides the version you browser rversion (such as "17.0" for firefox).

- rv: geckoversion indicates the version of Gecko (such as "17.0"). In recent browsers, geckoversion is the same version of browser.

- Gecko / geckotrail indicate that the browser is based on Gecko.

Note:

Gecko is software for presenting web pages. Launched by the Mozilla Foundation in 1998, open source and free, it is incorporated in various applications such as Netscape Navigator, Firefox, Thunderbird, AOL Explorer or Camino.

B. Identification algorithm

For the identification of the user, we use as shown in the figure 8, the combination of fields:

- IP address

- User Agent of a Web log.

We consider two requests from the same IP address but from two different user-agents belong to two different sessions so they are carried out by two different users.

However, we cannot deny the inherent limits to this method. In fact confusion between two different users using the same IP address and the same User-Agent is always possible especially when using a proxy server or firewall. [14]

```
Algorithm: user identification
Inputs : preprocessed web log file
Output : Identified users
For each record in the data set do
        1. If currentIP is not in ListOfIP Then
                add currentIP in ListOfIP
                mark it as a new user
                assign a new userID
        2. Else if currentOS if not in ListOfOS Then
                add currentOS in ListOfOS
                mark it as a new user
                assign a new userID
        3. Else if currentBrowser if not in ListOfBrowser Then
                add currentBrowser in ListOfBrowser
                mark it as a new user
                assign a new userID
        4. Else
        mark the current record with its existing userID
        end if
End for
END
```

Fig. 8: Users classification algorithm [12]

3.3.2 Identification of sessions:

A session consists of all page views by the same user during the analysis period. We used the pair (IP address, user agent) for the identification of the user. For the session of each user, we ordered the log file by the pair (Host, User Agent) and then by time.

The session start is defined by the arriving of the user (stored in the referrer URL) is external to the site. Contrariwise, no signal indicates the disconnection of the site, which is a problem to determine the end of the sessions.

In the literature the criteria proposed for the end of the sessions are in fact idle time thresholds ranging from 25-30 minutes to 24 hours.

We changed the user identification algorithm, to combine the identification of users and sessions; this algorithm has been improved as shown in Figure 9:

```
Algorithm: Identification user && sessions
Inputs : preprocessed web log file
Output : identified users and sessions
Pour chaque enregistrement de  l'ensemble de données faire
        1. . If currentIP is not in ListOfIP Then
                add currentIP in ListOfIP
                mark it as a new user and new session
                assign a new userID and a new sessionID
        2. Else if currentOS if not in ListOfOS Then
```

```
                        add currentOS in ListOfOS
                        mark it as a new user and a new session
                        assign a new userID and a new session
        3. **Else if** currentBrowser if not in ListOfBrowser **Then**
                        add currentBrowser in ListOfBrowser
                        mark it as a new user and a new session
        assign a new userID and a new sessionID
        4. **Else**
        **mark** the current record with its existing userID and sessionID
        **End if**
        **End for**
**END**
```

Fig. 9: Identification algorithm sessions (improved version)

2.4.3. Identification of visits

A visit consists of a series of sequentially ordered queries performed during the same session and having no break of more than 30 minutes sequence (based on empirical criteria Kimball). [14]. considering $\Delta t= 30$ minutes, widely used as a standard time threshold. It has been demonstrated also by [18] a value of 25.5 minutes is required to determine the limits of a visit.

A. Approach of Identification of visits

Identification of site visits, is performed using the following approach (Figure ??):

- Order the following basic variables "session ID" and time of the request.

- Determine the consultation period of the pages, which is the time between two http requests. So the consultation period of one page is calculated by the difference between the dates and times of successive recordings. Consequently only the consultation period of the last page of each session is unknown.

- If the consultation period of one page exceeds 30 minutes, then the next page in the same session is assigned to another visit (the user has spent more than 30 minutes to read the same page which is unlikely, or he left the site to return 30 minutes after).

- Once identified visits, the last page of each consultation period is obtained from the average of the consultation periods of the previous pages in the same visit.

B. Visits identification algorithm

```
**Algorithm:** User, Session & Visit Identification
**Input**      **:** processed weblog file
**Output**     **:** Identified User, Session & Visit.
**BEGIN**
 **For** each record in dataset **do**
                  1.  **If** currentIP is not in ListOfIP **Then**
                          add currentIP in ListOfIP
                          mark whole record as a new user and session
                          assign a new sessionID and userID
                  2.  **Else** if currentOS is not in ListOfOS **Then**
                          add currentOS in ListOfOS
                          mark whole record as a new user and session
                          assign a new sessionID and userID

                  3.  **Else** if currentBrowser is not in ListOfBrowser **Then**
```

```
                    add currentBrowser in ListOfBrowser
                    mark whole record as a new user and session
                    assign a new sessionID and userID
             4.  Else mark current record  with  existing sessionID and userID
                    End If
             5.  If User and Session are well identified (userID, sessionID)
                    if current record timestamp is more than 1800 seconds
                 #30minutes * 60 seconds
                          mark whole record as a new visit
                          assign a new visitID
          Else mark current record  with  existing visitID
           End If
        End If
End  For
END
```

Fig. 10: User, Session & Visit Identification algorithm

Note: A timestamp is a sequence of characters that contains sufficient information, usually a date and time to locate an event in time. It is displayed in a way that facilitates comparison with other timestamp.

4. Conclusion

In this paper, we initially presented the different formats of log files. Then we presented a method of data preprocessing with heuristics establishment that transform all queries registered in the Logs files cleaned and structured data ready for analysis by applying methods data mining. In fact the new database size (30% of original size) shows the importance of this stage of pretreatment of log files, especially the phase of cleaning.

References

[1] Cooley, R., Mobasher, B., & Srivastava, J. (1999). Data preparation for mining world wide web browsing patterns. Knowledge and information systems, 1(1), 5-32.

[2] Tan, P. N., & Kumar, V. (2004). Discovery of web robot sessions based on their navigational patterns. In Intelligent Technologies for Information Analysis (pp. 193-222). Springer Berlin Heidelberg.

[3] M. Spiliopoulou. Data Mining for the Web. Proceedings of the Symposium on Principles of Knowledge Discovery in Databases (PKDD), 1999.

[4] Tanasa, D., Trousse, B., Masseglia, F., & AxIS, P. (2004). Application des techniques de fouille de données aux logs web: Etat de l’art sur le Web Usage Mining. Mesures de l'internet, 126-143.

[5] Tanasa, D., & AxIS, A. (2002, December). Lessons from a web usage mining intersites experiment. In Proceedings of the First International Workshop on Data Cleaning and Preprocessing of the ICDM02 (pp. 99-107).

[6] R. Cooley. Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data. PhD thesis, University of Minnesota, 2000.

[7] Aye, T. T. (2011, March). Web log cleaning for mining of web usage patterns. InComputer Research and Development (ICCRD), 2011 3rd International Conference on (Vol. 2, pp. 490-494). IEEE.

[8] Pamutha, T., Chimphlee, S., Kimpan, C., & Sanguansat, P. (2012). Data Preprocessing on Web Server Log Files for Mining Users Access Patterns.International Journal of Research and Reviews in Wireless Communications (IJRRWC) Vol, 2.

[9] Merzoug, N., & Bessa, H. Application du processus de fouille de donnees d'usage du web sur les fichiers logs du site cubba.

[10] Charrad, M. (2005). Techniques d'extraction de connaissances appliquees aux donnees du Web. Transformation, 56, 5-2.

[11] Tanasa, D., & Trousse, B. (2003). Le prétraitement des fichiers logs web dans le “Web Usage Mining” multi-sites. Journées Francophones de la Toile (JFT’2003), 113-122.

[12] Langhnoja, S., Barot, M., & Mehta, D. (2012). Pre-Processing: Procedure on Web Log File for Web Usage Mining. International Journal for Emerging Technology and advanced enfineering, 2(12).

[13] Tanasa, D., Trousse, B., Masseglia, F., & AxIS, P. (2004). Application des techniques de fouille de données aux logs web: Etat de l’art sur le Web Usage Mining. Mesures de l'internet, 126-143.

[14] Charrad, M., Ahmed, M. B., & Lechevallier, Y. (2005). Extraction des connaissances à partir des fichiers logs. Atelier fouille du Web EGC2006, 768.

[15] Sharma, A. (2008). Web Usage Mining: Data Preprocessing, Pattern Discovery and Pattern Analysis on the RIT Web Data (Doctoral dissertation, PhD thesis, Rochester Institute of Technology).

[16] Khalil Gdoura, Web Usage Mining-Détermination des facteurs de succès d’un site web par un modèle de régression logistique, Ecole Supérieure de la Statistique et de l’Analyse de l’Information, 2008 / 2009.

[17] https://developer.mozilla.org/fr/docs/Gecko_user_agent_string_reference

[18] Catledge, L. D., & Pitkow, J. E. (1995). Characterizing browsing strategies in the World-Wide Web Computer Networks and ISDN systems, 27(6), 1065-1073.