Web Page Analysis Based on HTML DOM and Its Usage for Forum Statistics, Alerts and Geo Targeted Data Retrieval

ROBERT GYŐRÖDI, CORNELIA GYŐRÖDI, GEORGE PECHERLE, GEORGE MIHAI CORNEA
Department of Computer Science
Faculty of Electrical Engineering and Information Technology, University of Oradea
Str. Universitatii 1, 410087, Oradea
ROMANIA
rgyorodi@uoradea.ro, cgyorodi@uoradea.ro, gpecherle@uoradea.ro, generalmip@yahoo.com

Abstract: - Message boards are part of the Internet known as the 'Invisible Web' and pose many problems to traditional search engine spiders. The dynamic content is usually very deep and difficult to search. In addition, many of these sites change their locations, servers, or URLs almost daily creating problems with the indexing process. However, during the growth of the World Wide Web and with the help of search engines, they represent an important source of information to solve different problems. Another interesting feature of this type of web pages is that a big community has been developed, expressing different opinions and discussing various topics. Using special retrieval and indexing algorithms, mostly based on the HTML DOM tree, we have developed an algorithm to obtain detailed and accurate trend statistics that can be used for different marketing solutions and analysis tools. Combined with the services provided by traffic ranking sites like Alexa.com, we can also provide geo targeting functionality to deliver even more accurate results to the end user, such as what percentage of the users who are visiting a certain forum is coming from a certain country.

Key-Words: - data analysis, data models, HTML DOM, information extraction, text recognition, geo targeting

1 Introduction

During the last decade, most websites have been providing the information generated from their structured data in an underlying database through certain predefined templates or layouts. The message boards are also part of these categories.

Following the great number of message boards available on the Internet, these semi-structured web sources contain rich and unlimited valuable data for a variety of purposes. Extracting this data and then rebuilding them into a structured database represents a challenge to perform automatic data mining from web sources.

Several approaches have been reported in the literature for the purpose of building and maintaining mining scripts for semi-structured web sources. Some of them can be classified as the so-called wrappers [1]. The wrapper technique allows automatic data extraction through a predefined wrapper created for each target data source. The wrappers then accept a query against the data source and return a set of structured results to the calling application. This method is easy to implement but is hard to maintain and extend.

On the other hand, there are several automatic methods without requiring an initial manual learning process. For example, some methods are based on the automatic generation of a template from the first multiple pages, before extracting the rest of the data based on the template [2], [3], [4]. Another method to generate a template automatically is based on finding certain repetitive patterns contained in a single page based on the structure of the DOM three. The method is extensively described in [5].

Taking into account the demand for such services, we have developed an algorithm that can accurately analyze web forums and message boards and provide accurate statistics. This algorithm has been developed using special retrieval and indexing methods that will be described later in this paper.

Even if the existing methods described above can be successfully applied in our system to detect different repetitive structures like forums, threads and posts, it alone can't gather all the information required for the further processing of the message board and for an effective update process.

Until we further develop the fully automatic template generation system, we are using an alternative method that combines a reverse learning process with a system to test the templates we have already used in our system. Because of the special needs of our crawling process, we will always look for certain types of information that have a certain link between them.

Besides offering accurate trend statistics, the unique way in which our spider works allows us to give almost real time alerts when someone posts a message which contains a certain phrase in it. This way, we can offer services to our users, regarding a wide range of topics discussed in the forums: buying and selling of different products, buying and selling of services, offer help on different topics, etc.

Alexa.com [9] is a web information service that offers free traffic metrics, search analytics and demographics data. Coupled with the services offered by Alexa.com, we can also offer geo-targeting features for the statistics and the alert system that can greatly improve the quality of the services. This way, we will only provide graph results for a required region. Being able to analyze, for example, the impact of a TV advertising campaign launched only in a certain region for testing purposes, the company doing this analysis can combine the internal intelligence (consisting in sales trends, for example) with the external intelligence provided by our system.

Without the use of an advanced system to index and process the message boards, all the above mentioned features would not be possible. The old methods of indexing would require huge levels of bandwidth and computation power to be able to provide daily alerts of new messages that are updated in the forums. However, using our implementation, the bandwidth usage required by the indexing of a new message will be of an average of 70% the page that contains the message. This rate is even better for frequently updated forums in which the number of new messages between two consecutive scans is higher. The worst case appears when we perform a search for a single message contained in a topic located within the oldest topics. However, the majority of the new messages will be located within the first topics, as they are the most active ones.

The introduction of our service will also bring great benefits to the forum owners, who will get more visitors and more potential members to their communities through the alert system.

Our statistics system will be a useful tool for all the marketing companies offering different marketing solutions to their clients, by providing external intelligence. It will also be useful for managers, political analysts, etc.

2 Our Implementation of the Reverse Process to Determine the Templates

The reverse process to determine the templates [1] has to take place if all the previously determined templates failed to return an acceptable dataset, after they have been applied over the desired page. In this case, human intervention is required to generate the template. We will illustrate the process only for message boards homepages.

This process is divided in two steps:

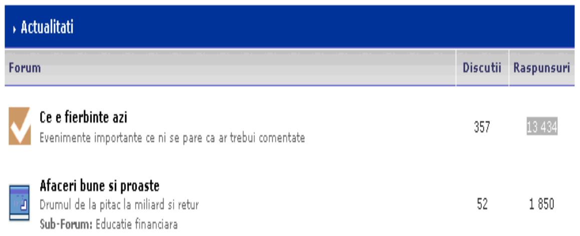
1. After it is determined that human intervention is required, a separate service will be used, that will provide support for region-of-interest (RoI) input. At the beginning of this step (because of the constant updates that are made in a forum's homepage, such as the last user who posted, number of messages, etc.), we have to save the HTML source of the page so that the reverse processing application can work on the same dataset as the human did.

2. When the database is populated with the targeted regions-of-interest, the reverse process can start to analyze the available data. It will search for all the nodes that contain the regions-of-interest provided by the human at step one above [1]. If more than one node is found to contain a targeted region-of-interest, the process will try to group them into the smallest subgraph containing all of the determined paths, taken once. This case is valid, for example, when the system has to process a page containing posts that contain other quoted posts.

For the first step, we have developed a web application to handle the interaction with the human, because it facilitates a faster and easier to implement access to the database. It consists in three major parts: a frame to browse the targeted message board page, a textbox containing the page's HTML source in case it is needed and the RoI input boxes. Using this interface an average user can insert or update around 30 message boards per hour. This operation results in more than 200 new templates added daily. Together with the increasing number of templates stored inside the database, this also generates an exponential capacity to process new message boards using the templates already stored in the database.

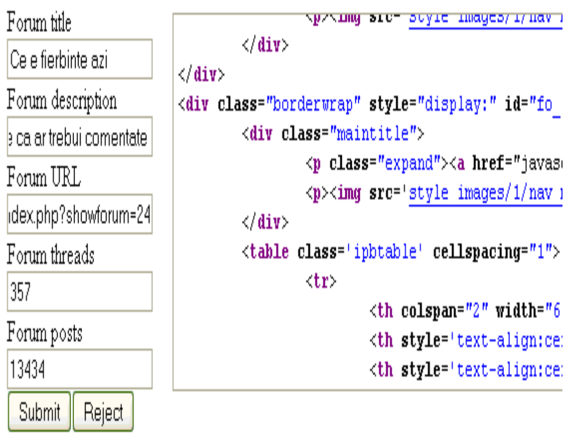
The second step is the one when the actual template consisting in the absolute un-indexed path to every RoI is determined, by searching for the node containing the targeted information.

In order to eliminate all possible errors due to repetitive information (for example, two forums with the same number of messages), all the matching nodes have to be selected. At the end of the process all these nodes will be grouped, one of each category, in the minimum tree spanning all the required vertices once [6]. All of these operations are done in the previously saved page's HTML source, to eliminate any discordances.



› Actualitati

Forum	Discutii	Raspunsuri
Ce e fierbinte azi Evenimente importante ce ni se pare ca ar trebui comentate	357	13 434
Afaceri bune si proaste Drumul de la pitac la miliard si retur **Sub-Forum:** Educatie financiara	52	1 850



Forum title

Ce e fierbinte azi

Forum description

e ca ar trebui comentate

Forum URL

idex.php?showforum=24

Forum threads

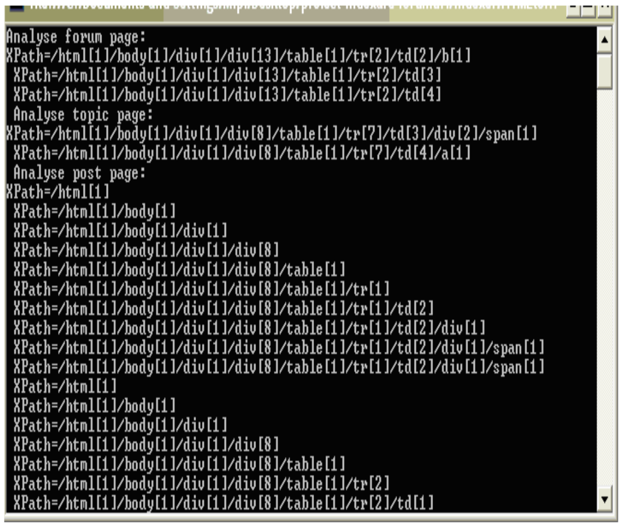
357

Forum posts

13434

```
                    <p><img src='style images/1/nav i
          </div>
</div>
<div class="borderwrap" style="display:" id="fo_
          <div class="maintitle">
                    <p class="expand"><a href="javas
                    <p><img src='style images/1/nav i
          </div>
          <table class='ipbtable' cellspacing="1">
                    <tr>
                              <th colspan="2" width="6
                              <th style='text-align:ce
                              <th style='text-align:ce
```

Fig. 1. Interface of our web application to provide RoI for forums



```
Analyse forum page:
XPath=/html[1]/body[1]/div[1]/div[13]/table[1]/tr[2]/td[2]/b[1]
 XPath=/html[1]/body[1]/div[1]/div[13]/table[1]/tr[2]/td[3]
 XPath=/html[1]/body[1]/div[1]/div[13]/table[1]/tr[2]/td[4]
 Analyse topic page:
XPath=/html[1]/body[1]/div[1]/div[8]/table[1]/tr[7]/td[3]/div[2]/span[1]
 XPath=/html[1]/body[1]/div[1]/div[8]/table[1]/tr[7]/td[4]/a[1]
 Analyse post page:
XPath=/html[1]
 XPath=/html[1]/body[1]
 XPath=/html[1]/body[1]/div[1]
 XPath=/html[1]/body[1]/div[1]/div[8]
 XPath=/html[1]/body[1]/div[1]/div[8]/table[1]
 XPath=/html[1]/body[1]/div[1]/div[8]/table[1]/tr[1]
 XPath=/html[1]/body[1]/div[1]/div[8]/table[1]/tr[1]/td[2]
 XPath=/html[1]/body[1]/div[1]/div[8]/table[1]/tr[1]/td[2]/div[1]
 XPath=/html[1]/body[1]/div[1]/div[8]/table[1]/tr[1]/td[2]/div[1]/span[1]
 XPath=/html[1]/body[1]/div[1]/div[8]/table[1]/tr[1]/td[2]/div[1]/span[1]
 XPath=/html[1]
 XPath=/html[1]/body[1]
 XPath=/html[1]/body[1]/div[1]
 XPath=/html[1]/body[1]/div[1]/div[8]
 XPath=/html[1]/body[1]/div[1]/div[8]/table[1]
 XPath=/html[1]/body[1]/div[1]/div[8]/table[1]/tr[2]
 XPath=/html[1]/body[1]/div[1]/div[8]/table[1]/tr[2]/td[1]
```

Fig. 2. Example of resulting paths from the DOM tree, after applying our algorithm

After the paths for a certain forum have been determined, the template path is considered to be the deepest sub-tree containing one of the predetermined RoIs, followed upwards until we reach a common root for all the selected RoI [14].

During this process, the following errors could appear resulting in a request to re-enter another set of RoI:

1. The most common one: a certain RoI was not found – it is usually caused by human mistake or by charset problems.

2. After the template was generated and a run test has been done, the template returns only one row – this is more a kind of a warning, because the template can be good but the page may consist of only one forum, thread or post.

3 Our Implementation of the Process to Determine the Match of a Template with a New Forum

This is the first step taken every time a new forum has been added in the indexing list. This was not implemented only for this reason, because it is also used to determine if there are changes in the structure of a forum that already has a template. With some new methods added to the first class (that only determines if a template returns a plausible result set), it is also possible to determine if a template in the database can be used over the new forum. A clustering based on the partial roots gives us an algorithm with O(log(n)), where n is the number of templates stored in the database.

The result check is done based on a few simple presumptions (this example is for forums):

1. The messages and the threads in a forum will always be numbers containing spaces, dots, or commas among numeric characters.

2. The number of messages will always be higher or equal to the number of threads.

3. The average length of the forum title will always be smaller than the average length of the description.

4 Our Indexing and Alert System

As it was told before, the main feature on which we are focusing is to provide a cost effective and fast way to crawl the deep invisible Internet. Other methods that have been implemented so far use an algorithm to determine the rate at which a page is updated and approximate the next date at which the new scan should occur.

This method doesn't provide either a cost effective way to crawl the Internet or an accurate one. The first inconvenience comes when a new page is crawled for the first time. In order to calibrate and obtain the most convenient update time, the page will be crawled several times more fervently. After the statistics algorithm has enough data to return a smaller error, it can be subject to an error increase if the way in which the page changes is unpredictable. Another disadvantage is that, this kind of algorithm will never be able to provide real time access to page changes without heavily overloading both the crawled server and the crawling server, in order to minimize the time between two scans [15].

The way in which our system works allows us to determine if there was an update on a certain forum. Tracking down its structure, we can determine exactly what page was updated and take the required actions.

The typical formulas to calculate the costs are as follows:

1. Forum update check:

Cost/day [KB/day] = mainPage [KB] * 6 * 24

mainPage = the size of the main (home) page in KB

This is the cost representing the amount of information exchange between the crawling and the crawled server to determine if a new message has been added on a certain forum. The update is done once, every 10 minutes. This way, the size of the main page is multiplied by 6 (in one hour, there are 6 groups of 10 minutes) and then by 24 (the number of hours in one day).

2. Topic update check:

Cost/topic [KB/topic] = mainPage [KB] + topicOffset [number] * forumSize [KB]
mainPage = the size of the main (home) page in KB
topicOffset = the nearest integer greater than or equal to (the number of topics, between the first topic and the topic that was updated / number of topics per page)
forumSize = the size of the forum in KB

This is the cost representing the amount of information exchange between the crawling and the crawled server to determine which topic has been updated. Since we already have the information contained in mainPage, we don't have to download it again, in case of a periodic update.
As it can be seen from the cost formula, the worst case occurs when we have an update in the last few pages of the forum (topicOffset takes the maximum value). However, this case does not occur too often, because the last topics are the older ones, abandoned or closed.

3. Message update check

Cost/message [KB/message] = mainPage [KB] + topicOffset [number] * forumSize [KB] + k [number] * topicSize [KB]
mainPage = the size of the main (home) page in KB
topicOffset = the nearest integer greater than or equal to (the number of topics, between the first topic and the topic that was updated / number of topics per page)
forumSize = the size of the forum in KB
k = the nearest integer greater than or equal to (msgNb / msgPerPage)
msgNb = the number of messages that have been updated
msgPerPage = the number of messages in a page
topicSize = the size of the topic, in KB

This is the cost representing the amount of information exchange between the crawling and the crawled server to determine and to process the new messages. As it can be seen from the formula, because the new messages are well ordered there is no need to check all the messages, only the last msgNb of them determined at the previous stage. The first two arguments are already obtained from the previous step so they will be shared every time there is more than one message to update.

Considering the fact that at every new update that we perform, we have more than one topic to process and more than one message to add to the database and to process, we won't have the entire cost calculated using the cost/message formula. Instead, parts of that cost will be shared by multiple updated messages, resulting in an even lower total cost per number of messages indexed. As an example from a medium Romanian forum, we have the following average values during 24 hours of testing:

1. 25 of 144 main page checks returned a positive message to follow through. This represents a percentage of 17.36% .
2. 3 follows were performed through sub-forums adding an extra cost required to further follow the forum tree downwards.
3. All the messages were in the first page of the topics, resulting in an offset cost of 1 (the best case because all the messages were in new threads).
4. The messages were clustered through the positive follow through with an average of 2.32 further decreasing the average cost per message.

At the end of the test period we had:

- 8485.7 KB downloaded for the forum homepage (consisting in the root of the forum). From this bandwidth usage, 17.36% was used to further detect threads.
- 100% of the threads were found in the first page of the forum. With an offset cost of 1, the total bandwidth used here was 1726.6 KB.
- there were 92 messages to process. The total bandwidth used to process those messages was 10952 KB. 74 page loads were required for this task.

The total bandwidth used for the scanning of the forum through the test period of 24 hours was of 21671.8 KB (21.2 MB). The extra bandwidth of 435.1 KB comes from the messages to which the tree has been followed through sub-forums.

The share rate resulting in decreased bandwidth usage and greater performance was as follows:

- The main page was loaded 144 times during the test period of 24 hours. Only the 25 follow through are taken into account further.
- 31 threads were marked as updated using the 25 scans of the main page. This results in a share rate of 80.64%.
- furthermore, 74 message pages were determined by analyzing the thread information, resulting in a share rate of 41.89%. From those message pages, we were able to extract 92 messages.

The approximate bandwidth that could be used to periodically update this site in the worst possible case is 35MB, calculated using only the positive presumption that the forum offset is still equal to 1 because we have updates in the most recent threads.

The increased size appears because of zero clustering. So for every new message the tree has to be followed downwards without sharing nodes.

5 Process Diagram for Our System

In the next diagram we will try to illustrate the entire process that takes place every time a new forum is added to the database to be scanned and every time a forum is updated to scan for new messages.

The diagram has two main parts:

- the reverse process to determine the templates and the first indexing process
- the periodical update indexing, when data can be either added to the statistics database or to the alert system database

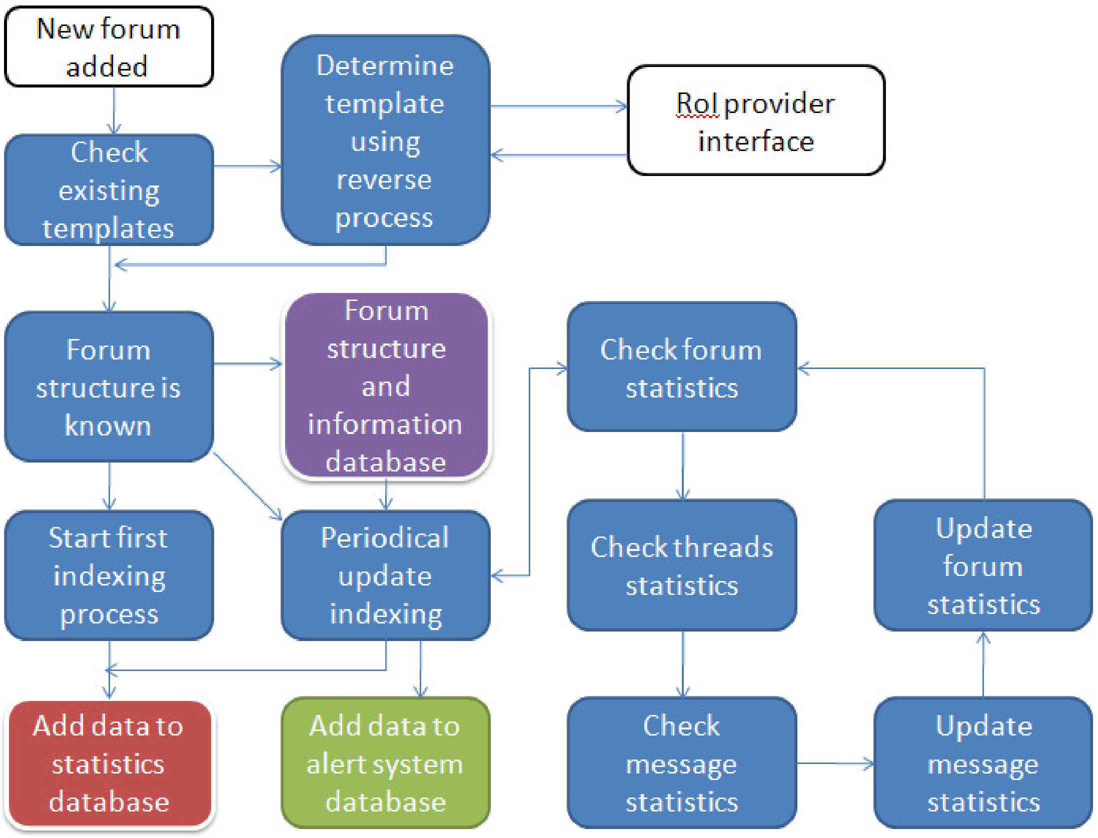

New forum added
Determine template using reverse process
RoI provider interface
Check existing templates
Forum structure is known
Forum structure and information database
Check forum statistics
Start first indexing process
Periodical update indexing
Check threads statistics
Update forum statistics
Add data to statistics database
Add data to alert system database
Check message statistics
Update message statistics

Fig. 3. Complete process diagram for our system

6 Large Database Support

Over time, the database where we store all the information may become very large, due to the amount of data gathered from multiple forums. Also, our system should support expanding to an unlimited number of forums to be analyzed, therefore large database support is needed.

For practical reasons, we use MySQL, however any other database management systems that support large databases can be used. In MySQL, there is a feature called partitioning. This feature allows you to split parts of individual tables across a file system. This operation is done according to specific rules. In effect, different portions of a table are stored as separate tables in different locations [7].

Here are some of the advantages of using partitioning:

- Save more data in a single table, than can be held on a single disk.
- When some of the data loses its usefulness, it can be easily removed from the table, by eliminating the partition containing only that data. On the other hand, when new data has to be added, this process can be facilitated by adding a new partition specifically for that data.
- Some queries can be optimized taking into account the fact that data satisfying a given condition (specified in the WHERE clause) can be stored only on one or more partitions. This excludes any remaining partitions from the search, thus optimizing the query. Because partitions can be altered after a partitioned table has been created, you can reorganize your data to enhance frequent queries that may not have been so when the partitioning scheme was first set up. This capability is referred to as partition pruning [7].

Because data collections can sometimes be beyond the amount of RAM that can be installed, we need to use database partitioning. In most DMBS, the indexes are cached in RAM; this allows fast retrieval of records. This is the reason we need to use a different approach (data can reach sizes that can't be saved in RAM) [8].

One useful way to apply partitioning in our system is to make partitions based on the year, because each message in a forum has a datestamp attached. This way, we will have all our forum messages organized by year, allowing an easier management and faster data retrieval.

Because the native date type is not supported, we must convert the date into a number. We must take into account the fact that only two date functions can trigger the partition pruning. Therefore, if we have to deal with a column of DATE type, we need to use one of these functions: YEAR or TO_DAYS.

When using the YEAR() function, partitioning is easy [8].

```
CREATE TABLE by_year (d DATE)
PARTITION BY RANGE (YEAR(d))
(PARTITION P1 VALUES LESS THAN (2007),
PARTITION P2 VALUES LESS THAN (2008),
PARTITION P3 VALUES LESS THAN (2009),
PARTITION P4 VALUES LESS THAN
(MAXVALUE))
```

The basic concept behind partition pruning is that of not scanning partitions where there can be no matching values. For example, when we need to retrieve messages older than 2007, we will only scan partition p1 from the example above and do not scan partitions p2, p3 or p4. This greatly improves performance and the time needed to retrieve data from our tables.

Another way of partitioning is at the forum level, not at the message level. This way, we identify the following possible partitioning types:

1. Partitioning by forum type: when a new forum is added to the database, the forum type has to be specified (for example IT, Music, Art, etc.). We can group the messages inside these forums by the main forum type. When someone searches just for messages in a certain category (for example "Music"), just the partition corresponding to that category is queried (in this case "Music").

2. Partitioning by forum country: this can be either added manually or determined from the domain name (.ro for Romanian, etc). Possible partitions are one for each country. If the country is not known, these forums can be grouped in a separate partition (a miscellaneous partition). Also, it would be a very good idea to correlate this with the Alexa Geo Targeting services.

3. Partitioning by forum popularity. We can use the Alexa Traffic Rank to determine the popularity of a forum. The way we build the partitions by popularity is the following: for the most popular forums (small rank), we have a small number of forums in one partition; for the less popular forums (high rank), we have a larger number of forums in one partition. This is because the most popular forums have more frequent updates and we have to make sure we don't insert too many popular forums in one partition, to avoid server overloading. For this reason, we are using a logarithmic function such as: $\mathbf{y = \ln(k*x) + 1}$, where

y = the number of forums in one partition

k = a constant value that is chosen based on server overloading (if the server can accept more forums in one partition, k may be higher)

x = Alexa Traffic Rank (1 is the most popular)

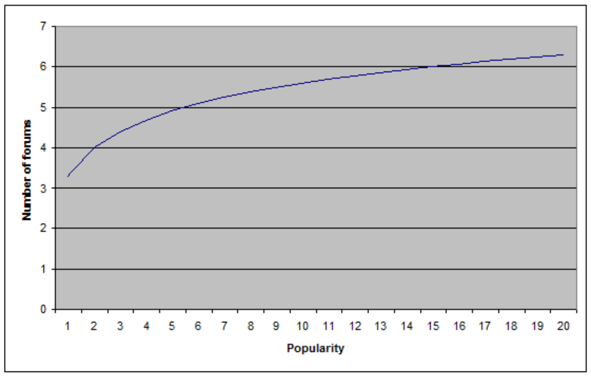


Fig. 4. Number of forums in a partition based on Alexa Traffic Rank (popularity)

7 Alexa and geo targeting

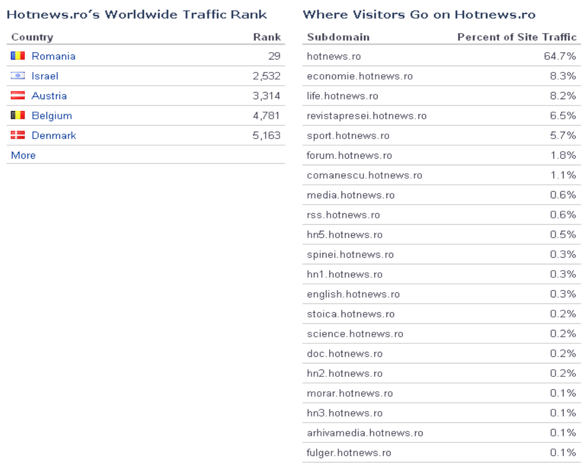
Geo targeting is used to determine the physical location of a website visitor in order to provide different content based on his/her region, local preferences, IP address, payment methods or other criteria. [10]

Because we don't have access to the end users (in order to get more information about them), we have to build our system taking into account a statistical background to provide users with complete information for their marketing campaigns.

Alexa Ranking Services provide information about Internet sites. Some of this information includes Top Sites, Internet Traffic Stats and Metrics, Related Links, Online Reviews, Contact Information and Search Analytics for SEM and SEO optimization. Also, Alexa provides its own toolbar (named Alexa Toolbar), that is a browser add-on that shows data about websites as they are visited by Internet users (the Alexa Traffic Rank is one of the most useful data shown in this toolbar).

The purpose of Alexa is to collect and keep information about the web pages viewed by Internet users, the data entered in online forms and also in search input fields (this is only possible while using the Alexa browser companion software). With versions 5.0 and higher of the browser companion software, Alexa also keeps track of the products you purchase online. Although Alexa does not attempt to analyze web usage data to determine the identity of any user, some information collected by the software is personally identifiable. Also, Alexa analyzes the information it collects for the purpose of improving its service and to deliver reports about aggregate web usage and purchasing habits. [9]

Figures 5 and 6 illustrate some of the statistics provided by Alexa for the Romanian News Website, www.hotnews.ro:



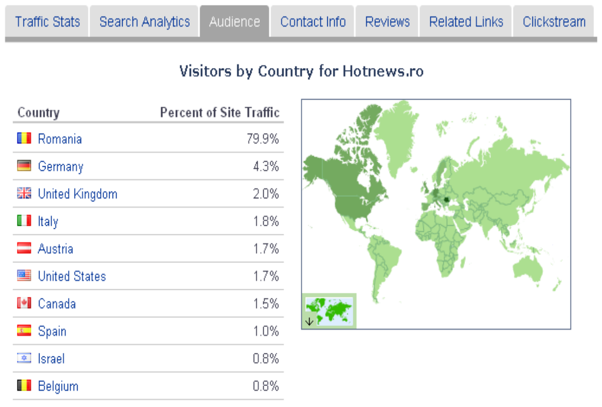
Hotnews.ro's Worldwide Traffic Rank

Country	Rank
Romania	29
Israel	2,532
Austria	3,314
Belgium	4,781
Denmark	5,163
More	

Where Visitors Go on Hotnews.ro

Subdomain	Percent of Site Traffic
hotnews.ro	64.7%
economie.hotnews.ro	8.3%
life.hotnews.ro	8.2%
revistapresei.hotnews.ro	6.5%
sport.hotnews.ro	5.7%
forum.hotnews.ro	1.8%
comanescu.hotnews.ro	1.1%
media.hotnews.ro	0.6%
rss.hotnews.ro	0.6%
hn5.hotnews.ro	0.5%
spinei.hotnews.ro	0.3%
hn1.hotnews.ro	0.3%
english.hotnews.ro	0.3%
stoica.hotnews.ro	0.2%
science.hotnews.ro	0.2%
doc.hotnews.ro	0.2%
hn2.hotnews.ro	0.2%
morar.hotnews.ro	0.1%
hn3.hotnews.ro	0.1%
arhivamedia.hotnews.ro	0.1%
fulger.hotnews.ro	0.1%

Fig. 5. Hotnews.ro traffic [11]



Visitors by Country for Hotnews.ro

Country	Percent of Site Traffic
Romania	79.9%
Germany	4.3%
United Kingdom	2.0%
Italy	1.8%
Austria	1.7%
United States	1.7%
Canada	1.5%
Spain	1.0%
Israel	0.8%
Belgium	0.8%

Fig. 6. Hotnews.ro geo targeting [11]

Together with this information, Alexa also stores geographical information about the people who are using the Alexa Toolbar and generates a statistical report with the percentage of visits coming from different countries to a single website.

This way we can know what percentage of the users who are visiting a certain forum is coming from a certain country [13]. Assuming that the number of visits from a certain country is proportional with the number of posts made by users from that country, we can calculate a statistical value for how many keywords have been posted by users from a certain country using the following proposed formula:

geotargetedKeywords = totalKeywords * geotargetedPercent / 100

where

- geotargetedKeywords represents the number of keywords that are assumed to come from a certain country
- totalKeywords represents the total keyword appearances in the forum
- geotargetedPercent represents a parameter taken from Alexa that shows the percentage of visitors from certain countries. In this case, we have to choose the one that represents the visits from the country selected by the user.

Many websites have their forums listed as subdomains. Because we are only using the statistical percentages from Alexa, this is not affecting our system at all; we can still assume that the above equation is statistically correct [16].

If we take a look at the other Alexa statistics that we can obtain for a certain site, we can see that the system has huge potential for further development to include even more specific statistics about the age, gender, education, etc.

Also, we can provide statistics on the popularity of the web forums, based on an indicator from Alexa, called Alexa Traffic Rank. The lower this number, the higher the traffic of that website. For example, Hotnews.ro had an Alexa Traffic Rank of 6,494 at the date of writing this paper. This means that it has a very high traffic and a large amount of visitors (actually sites below 100,000 in Alexa Traffic Rank are considered to be highly visited websites).

In addition to the general traffic rank (not geographically dependent), Alexa also provides Traffic Rank for specific geographical locations. For example, Hotnews.ro had a traffic rank of 22 in Romania (see Figure 7 below).

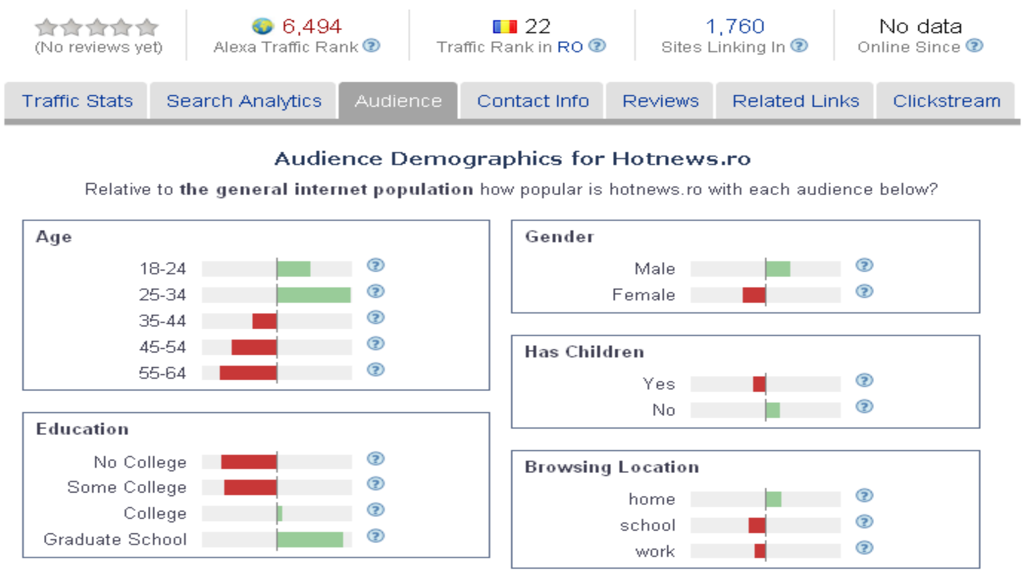


Fig. 7. Other statistics provided by Alexa [11]

Other advanced demographic information we can take from Alexa includes: audience breakdown by income, ethnicity, age, education, gender, children and browsing location.

8 The Usage of HtmlAgilityPack to Build the DOM Tree of Web Forums

HtmlAgilityPack is a HTML parser that builds a read/write DOM (Document Object Model). It supports plain XPATH or XSLT. Internally, HtmlAgilityPack is a .NET library that allows you to parse out HTML files. An advantage of this parser is that it is very tolerant with malformed HTML, which is common in many websites. The object model is very similar to what proposes System.Xml, but for HTML documents (or streams). [12]

We have used this parser in our project to convert the stream that we obtained from the server into a more computer readable structure that we are able to process and extract useful information. After the stream is processed by this library, we only have to apply XPATH on it to get the desired information.

In the next paragraphs, we will present and explain some of the code used in the early testing stage:

```
ArrayList forums = new
ArrayList();

ArrayList topics = new
ArrayList();

ArrayList posts = new
ArrayList();

Boolean stopIndexation =
true;

Boolean analyse = true;
Boolean DOMPath = false;
HtmlPage page = new
HtmlPage();

if (DOMPath == true)
    {
      page.url =
"http://forum.hotnews.ro/";
```

```
    page.poweredBy =
    "invisionboard";
    page.forumId = 1;
    page.subforumId = 0;
    page.UrlOpen(page.url);

    HtmlDocument html = new
    HtmlDocument();

    html.LoadHtml(page.htmlText
    );
    HtmlNodeCollection nod =
    html.DocumentNode.SelectNod
    es("//*");

    for (int i = 0; i <
    nod.Count; i++)
    {
    Console.WriteLine(nod[i].XP
    ath);
    Console.ReadKey();
    }
   }
```

Fig. 8. Code that lists the structure of the webpage

The above piece of code does nothing else than list the structure of the webpage we have just opened. Most of the classes and methods used here are writen by us, but their names are intuitive enough.

Some of the functions from the HtmlAgilityPack are used at the end, in order to generate the DOM tree and to output its structure.

```
public void
analysePost(invisionboardPost post)
{
HtmlDocument html = new
HtmlDocument();
html.LoadHtml(this.htmlText);
            String path = "";
            HtmlNode nod;
do
{
nod =
html.DocumentNode.SelectSingleNode(pat
h + "//*[contains(.,'" + post.date +
"')]");
if(nod==null)
break;
path = nod.XPath;

Console.WriteLine("XPath=" +
nod.XPath);
  Console.ReadKey();
}while (nod != null);

Console.WriteLine("XPath=" + path);
Console.ReadKey();
path = "";

do
{
nod =
html.DocumentNode.SelectSingleNode(pat
h + "//*[contains(.,'" +
post.description + "')]");
if (nod == null)
break;
path = nod.XPath;

Console.WriteLine("XPath=" +
nod.XPath);
Console.ReadKey();
} while (nod != null);

Console.WriteLine("XPath=" + path);
Console.ReadKey();
path = "";

do
{
nod =
html.DocumentNode.SelectSingleNode(pat
h + "//*[contains(.,'" + post.message
+ "')]");
if (nod == null)
break;
path = nod.XPath;

Console.WriteLine("XPath=" +
nod.XPath);
Console.ReadKey();
} while (nod != null);

Console.WriteLine("XPath=" + path);
Console.ReadKey();
}
```

Fig. 9. Determine the path to a previously determined RoI taken from the database

The above example is also taken from the test application, written before the start of the main project. However, without some blocking call functions and database and serialization support, it is about the same as the one actually written for the main project.

This method is used to determine the path to a previously determined RoI taken from the database.

9 Conclusions and Future Work

The algorithm we propose is a new concept in the process of indexing structured web pages with a lower cost (lower bandwidth and increased performance). The method we described in this article can be applied to other types of structured web pages, not only for web forums.

Also, it can be a useful tool for gathering marketing statistics, based on the discussions on a certain product or service, identified by one or more keywords.

There are several issues that require further modifications and implementation in the algorithm. The first one represents the poor reachability of a message that is placed among the last threads. In the case in which we have an extremely active forum with large numbers of updates, the time between a scan of the main page and until all the required messages were found is big enough to manage new messages. The scan for new messages will be stopped as soon as we reach the pre-determined number of messages, so that it is possible to miss the last messages. Another issue appears when there are deleted messages or threads, or moved threads. This case is completely unhandled, human intervention being required to correct the data and continue the scan.

In the near future, we want to implement handling routines for these special cases. Another future implementation we want to do is to switch from the test server to a live one and optimize it so that it can work on multiple machines simultaneously. We intend to offer this solution to webmasters or marketing companies who want to offer quality information to their customers, about the popularity of certain topics within web forums.

References:

[1] Z. Akbar and L.T. Handoko, “Reverse method for labeling the information from semi-structured web pages”, proceeding of the 2009 International Conference on Signal Processing Systems pp. 551-555

[2] L. Arlota, V. Crescenzi, G. Mecca, P. Merialdo, Automatic annotation of data extracted from large websites, in: Proceedings of the WebDB Workshop, 2003, pp. 7–12.

[3] N. Kushmerick, Regression testing for wrapper maintenance, Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence, July 18-22, 1999, pp. 74– 79.

[4] J. Wang , F.H. Lochovsky, Data extraction and label assignment for web databases, Proceedings of the 12th international conference on World Wide Web, May 20–24, 2003, pp. 187–196.

[5] Li WeiDong, Dong Yibing, Wang RuiJiang, Tian HongXia, “Information Extraction from Semi-Structured WEB Page Based on DOM Tree and Its Application in Scientific Literature Statistical Analysis System”, 2009 IITA International Conference on Services Science, Management and Engineering

[6] Jie Zou Le, D. Thoma, G.R., "Combining DOM tree and geometric layout analysis for online medical journal article segmentation", Digital Libraries, 2006. JCDL '06. Proceedings of the 6th ACM/IEEE-CS Joint Conference, pp. 119 - 128

[7] MySQL Developer Zone (http://dev.mysql.com) – “Chapter 18. Partitioning”

[8] MySQL Developer Zone (http://dev.mysql.com) – “MySQL partitions in practice – Giuseppe Maxia”

[9] Alexa.com – The Web Information Company - Free traffic metrics, search analytics, demographics

[10] http://en.wikipedia.org/wiki/Geo_targeting

[11] Alexa.com – traffic information for Hotnews.ro http://www.alexa.com/siteinfo/hotnews.ro

[12] HTML Agility Pack DOM parser - http://htmlagilitypack.codeplex.com/

[13] "Text Analytics as a Form of Knowledge Mining" - ZELJKO PANIAN, The Faculty of Economics and Business University of Zagreb, WSEAS - Proceedings of the 4th EUROPEAN COMPUTING CONFERENCE, pp. 112-118, ISSN: 1790-5117, ISBN: 978-960-474-178-6

[14] “Artifacts Extraction Technique” - Nadim Asif, Dept. of Computer Science, Islamia University of Bahawalpur, Pakistan, WSEAS - PROCEEDINGS OF THE 3RD INTERNATIONAL CONFERENCE ON COMMUNICATIONS AND INFORMATION TECHNOLOGY (CIT'09), pp. 129-139, ISSN: 1790-5109, ISBN: 978-960-474-146-5

[15] “A new approach for better document retrieval and classification performance using supervised WSD and Concept Graph” - Reza Soltanpoor, Mehran Mohsenzadeh, Morteza Mohaqeqi, WSEAS - PROCEEDINGS OF THE 3RD INTERNATIONAL CONFERENCE ON COMMUNICATIONS AND INFORMATION TECHNOLOGY (CIT'09), pp. 176-183, ISSN: 1790-5109, ISBN: 978-960-474-146-5

[16] "Domain specific information retrieval system" - SANDI POHOREC, MATEJA VERLIČ, MILAN ZORMAN, Laboratory for system design, University of Maribor, Faculty of Electrical Engineering and Computer Science, SLOVENIA - Proceedings of the 13th WSEAS International Conference on COMPUTERS (2009), pp. 465-470, ISSN: 1790-5109, ISBN: 978-960-474-099-4