An Effective Defense against Intrusive Web Advertising

Viktor Krammer [1,2]
[1] Secure Business Austria [2] Vienna University of Technology
A-1040 Vienna, Austria

E-mail: vkrammer@acm.org

Abstract

Intrusive Web advertising such as pop-ups and animated layer ads, which distract the user from reading or navigating through the main content of Web pages, is being perceived as annoying by an increasing number of users. As a response to the growing amount of extraneous content on today's Web and due to the lack of regulations imposed on abusive advertisers the author discusses the pros and cons of ad blocking, explores the different types of Web advertisements currently available and presents Quero, a novel Web browser-based content filter which implements a rule-based classifier that exploits, for example, hints present in the URL in order to classify objects as ads. Additionally, the author conducts a Web study to characterize online ads and measure the effectiveness of his solution against a manual classification. As a result, it is shown that a surprisingly small number of rules is sufficient to block almost all ads on the Web.

Keywords: *E-Commerce, Web Advertising, Content Filtering, Ad Blocking, Web Browsers, Internet Explorer*

1 Introduction

With the rapid rise of the World Wide Web in the early and mid-1990s, an unprecedented commercialization of the Internet has taken place, turning the former academic network into a mass medium for information gathering, shopping and communication. Different sources on the Web[1] state that the first paid, large-scale advertising campaign was a Web banner ad from AT&T placed on HotWired, one of the first commercial magazines on the Web, in 1994. The Web advertising business model was born and the struggle for users' attention began. Existing pay models are based on the number of ad impressions, clicks or subsequent orders triggered by the ad. Having almost no regulations other than the market and due to the technological possibilities of modern Web browsers, ads evolved from static banners to large, animated, flashing, moving, overlaying units that not only consume considerable bandwidth, but—more annoyingly—also hinder the user from reading the actual content. This latter form of intrusive Web advertising, which causes users to lose control, is cited as one of the major annoyances of today's Web [17] and has led to the development of ad blocking software such as pop-up blockers and content filters.

But how effective is banner-like advertising if it is not blocked? According to a report by ADTECH[2] the click-through rate, i.e. the ratio of the number of clicks to the number of views, has recently fallen below 0.20% on average. This means that only every 500th ad impression leads to a click; this click, however, does not guarantee that the surfer will actually buy anything. *Benway* [6] and others [9, 32, 4, 5] have discovered that Web searchers tend to ignore irrelevant information when focusing on completing a task, and named this phenomenon *banner blindness*. As a consequence of the low effectiveness of conventional Web advertising, only a very small fraction[3] of highly frequented Web sites have enough potential to actually generate sufficient revenue from advertising. Although intrusive advertising can temporarily increase the click-through rate [30], it can also negatively affect brand perception, ultimately leading to ad avoidance [19]. All of this, however, does not mean that Web advertising does not work, however. Search engine advertising, for example, is an exception since it is usually more relevant, less intrusive and presented at exactly the right time—when the user is actually searching for it. Intrusiveness, however, is quite subjective and as Yahoo! noted [27] not everyone dislikes online ads. Even the most intrusive ads are enjoyed by a minority of users. Since we cannot divide ads generally into good and bad ones, my goal in this article is to propose a solution that is able to block as many ads as possible, while simultaneously giving the user

[1] http://en.wikipedia.org/wiki/Web_banner

[2] http://www.adtech.info/en/pr-07-10.html

[3] 0.01% as estimated by Nielsen [21]

control over which types of ads to block on which sites.

While it is assumed that blocking ads is legal (at least in my country), one may ask, however, if it is also ethical. I will therefore discuss this controversial topic from both the user's and content owner's perspective [1, 22]. On the one hand, webmasters often argue that the only way to be compensated for a free service is by advertising. Exposure to ads can thus be seen as the price the user has to pay for receiving the content. On the other hand, users claim that since it is their bandwidth and computer that advertisers wish to utilize, they have the right to decide which content is displayed on it. For them ad blocking is their legitimate response to the abusive activity by advertisers, given that there are no commonly enforced regulations on the Web with respect to the amount of ads per page and their intrusiveness. Additionally, there are obviously other ways of monetizing a Web site: asking for donations (the best example of this is Wikipedia) or letting the user pay a small fee, which can be seen on news sites that ask for a fee for accessing their news archives. Annoyed users often assume that they will not buy anything through ads on Web pages (and usually never do), so by blocking these ads they actually save the sites some bandwidth and therefore money. Other reasons for blocking ads are: aesthetic considerations, eliminating distractions, and preventing privacy leaks by blocking unnecessary access to third-party tracking and profiling servers [18].

Despite this discussion, ad blocking software has for a long time been a reality in today's Internet. Google, for example, helped to abandon pop-up based advertising by including a pop-up blocker in its popular toolbar. Major security software vendors have also started to include ad filters in their Internet security suites. Obviously this has been driven by the threat that malicious advertisers can deploy spyware or viruses on a very large scale through advertising networks. A study by Finjan [26, 12] discovered that online ads indeed deliver the largest number of hacks.

In this paper I present a novel Web browser-based content filter for detecting and removing online ads. In addition to protecting users from intrusive ads, Web crawlers detect extraneous content in order to increase efficiency by skipping irrelevant content and counter services.

My work makes the following contributions:

- I show that a rule-based content filter with only a small number of rules can actually be quite effective in recognizing and removing ads. Moreover, I have implemented the proposed classifier as an add-on for Internet Explorer, and describe the architecture and "hacks" that were necessary to gain access to the JavaScript engine of the Web browser.

- In carrying out a Web study of the 500 most visited Web sites, as reported by Alexa, I classified ads with regard to their type, size and other features that might be useful for recognizing ads. Additionally, I examined the differences in ad usage among countries and measured the effectiveness of my ad blocker against a manual classification that I performed by crawling all pages and judging which items on these pages were advertisements and which were not.

- In my experiments, Quero substantially outperformed Adblock Plus, a very popular ad blocker for Firefox, with the recommended EasyList subscription.

The paper proceeds as follows. In section 2 I give an overview of the different forms of Web advertising and propose categories for ad classification. Section 3 describes my filtering solution and presents an overview of possible features that can be used to recognize ads. Section 4 then focuses on the details implementing the Web browser plugin. In section 5 I evaluate my solution against another state-of-the-art ad blocker and further study the characteristics of ads on the 500 most visited Web sites according to Alexa's Web ranking. Related work is cited in section 6. Concluding remarks are given in section 7.

2 Forms of Web Advertising

The aim of Web advertising is to attract potential customers to the advertiser's Web site and/or to strengthen brand recognition by placing promotional content and a link on other Web sites. In the constant struggle for attention, Web advertisers tend to be creative with respect to how they design and present their message. However, the requirements of the advertising mass market lead to a standardization of the formats and technology in use. Generally speaking, ads on the Web are becoming larger, more interactive and media rich, reflecting the advances in both Web browser technology and broadband Internet penetration. Before examining the currently most used ad formats, I would like to introduce the following conceptional dimensions to classify online ads:

- **Media**
  What kind of media (text, static/animated image, video, sound, 3D) does the ad use?

- **Size**
  How much screen area does the ad cover?

- **Integration**
  How tightly is the ad embedded in the primary content of the Web page? Or is the ad presented outside the main browser window?

- **Interactivity**
  Does the ad allow interaction with the user or track what the user is doing on the page?

- **Intrusiveness**
  How strongly does the ad force the user to view or interact with the ad? Does the ad deter the user from reading or navigating through the primary content? Does the ad force a delay or require interaction before the user can continue browsing the site? Does the ad use aggressive colors, flashing animations, subliminal advertising or sudden sound effects to draw attention to its message?
- **Privacy impact**
  How much information about the ad impression does the ad provider store in his logs and how long is the data kept? If the ad is delivered by an ad network, does the ad provider track individual users across different domains in order to generate user profiles? If the ad is personalized, how is the personalization performed? How does the ad provider handle personally identifiable information?

Below I give some examples of popular ad types that are currently found on the Web.

2.1 Banner Ads

Banners are integrated rectangular ad units consisting of static or animated images. Banners currently come in various standardized [13] and non-standardized sizes. Originally, banners were non-interactive and relatively unintrusive. Nano-site banners consist of a small HTML page (IFRAME) that can contain an HTML form for interaction. Flash-based banners can play animations and sound, and often interact with the user. Some intrusive Flash-based banners play a full-screen overlaid animation when they are loaded and then shrink to their original size. Some banners expand when the user hovers over them with the mouse.

2.2 Video Ads

Video ads are often found on Web sites offering online videos, and are played in front of the main video. Video ads resemble conventional TV spots and usually cannot be skipped. A newer form of video ad is integrated into ordinary Web pages like a larger banner ad. Playback is either started automatically or when the mouse hovers over the embedded video player.

2.3 Text Ads

When used sparingly, text ads (which are usually non-interactive) are the most unintrusive form of advertising. Currently almost all the revenues of Google Inc. come from text ads, which are integrated either alongside Google's search results or into its ad network of participating sites. Google's text ads are HTML-based banner ads that are available in various sizes and forms. Another approach to text ads turns related words in the primary content into links to advertiser's Web sites. These ad links are usually underlined twice to distinguish them from normal links. Additionally, a small bubble describing the underlying site appears when the user hovers over such links.

2.4 Pop-ups

Pop-ups are intrusive, non-integrated, HTML-based ads that are opened in a different browser window when the user enters or leaves a page. A slightly less intrusive form are pop-unders, which open in the background and do not acquire focus. Pop-ups are initiated by the JavaScript methods `window.open` or `window.showModelessDialog` (IE-specific). Pop-up blocking Web browser add-ons have become so popular that the feature was adopted by all major Web browsers. A newer form of pop-up is the layer ad, which opens inside the main window and has to be closed by clicking on a close button that is sometimes hard to find.

2.5 Sticky Ads

This very annoying form of integrated advertising overlays an ad of any form in a fixed position in the browser window, in such a way that it is not affected by scrolling and has to be manually closed by pressing a small close button that is often hard to find. These ads are often based on an IFRAME or DIV element that is automatically repositioned in IE, since IE still does not support the `position:fixed` CSS property.

2.6 Ad Games

A more creative and highly interactive way to lure users away from the primary content is to embed Flash-based games into Web sites. This can be combined with a lottery at the end of the game, which brings the user to a questionnaire that collects personal information.

2.7 Interstitials

Like commercial breaks on TV, interstitials are Web pages which are loaded in front of the page the user has navigated to and which contain any of the above-mentioned ad formats. In order to view the intended content, the user must either actively click on a link to proceed or wait until the end of the presentation, which makes this quite an annoying form of advertising.

2.8 Content Sponsoring

Content sponsoring is a business model that works, for example, for news sites or tourism portals where advertisers pay for the inclusion of a self-written article or a sponsored entry that does not differ much from the rest of the site. This very unintrusive and seamlessly integrated form of advertising can also be problematic if the paid content is not adequately marked as advertising.

3 Proposed Filtering Solution

Before presenting an effective filtering solution, we can make the following observations:

- According to a similar study [17], extraneous content is predominantly delivered by a host different from the one that serves the main content, e.g. a third-party ad provider such as DoubleClick.
- In many cases, ads are retrieved and integrated into the Web page by executing some client-side JavaScript. This makes it very easy for content owners to integrate ads into their sites, and also allows them to track and measure ad performance (click-through rates, click fraud, etc.) better.
- Despite the axiom of opaque URLs [7], content providers choose words or phrases for the URL that have a semantic meaning in their natural language. Reasons for this common practice are search engine optimization, easier internal organization and marking ads intentionally for easier removal. Combined with the path or subdomain structure, where ads are hosted in a specific subtree, URL-based features have the advantage that they can be used very effectively and efficiently for classification without the need of retrieving the linked content first.

3.1 Feature Selection

Taking these observations into account leads us to single out the following features, which can be both efficiently recognized and evaluated:

Media Type: The proposed solution distinguishes between the following ad types based on how they are represented in HTML or generated by script code: pop-up ads (methods `window.open`, `window.showModelessDialog`), Flash ads (OBJECT, EMBED), image ads (IMG), Google/IntelliTXT text ads (SCRIPT), and DIV or IFRAME-based layer ads.

Size: The image size derived from the width and height attributes of the IMG element.

Dynamic creation: Is the content part of the static HTML file or is it dynamically created by one of the following JavaScript methods: `document.write`, `document.writeln`, `document.createElement`, `element.innerHTML`, `element.outerHTML`, `element.insertAdjacentHTML`?

Different Domain: Is the content served from a domain different from that of the primary content?

Different Host: Is the content served from a different host but from the same first and second-level domain as the primary content?

URL tokens: The presence of certain ad-related keywords in the content URL are clues for recognizing ads. For every keyword one Boolean variable is added to our model.

The above-mentioned features have proven to be sufficient to effectively block ads on today's Web sites. Should Web advertising evolve in the future to circumvent ad blocking, we can include the following features to our model if necessary:

Target URL: In addition to the URL of the ad content, we can also analyze the anchor URL that links to the advertiser's Web site.

Presence of HTTP redirection: *Esfandiari* and *Nock* [10] have proposed a simple heuristic to detect ads with minimal user input by automatically following the links on a page and looking for HTTP redirections, whose child element is then classified as an advertisement.

DOM tree and page position: Ads are often contained in DIV layers or table cells, and are placed in preferred positions on the page.

Element attributes: CSS class names or element identifiers can give us hints about the content contained in the element.

Surrounding Text: Although it is not obligatory to label ads on the Web, many content providers do so.

Image-based analysis: Possible features that can be extracted from the object itself are size, state (static or animated), text, embedded links etc. The disadvantage of this approach is that ads must be downloaded in order to classify them.

3.2 Rule-based Classifier

Based on the evaluation provided later on and in accordance with other findings [17, 18], a rule-based classifier consisting of about 30 fine-grained rules seems to be sufficient to tackle the problem of classifying ad and non-ad content. Another advantage of our rule-based approach is that the rules and their consequences are easy to understand and updateable by normal users of the software. Below I give a conceptual overview of the rules that I have developed and tested for my ad blocking solution:

Rule 1: *Block all Flash-based content by default.* Since Flash is used almost exclusively for ads or decorational purposes, the user can decide to block this type of media by default. In order to make this restrictive rule work, it is paramount to make the blocked content easily accessible again for the user. In my proposed solution the user can either turn the ad blocker off at will, which causes a reload of the page, or selectively view the blocked content by clicking on an icon in the address bar, which appears whenever content has been removed from the current page. By clicking on the blocked content icon, a drop-down list appears with the URLs of the blocked items. In cases where the entire site is Flash-based or where Flash is used for navigation the user can add that particular site to her whitelist (see Rule 7).

Rule 2: *Block unwanted pop-ups.* We define unwanted pop-ups as unsolicited, extraneous content that is opened in a new window or tab without any user interaction such as clicking on a button or link on a Web page. Another rule that I enforce is that Web sites are only allowed to open one pop-up after each user-interaction, and that only one pop-up is allowed within the time frame of one second.

Rule 3: *Block ad banners based on their size.* I have identified common banner ad sizes and propose blocking images based on their size in pixels.

Rule 4: *Block content that comes from well-known ad providers.* This rule is implemented by matching content URLs against the domain names of well-known ad providers. This simple rule also makes it possible to identify the generating script for text ads provided by Google or IntelliTXT.

Rule 5: *Block images based on ad-related keywords in their URL.* As discussed earlier, it is possible to classify content based on its URL tokens.

Rule 6: *Block absolute-positioned DIV or IFRAME elements that are dynamically created.* This rule takes advantage of the common practice that ads on Web pages are created in separate layers that can overlay the primary content.

Rule 7: *Do not block content on sites that are whitelisted.* This gives the user the possibility of fine-tuning the ad blocker for certain sites that rely, for example, on Flash.

4 Implementation Details

I implemented the proposed rule-based ad blocker as part of the Quero Toolbar, an Internet Explorer browser helper object (BHO) [11]. Quero is a navigation bar replacement that originally leveraged the idea of searching from the address bar by combining navigation and search functionality into one toolbar.

The proposed filtering solution outlined in this article was successively incorporated into Quero during the past three years and reached full implementation in version 3.4.

Quero consists of two COM objects, one for the toolbar and one for the content filter, that are written in C++ using the ATL/WTL framework. I chose the Internet Explorer platform because of its significant market share, excellent documentation and ease of extension. The advantage of implementing an ad blocker as a Web browser extension over external proxy-based solutions is that the ad blocker has access to the Web browser's state and can be seamlessly integrated into its user interface. Extensions, however, are Web browser dependent.

4.1 Architecture

One of the challenges that had to be met was finding a feasible and effective way to block content in IE, since the extension mechanisms of IE did not foresee the rise of ad blocking software. IE thus does not support this feature directly. However, with IE 4.0 two new content extensions [20] were introduced, which were meant to add support for new URL schemes or MIME types but can be used to override the existing handlers for `http` or `text/html` respectively. Quero uses a pluggable MIME filter and interposition techniques to filter the content at the HTML level instead of inspecting only the URLs that are requested. Filtering the actual HTML code also allows Quero to remove extraneous content or rewrite pages.

4.1.1 Asynchronous pluggable MIME filter

A pluggable MIME filter is an object that implements the IInternetProtocolSink interface and is associated with a specific MIME type. Quero registers a temporary MIME filter for `text/html` that is invoked whenever IE is about to download a file that is supposed to be of that type. The MIME filter allows Quero to parse and alter the downloaded HTML code in an asynchronous manner by retrieving the data in blocks of various sizes. I have created a corrective HTML parser from scratch that can also deal with common coding mistakes and non W3C-compliant pages.

4.1.2 Interpositioning script calls

Unfortunately, the MIME handler is invoked only when static HTML files are downloaded, but not for dynamically generated HTML content that is created by client-side script. In order to filter dynamic content as well, it was necessary to hack IE and interposition the JavaScript method invocations for content or DOM tree manipulations with my own code. This was done by exploiting the COM architecture and manipulating the vtable of the interfaces IHTMLDocument2 and IHTMLElement2. I hope that Microsoft will extend the possibilities of filtering dynamic content in future versions of the IE platform.

4.1.3 URL pattern matching

In order to implement ad blocking rules 4 and 5, we must test each image URL in order to find out whether it originates from a known ad provider's domain or contains indicative keywords that are found in ads with a high degree of frequency. Both are done by pattern matching. Formally, we have a URL string U of length n, m patterns F of the form `*pattern*`, and wish to know whether U matches any pattern in F. *Palant* [23] has shown that this problem can be solved efficiently with a time complexity of O(n), which is independent of the number of filters. Instead of implementing his approach, similar to that of *Boyer-Moore* [8], I have found that most real-world filters[4], such as those used in Adblock Plus [25], are one-word patterns starting with a separator (dot or slash). I have therefore narrowed the problem definition to patterns of this form. Quero tokenizes the URL first and then matches each token against the filter set. The algorithm outlined below also gives us O(n) complexity if the filter matching in line 3 is done in constant time. This can be achieved by using a reasonably sized hash table, but is not necessary in view of the very small number of $m = 23$ at this moment.

Algorithm 1 `IsAdURL`(U, F)

```
Input: $U$ URL, $F$ set of filters
Output: true if $U$ matches one or more filters from $F$,
    false otherwise
1: $T \leftarrow$ Tokenize($U$)
2: for all $t_i \in T$ do
3:   for all $f_j \in F$ matching $t_i$ do
4:     if $f_j$ is a simple pattern or $t_{i+1}$ matches the TLD
       of $f_j$ (for domain name patterns) then
5:       return true
6:     end if
7:   end for
8: end for
9: return false
```

5 Evaluation

I carried out a Web study in order to answer the following questions:

- What percentage of popular sites on the Web display ads?
- If a page contains ads, what type of ads are used and to what extent?
- What is the proportion of ad to non-ad content on individual pages?
- Are there country-specific differences in the usage of Web advertising?
- Are URL-based characteristics suitable for identifying ads?
- What are the most frequently used banner ad sizes on the Web?
- Which ad providers and URL patterns are worth including in the filter?
- How effective and accurate is the proposed rule-based classifier in detecting ads?

I chose to study the Alexa Global Top 500 [2] list of popular Web sites to study and extracted the list of the 500 most visited domains as of 2007-02-19. Alexa monitors global Web traffic by deploying a Web browser toolbar that tracks the surfing activity of users who voluntarily agree to install it. In return, Alexa makes the aggregated traffic data publicly available through their toolbar and Web site.

To keep things practical, I limited my study to the front page of the 500 Web sites and assumed that the first page is usually quite representative with respect to page layout and ad usage. In cases of redirect or gateway pages, I manually proceeded to the main page. I used a browser-based, semiautomatic approach to crawl the pages. More specifically, I used a Windows XP machine with the latest release of Internet Explorer 6 and a modified version of the Quero add-on, which additionally wrote to a log file information about the embedded objects such as the URL, content type, size and access time. Another tool automatically visited 10 pages at a time. My task was to manually review the pages and check whether the content was correctly classified as ad and non-ad. In doing so, I did not count the explicit mentioning of Web site sponsors as extraneous content, and separated counter and tracking services from the analysis. The advantage of this browser-based approach over a classic stand-alone crawler was that the pages were loaded and rendered as intended, i.e. with JavaScript, and all the dynamically generated content was in place, which is important when studying ads. Multiple instances of the same object, identified by the same URL, were counted as one. It took me about two weeks of full-time work to visit the sites and review the logged data.

5.1 Results

Of the 500 domains visited, three were inaccessible and five brought me to another domain through a redirect or gateway page, resulting in a total of 502 pages examined. 314 (63%) of them displayed some form of advertising.

Table 1 shows the amount of ad and non-ad objects encountered and the percentage of sites making use of them.

[4] For example: `*/ads/*` or `*.doubleclick.*`

The average and median count are computed for this subset, exhibiting the given object type. Although pop-up blocking has become a standard feature of all major Web browsers, there were some sites that tried to open a pop-up ad upon being entered[5], targeting those users who have either intentionally or unintentionally turned off their pop-up blocker. Content residing in the blocked DIV layers is also counted in the other categories. In this study, I focused the evaluation primarily on the image filter because this was the part of the content filter for which I needed to find effective filter rules. For this purpose I further distinguished so-called "Web bugs", which are invisible images placed on the page for usage tracking, from the rest of the images. I identified them by their size (smaller or equal to 1x1 pixel), their query string (containing my screen resolution, for example), their referring to a known third-party Web analytics provider, their returning 404 not found HTTP status and/or their redirecting to blank.gif. Almost all sites without images were the Google home page in different languages, which occurred numerous times in the dataset. Although search results usually contain sponsored links, they were not included in the statistics since I analyzed only the landing page of each Web site.

Type	Count	Sites	in %	Avg	Median
Pop-ups	29	26	5.2%	1.1	1
Flash	690	215	42.8%	3.2	2
non-ads	110	66	13.1%	1.7	1
ads	580	182	36.3%	3.2	2
Images	15981	456	90.8%	35.0	27
non-ads	14350	452	90.0%	31.7	24
ads	1631	249	49.6%	6.6	3
Text ads	72	44	8.8%	1.6	1
Google	68	42	8.4%	1.6	1
IntelliTXT	3	3	0.6%	1	1
DIV layers	534	75	14.9%	7.1	2
Web bugs	730	230	45.8%	3.2	2

Table 1. Object Types

The next question I analyzed was: which ad characteristics are suitable for recognizing image ads? The result is summarized in table 2. In order to measure the effectiveness of each feature, I borrowed the *precision* and *recall* ratios from information retrieval:

$$precision(A \Rightarrow B) \quad = \quad P(B|A) = \frac{|\{A \wedge B\}|}{|\{A\}|} \qquad (1)$$

$$recall(A \Rightarrow B) \quad = \quad P(A|B) = \frac{|\{A \wedge B\}|}{|\{B\}|} \qquad (2)$$

where A is a property implying B. In other words, in our context the object is an ad, and $\{X\}$ denotes the set of objects satisfying condition X. Precision is also called *confidence* in association rule mining.

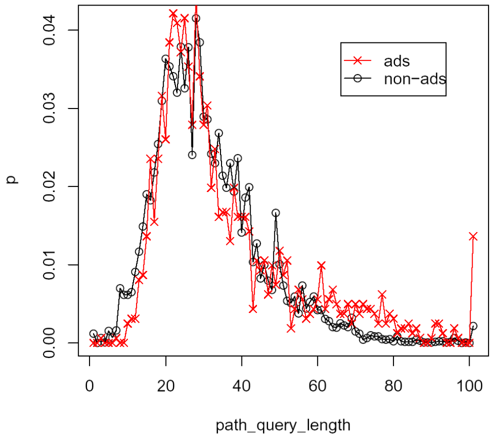

Figure 1. Distribution of URL path & query length, right-most points depict $P(length \geq 100)$

Interestingly, neither the fact that an image is "script generated"[6] nor served from a different host than that of the current Web site gives us enough confidence to classify the image as an ad. The same applies to image URLs containing a query string or its size in its filename. Also, an analysis of the distribution of the URL length *modulo* the domain name did not reveal any significant difference between ad and non-ad URLs, as illustrated in figure 1. A very small number of certain keywords, however, can give as a very good clue about the nature of the images.

Feature	Precision	Recall
Script generated	28.5%	32.1%
Different 2nd level domain	15.0%	30.6%
Different 3rd level domain	13.8%	83.6%
Query string in URL	34.4%	13.3%
Image dimensions in URL	34.9%	28.6%
Ad pattern in URL	95.5%	63.3%

Table 2. Characteristics of Image Ads

Based on my experience with online ads, I compiled a list of ad-related keywords and tested them against the dataset. The top 10 keywords are shown in table 3. For the classifier I chose 23 indicator words, including some well-known ad providers' domain names, that already cover two-thirds of all banner ads.

[5] pop-ups triggered by leaving the page were not measured

[6] "Script generated" means that the image is output directly by means of JavaScript instead of being part of an HTML file, which can, however, also be retrieved by script.

Keyword	Precision	Recall
ads	98.0%	31.5%
banner	87.0%	16.4%
adv	79.5%	6.6%
click	76.5%	4.7%
upload	15.2%	4.2%
adimages	100.0%	3.3%
banners	94.4%	3.2%
doubleclick	100.0%	2.9%
adimg	79.5%	2.2%
adserver	100.0%	2.1%

Table 3. Keyword Analysis

Additionally, banners are recognized by their size. The most frequently used ad dimensions are shown in table 4.

Width	Height	Precision	Recall
106	50	98.7%	6.0%
300	250	100.0%	5.7%
120	60	80.72%	5.4%
728	90	100.0%	5.0%
468	60	100.0%	4.4%
88	31	56.52%	4.2%
114	23	80.85%	3.0%
120	90	14.91%	1.4%
186	47	80.0%	1.3%
120	600	100.0%	1.3%

Table 4. Banner Dimensions

What is noteworthy here is that Chinese Web sites, unlike those in the U.S., do not adhere so strictly to standardized banner sizes and are filled with almost three times as many ads, albeit smaller ones. An analysis by country is given in table 5. All Web sites were manually associated with their main target country based on the following indicators: TLD, language, visitor's origin, Web site's imprint, admin-c and server location.

Finally, I compared the overall filter results of Quero with those of Adblock Plus (ABP), one of the most popular extensions for Firefox. Since ABP only provides the framework for blocking ads in Firefox, I performed the test against two recommended sets of filters. One of these, EasyList, is the number one subscription for ABP at the moment. In addition to URL-based filters, ABP also supports so-called element hiding rules, which depend on HTML element types and attributes which I intentionally removed from the "Dr. Evil" set of filters because they were hard to simulate with the available data. To make the results comparable, I also did not apply Quero's HTML element rule 6, and assumed that the pop-up blocker of Firefox would block all pop-ups. The final results are summarized in table 6. Surprisingly, Quero's few but nevertheless effective rules significantly outperform the market leader in ad blocking, recognizing almost a third more ads with even a slightly higher degree of precision.

6 Related Work

Ad blocking is somewhat related to email *spam filtering* and automatic Web *content classification* in general. In order to prevent ads from being downloaded, ad blockers usually rely on the URL and other meta-information to predict the nature of the content. *Kan* and *Thi* [15, 14] have demonstrated that the URL is indeed a very good predictor for classifying Web pages and is almost as effective as using the text itself.

From an architectural point of view, ad blockers can be implemented as an extension to the Web browser, as an HTTP proxy running on the local system, or by otherwise intercepting HTTP requests on a system-wide level. Browser-based filters have the advantage of being able to integrate themselves smoothly into the UI of the browser and having access to its internal state and services. However, such plug-ins are browser dependent and usually hard to port to other browsers.

As mentioned earlier in this article, Adblock Plus (ABP) [24] is currently the most popular and advanced ad blocker for Mozilla Firefox and is one of the most downloaded add-ons for this browser. It is a complete rewrite of the original Adblock extension [31] that certainly helped increase the popularity of Firefox. ABP allows the user to specify a set of URL patterns that can include wildcards or be specified as regular expressions. In addition to URL-based rules, ABP also supports the hiding of entire HTML elements based on their element type, attributes and the site on which they appear. Unlike Quero, ABP currently[7] does not allow users to selectively unblock blocked objects and leaves the creation of filter rules to a couple of enthusiastic users, who publicly offer their filter sets as ABP subscriptions.

Although *Rowe* et al. reported that they implemented a banner ad blocker as a Java servlet-based HTML filter [28], they have not yet published any large-scale study of their filter. They use a linear model and eliminate images when a certain threshold is exceeded. Among their criteria are image size, words in the URL, the image's "alt" description and the text around the image.

Shih and *Karger* [29] take advantage of the tree structure of the URL by assuming that ad content is usually stored in a separate subtree on the Web server. Additionally, they take table layout into account and estimate the probability of each table cell containing ads based on the probability of

[7] as of version 0.7.5.1

Country →	us	cn	jp	de	tw	uk	hk	br	cz	vn
Sites	188	84	23	16	15	11	9	9	9	8
in %	37.5%	16,7%	4.6%	3.2%	3.0%	2.2%	1.8%	1.8%	1.8%	1.6%
Sites with ads	103	69	16	10	11	7	5	5	8	7
in %	54.8%	82.1%	69.6%	62.5%	73.3%	63.6%	55.6%	55.6%	88.9%	87.5%
# Ads	557	1071	74	77	148	52	26	52	38	194
Avg per site	5.4	15.5	4.6	7.7	13.5	7.4	5.2	10.4	4.8	27.7
Ad Pixels										
Avg per site	179,442	211,459	99,480	195,174	193,093	188,719	264,134	77.966	147,705	605,949
Avg per object	55,888	31,588	35,421	42,444	19,544	36,419	70,381	25,994	86,316	21,208
Image Filter										
Precision	96.5%	98.7%	98.0%	96.9%	95.7%	88.9%	94.1%	100.0%	100.0%	94.9%
Recall	97.1%	73.9%	94.1%	91.2%	82.6%	94.1%	100.0%	100.0%	50.0%	83.6%

Table 5. Ads per Country

Filter	Type	Count	Ads	Blocked	FP	Precision	Recall
Quero Version 3.4	Pop-ups	29	29	29	0	100.0%	100.0%
	Flash	690	580	690	110	84.1%	100.0%
	Images	15981	1631	1454	52	96.4%	86.0%
	Text	71	71	71	0	100.0%	100.0%
	Overall	16771	2311	2244	162	92.8%	90.1%
Adblock Plus Filter: EasyList 495 rules (2007-06-07)	Pop-ups	29	29	29	0	100.0%	100.0%
	Flash	690	580	316	4	98.7%	53.8%
	Images	15981	1631	1103	151	86.3%	58.4%
	Text	71	71	71	0	100.0%	100.0%
	Overall	16771	2311	1519	155	89.8%	59.0%
Adblock Plus Filter: Dr. Evil 525 rules (2007-06-05)	Pop-ups	29	29	29	0	100.0%	100.0%
	Flash	690	580	204	1	99.5%	35.0%
	Images	15981	1631	792	150	81.1%	39.4%
	Text	71	71	71	0	100.0%	100.0%
	Overall	16771	2311	1096	151	86.2%	40.9%

Table 6. Filter Results and Comparison

its parent nodes. The learning problem is then modeled as a Bayesian net, which is then trained on either the output of Webwasher, a commercial ad blocker, or on the following heuristic. The content within links that redirects to another site is assumed to be an advertisement. The drawback of this heuristic, however, is that each link has to be accessed to check the server's response for a possible redirection. The authors tested their tree-based URL classifier and their heuristic against Webwasher on 25 sites and achieved an accuracy similar to that of Webwasher. Although a CSS-based layout was not mentioned in the article, the approach could be extended to work on layer-based Web sites as well.

A browser-based filter [10] that learns to block images and Flash animations based on minimal user feedback was designed and implemented as a browser extension for Mozilla Firefox by *Esfandiari* and *Nock*. The filter is based on a weighted majority algorithm that uses tokens of the URL separated by '/' as predictors, and is trained by the user, who right-clicks on images she wishes to block.

In addition to the above-mentioned academic content filters, there are many commercial and freeware products available. These include proxy-based or browser-independent solutions such as Webwasher[8], Proxomitron[9] and AdMuncher[10]. Ad blockers have also become part of security and personal firewall software like Norton[11] and Kaspersky[12] Internet Security products. Even more browser extensions and toolbars exist for blocking pop-ups. Pop-up blockers became so widespread that this feature was eventually included in all major Web browsers. While I doubt that the same will happen to ad blockers in the near future, Opera has already included a simple URL-based content filter in Opera 9 that can be used to block ads but must be manually configured by the user.

Finally, I would like to present two ad-related Web studies. *Krishnamurthy* and *Wills* identified and studied two recent Web annoyances [17]. The first is Web advertising and the second is the increasing number of Web sites that require the user to register for free in order to access the site or services within. In order to study the first annoyance, they crawled approximately 1,200 top sites cited by Alexa in 12 chosen categories, and additionally included all sites from the Global Top 500 list. However, instead of manually identifying extraneous content, as I did in my study, they used Adblock's Filterset.G[13] to automatically classify content. The focus of their study was on analyzing the amount of ads per category, the ad server distribution and the performance impact of ad delivery. The most noteworthy aspect of the results is that they show a reduction of 57% of servers accessed or a reduction of about one-third of the median download time, when ads are blocked. Moreover, it was shown that the majority of ads were delivered from only a small number of ad servers, which are blocked by matching only a small subset of rules from the tested filterset—a statement which is consistent with my findings.

Another study, conducted by *Bacarella* et al., focused on finding ad or tracking servers through traffic data analysis [3]. Web site usage here is modeled by a weighted traffic graph, and heuristics are used to identify ad or tracking servers. The authors assume that the higher the relative traffic (i.e. the probability that the user will click on an ad compared to other links on the page), the higher the probability that the page contains advertising. Although this assertion is questionable, it does apply to intrusive advertising that lures the user into clicking on the ad before anything else. Unfortunately, low traffic ads were not analyzed due to the limits of the gathered dataset.

7 Conclusion

I have presented an effective rule-based classifier for recognizing ads on the Web and implemented the proposed method in the Quero add-on [16] for Internet Explorer, which runs on Windows 98 or later and is available as freeware. While the IE platform does not explicitly support the blocking of ads, I have found ways to gain access to the internal state of the browser. Nevertheless, I hope that Microsoft will include better interface support for content filtering in future versions of IE.

By conducting a Web study of the 500 most visited Web sites reported by Alexa, I have found that only a very small number of rules is sufficient to block almost all ads. I have shown that the URL is still the most promising indicator for blocking ads without the need of downloading them first. In addition to the URL, Quero also takes into account other characteristics such as element type, size and whether the content is statically or dynamically created. At the same time, Quero makes it easy to access blocked content by clicking on an icon in the address bar that pops up whenever content is removed from the current page. While on average two-thirds of all Web sites display some form of advertising, I have found that there are country-specific differences, with the trend that Asian sites are more ad-laden than North American ones.

Although ad blockers are becoming increasingly popular, the consequences of this are not yet clear. While it is currently easy to identify and remove ads on the Web, I anticipate that the cat-and-mouse game between advertisers and annoyed users seeking ways to block ads will continue.

[8] http://www.webwasher.com/

[9] http://www.proxomitron.info/

[10] http://www.admuncher.com/

[11] http://www.symantec.com/

[12] http://www.kaspersky.com/

[13] This is a set of URL patterns provided by a user called "G" at http://www.pierceive.com/. This filterset is disputed, however, because of its excessive use of regular expressions that make the list difficult to maintain.

Advertisers and content owners could circumvent ad blocking software by obfuscating the URL and other characteristic features of ads, make ads even more indistinguishable from content, or develop ad blocker detection scripts that forbid access to sites unless the ad blocker is deactivated. But then again, it could be possible to block such scripts and develop filter rules on a per site basis. In the long run, site owners should try to deliver fewer ads, but ones that are more relevant and unintrusive to their user base, so that Web advertising is once again perceived as adding value.

Acknowledgements

This work was performed at the Secure Business Austria research center funded by the Federal Ministry of Economics and Labor of the Republic of Austria (BMWA) and the federal province of Vienna.

References

[1] Is Ad Blocking Ethical? http://www.adblock.org/2006/02/is-ad-blocking-ethical/, 2006.

[2] Alexa Traffic Rankings: Global Top 500 (accessed 2007-02-19). http://www.alexa.com/site/ds/top_sites?ts_mode=global, 2007.

[3] V. Bacarella, F. Giannotti, M. Nanni, and D. Pedreschi. Discovery of ads web hosts through traffic data analysis. In *Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 76–81. ACM Press, June 2004.

[4] M. Bayles. Just How 'Blind' Are We to Advertising Banners on the Web? http://psychology.wichita.edu/surl/usabilitynews/2S/banners.htm, July 2000.

[5] M. E. Bayles. Designing online banner advertisements: Should we animate? In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves (CHI 2002)*, pages 363–366. ACM Press, April 2002.

[6] J. P. Benway and D. M. Lane. Banner Blindness: Web Searchers Often Miss "Obvious" Links. http://www.internettg.org/newsletter/dec98/banner_blindness.html, December 1998.

[7] T. Berners-Lee. Universal Resource Identifiers – Axioms of Web Architecture. http://www.w3.org/DesignIssues/Axioms.html#opaque, 1996.

[8] R. S. Boyer and J. S. Moore. A fast string searching algorithm. *Communications of the ACM*, 20(10):762–772, October 1977.

[9] M. Burke, A. Hornof, E. Nilsen, and N. Gorman. High-cost banner blindness: Ads increase perceived workload, hinder visual search, and are forgotten. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(4):423–445, December 2005.

[10] B. Esfandiari and R. Nock. Adaptive filtering of advertisements on web pages. In *Special interest tracks and posters of the 14th international conference on World Wide Web (WWW '05)*, pages 916–917. ACM Press, May 2005.

[11] D. Esposito. *[MSDN] Browser Helper Objects: The Browser the Way You Want It*. Microsoft Corporation, January 1999. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwebgen/html/bho.asp.

[12] Finjan Malicious Code Research Center. Web Security Trends Report Q1/2007. http://www.finjan.com/GetObject.aspx?ObjId=375, March 2007.

[13] Interactive Advertising Bureau. Ad Unit Guidelines. http://www.iab.net/standards/adunits.asp, May 2007.

[14] M.-Y. Kan. Web page categorization without the web page. In *Proceedings of the 13th international conference on World Wide Web (WWW '04)*, pages 262–263. ACM Press, May 2004.

[15] M.-Y. Kan and H. O. N. Thi. Fast webpage classification using url features. In *Proceedings of the 14th ACM international conference on Information and knowledge management (CIKM '05)*, pages 325–326. ACM Press, October 2005.

[16] V. Krammer. Quero Toolbar. http://www.quero.at/, 2007.

[17] B. Krishnamurthy and C. E. Wills. Cat and mouse: Content delivery tradeoffs in web access. In *Proceedings of the 15th international conference on World Wide Web (WWW '06)*, pages 337–346. ACM Press, May 2006.

[18] B. Krishnamurthy and C. E. Wills. Generating a privacy footprint on the internet. In *Proceedings of the 6th ACM SIGCOMM on Internet measurement (IMC '06)*, pages 65–70. ACM Press, October 2006.

[19] S. McCoy, A. Everard, P. Polak, and D. F. Galletta. The effects of online advertising. *Communications of the ACM*, 50(3):84–88, March 2007.

[20] Microsoft Corporation. *[MSDN] About Asynchronous Pluggable Protocols*, 2007. http://msdn2.microsoft.com/en-us/library/aa767916.aspx.

[21] J. Nielsen. Why Advertising Doesn't Work on the Web. http://www.useit.com/alertbox/9709a.html, September 1997.

[22] W. Palant. Ethics of blocking ads. http://adblockplus.org/blog/ethics-of-blocking-ads-part-3, 2006.

[23] W. Palant. Investigating filter matching algorithms. http://adblockplus.org/blog/investigating-filter-matching-algorithms, 2006.

[24] W. Palant. Adblock Plus. http://adblockplus.org/, 2007.

[25] W. Palant. Known Adblock Plus subscriptions. http://adblockplus.org/en/subscriptions, 2007.

[26] PC World. Online Ads Deliver Most Hacks. http://www.pcworld.com/article/id,130129-pg,1/article.html, March 2007.

[27] C. Rohrer and J. Boyd. The rise of intrusive online advertising and the response of user experience research at yahoo! In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves (CHI 2004)*, pages 1085–1086. ACM Press, April 2004.

[28] N. C. Rowe, J. Coffman, Y. Degirmenci, S. Hall, S. Lee, and C. Williams. Automatic removal of advertising from web-page display. In *Proceedings of the 2nd ACM/IEEE Joint Conference on Digital Libraries (JCDL 2002)*, page 406. ACM Press, June 2002.

[29] L. K. Shih and D. R. Karger. Using urls and table layout for web classification tasks. In *Proceedings of the 13th international conference on World Wide Web (WWW'04)*, pages 193–202. ACM Press, May 2004.

[30] S. Shrestha. Does the Intrusiveness of an Online Advertisement Influence User Recall and Recognition? `http://psychology.wichita.edu/surl/usabilitynews/81/OnlineAds.htm`, February 2006.

[31] H. A. Sorensen, rue, B. Karel, and M. McDonald. Adblock. `http://adblock.mozdev.org/`, 2003.

[32] C. Y. Wong. Is banner ads totally blind for us? In *CHI '01 extended abstracts on Human factors in computing systems*, pages 389–390. ACM Press, March 2001.