

AD-EXTRACTOR TOOL



Developer: Lalit Agarwal

About Ad-Extractor

- A tool to extract and identify advertisements from a given list of webpages.
- Extracts contents of both image and textual ads.
- Outputs an excel file containing information of the ads present on the given webpages
- The tool was developed as a part of a research study.

Motivation

- Understand the nature of advertisements being shown to the users.
- On the basis of data collected, identify any common features to identify and block advertisements on the basis of categories.
- Identify any ads which can be considered inappropriate or embarrassing by users.

Methodology

- Collected history from users who were part of the user study after their consent.
- We ran this tool on three different set of webpages containing 500, 2500 and 5000 URLs respectively collected from user's browsing history during the user study.

Tools Used



EASY LIST

Tools Used

- In order to perform extraction of ads, **Selenium's Firefox web driver** was used which is a famous browser automation tool.
- It automates web application for testing purposes and allows running automated scripts to perform various tasks.
- To parse the HTML, Jsoup parser was used.
- Aho-Corasick string matching algorithm was used for comparing strings.
- We used the Easylist's list of filters which is also used by many adblockplus users.

Implementation

□ Ads inside an anchor tag

- These ads may be images or textual ads and are located inside anchor tags
- Usually these are non-behavioral ads ie. every user gets to see the same ads when they visit the same website.
- They are loaded without any use of JavaScript.
- They are of the form:

```
<a href=http://www.makemytrip.com/flights>  
<img src=http://makemytrip/flights/offers.jpg>  
</a>
```

Ads inside anchor tags



```
<a href="http://www.nytimes.com/adx/bin/adx_click.html?type=goto&opzn&page=hom..e%3Dhomepage.nytimes.com/index.html%26pos%3DHPMiddle%26campaignId%3D39XJX" target="_blank">  
    
</a>
```


Implementation

□ Ads inside an iframe-

- These ads are usually tailored based on the user's demographics, browsing pattern etc.
- They are loaded using JavaScript.
- For such kind of ads, the browser sends HTTP get requests to the web server along with the required cookies so that they can be customized for the user. They usually take some time to load.

- They are of the form:

```
<iframe src=http://google.ad.doubleclick.com">  
<html><body>  
<a href="googleadservices.com/pagead/adf">  
<img src=http://makemytrip/img/flight.jpg>  
</a></body</html>  
</iframe>
```

Ads inside iframes

Bangalore to Delhi @ 4499

Use code VIADOM & Get upto 5000 off Limited Period Offer. Book Today

Via.com/Bangalore-Delhi-Flights

Goibibo™ Flight Discounts

Book Now & Get FLAT Rs.1000 Off. on Any International Flights.Hurry!

goibibo.com/International-Flights

Ads by Google

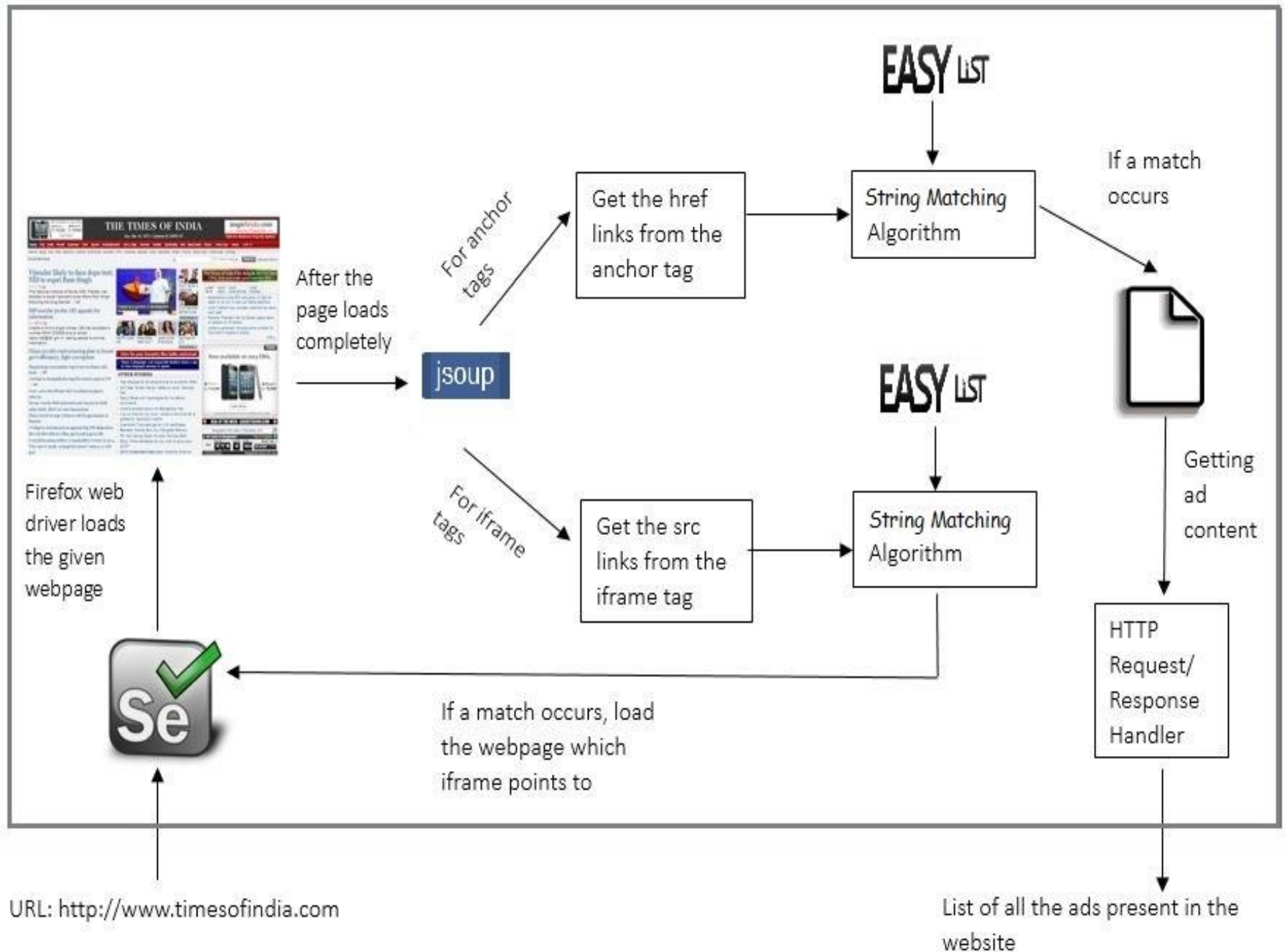
```
▼<iframe allowtransparency="true" title="Advertisement" align="center" scrolling="no" frameborder="0" vspace="0" hspace="0" marginheight="0"
marginwidth="0" height="82" width="670" src="http://timesofindia.indiatimes.com/configspace/ads/toiqoogleads.html">
  ▼#document
    ▼<html>
      ▶<head>...</head>
      ▼<body marginwidth="0" marginheight="0">
        <script language="JavaScript1.1" src="http://googleads.g.doubleclick.net/pagead/ads?client=ca-timesofindia_s...
        7&oid=3&loc=http%3A%2F%2Ftimesofindia.indiatimes.com%2F&fu=4&if=1&dtd=112"></script>
        ▼<table cellpadding="0" cellspacing="0" width="665" height="75" style="border-style:dotted;border-left-width:0px;border-right-width:0px;
border-top-width:1px;border-bottom-width:0px;border-color: #d9d8d8 " border="0" bgcolor="#ffffff">
          ▼<tbody>
            ▼<tr>
              ▼<td valign="top" align="left">
                ▼<table cellpadding="0" cellspacing="0" width="355" border="0" style="border-style:line;border-left-width:0px;border-right-width:0px;border-top-width:
0px;border-bottom-width:0px;border-color: #ebebcb">
                  ▼<tbody>
                    ▼<tr>
                      ▼<td bgcolor="#ffffff" style="padding-left:5px;padding-right:10px;padding-top:0px;padding-bottom:0px;" colspan="2" valign="top">
                        <a href="http://www.googleadservices.com/pagead/aclick?sa=L&ai=C4L_5YkI8UZydCuTDi...
                        ource%3Dgoogle%26utm_medium%3Dcpc%26utm_campaign%3Dremarketing-blr-del-txt" target="_blank" style="font-family:georgia;font-
                        size:16px;color:#024D99;font-weight:bold;" onmouseover="window.status='Via.com/Bangalore-Delhi-Flights';return true"
                        onmouseout="window.status='';return true">Bangalore to Delhi @ 4499</a>
                      </td>
                    </tr>
                  </tbody>
                </table>
              </td>
            </tr>
          </tbody>
        </table>
      </body>
    </html>
  </iframe>
```

Procedure

- The first step was to fetch the webpage from which ads are to be extracted using Firefox web driver. The web driver automates the Firefox browser i.e. it opens the Firefox browser and waits for the page to load, allowing JavaScript to execute if required.
- Once the page completely loads,
 - ▣ **For ads inside anchor tags,**
 - Tool parses the HTML content of the page using Jsoup to search for all the anchor tags.
 - For each anchor tag, it compares the href link in that anchor tag with the list of advertisement filters to see if that link is an ad or not. If it is an ad, it stores the link in a file.

Procedure

- **For ads inside iframes**, the web driver identifies all the iframes in the HTML page using Jsoup and compares the source links to a list of all third-party advertisers using the Aho-Corasick string matching algorithm.
 - If string match occurs, then another instance of web driver is used to load the iframe webpage from the source link.
 - The new webpage is parsed to look for all anchor tags and compare them with a list of ad filters. If the string match occurs again, the link is stored on a file.
- Finally to get the content of these ads, HTTP get requests is sent to the web sever on all the links stored in the file which were identified as ads and content of the ads are fetched from the HTTP response received.



Data collected

- Ad-Title
- Ad-Content
- Ad-Display URL
- Ad-Source URL
- Landing Page Title
- Landing Page URL
- Image Source
- URL of the main page
- isThirdParty?
- isIFrame?



Bangalore to Delhi @ 4499
Via.com/Bangalore-Delhi-Flights
Use code VIADOM & Get upto 5000 off Limited Period Offer. Book Today



```
<a href="http://www.googleadservices.com/pagead/ac1k?sa=L&ai=CakYznLt2UZf6B0a-i...ource%3Dgoogle%26utm_medium%3Dcpc%26utm_campaign%3Dremarketing-blr-del-txt" target="_blank" style="font-family:georgia;font-size:16px;color:#024D99;font-weight:bold;" onmouseover="window.status='Via.com/Bangalore-Delhi-Flights';return true" onmouseout="window.status='';return true">Bangalore to Delhi @ 4499</a>
```

Simulation Results

- Avg. time taken to get all the ads/webpage: 8 sec
- Avg. time taken to
 - ▣ Load a web-page: 4.5 sec
 - ▣ Fetch all anchor tags/webpage: 0.5 sec
 - ▣ Fetch all iframe tags/webpage: 4.3 sec
- Avg. no. of anchor tags in a webpage: 290
- Avg. no. of iframe tags in a webpage: 4

Results

Data Set	Text Ads	Embarrassing Text Ads	Image Ads	Embarrassing Image Ads
Set 1 (500 URLs)	192	4	156	5
Set 2 (2500 URLs)	1235	29	742	16
Set 3 (5000 URLs)	2587	40	1423	30
Total	4014	73 (2%)	2321	51 (2%)

Embarrassing Ads

Text Ads

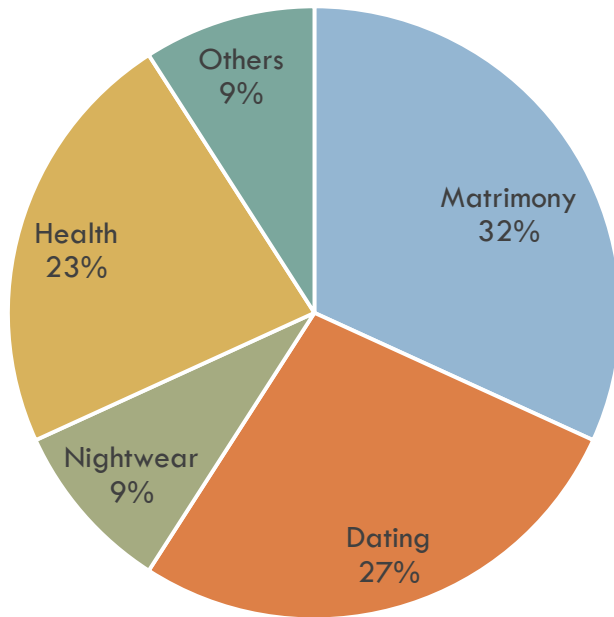
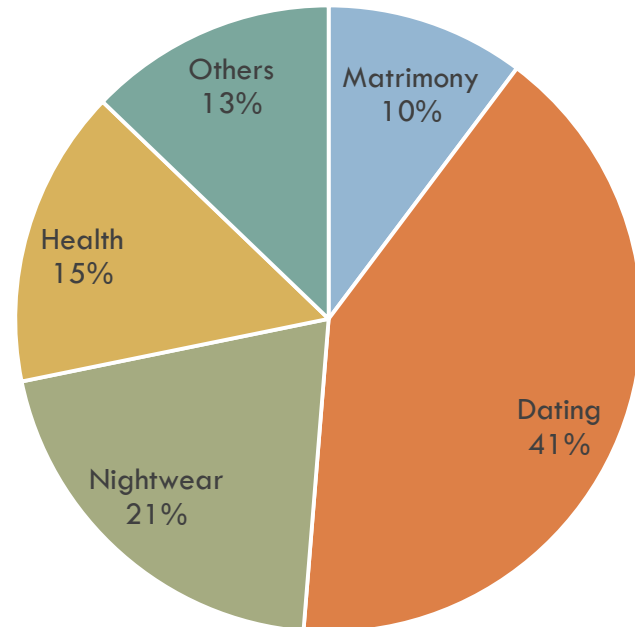


Image Ads



■ Matrimony ■ Dating ■ Nightwear ■ Health ■ Others

■ Matrimony ■ Dating ■ Nightwear ■ Health ■ Others

Limitations of the tool

- The tool identifies only textual and image ads. It does not identify flash ads.
- Since some of the ads are loaded using Javascript, the tool waits for the entire webpage to load before it can extract the ads.
- Headless browsers tool which can extract ads loaded using Javascript are currently not available.

Acknowledgement

I would like to thank Dr. Saurabh Panjwani, Dr. Sharad Jaiswal and Dr. Nisheeth Shrivastava (Bell Labs, India) for their constant feedback. The tool would not have been possible without their guidance and support.

References

- [1] F. Roesner, T. Kohno, and D. Wetherall. Detecting and Defending Against Third-Party Tracking on the Web. In Proc. of NSDI, 2012
- [2] J. Mayer and J. Mitchell. Third-party web tracking: Policy and technology. In Proc. of IEEE Symposium on Security and Privacy, 2012.
- [3] B. Ur, P. G. Leon, L. F. Cranor, R. Shay, and Y. Wang. Smart, useful, scary, creepy: Perceptions of online behavioral advertising. In Proc. SOUPS, 2012.