WEBREVIEW: AN INTELLIGENT CLASSIFICATION PLATFORM TO AUTOMATE THE EVALUATION AND RANKING OF WEBSITE QUALITY AND USABILITY USING ARTIFICIAL INTELLIGENCE AND WEB SCRAPING TECHNIQUES

Darren Xu[1], Dexter Xu[1] and Ang Li[2]

[1]Redlands High School, 840 E Citrus Ave, Redlands, CA 92374, USA
[2]California State University, Long Beach, 1250 Bellflower Blvd, Long Beach, CA 90840, USA

ABSTRACT

Paywalls are a staple of the internet and seen in a vast amount of websites [1]. Encountering a paywall is always annoying, whether you're doing work for school or just trying to catch up on the latest news [2]. To eliminate this annoyance we have created Wall Breaker, a google extension with the primary task of bypassing any paywall using a variety of methods [3]. Our extension uses methods such as opening the website in an incognito tab or acting as a new user when clicking on a link. Although not the first of its kind, our extension is truly unique in the methods and techniques used. The popup used is easy to use and simple to look at, providing the best user experience. Wall Breaker will work on most websites, both popular and lesser known ones. It makes no distinction between certain types of websites and the methods can be used on any page. While Wall Breaker might not work on every website those are few and far between.

KEYWORDS

Web Scraping techniques, Google, paywall.

1. INTRODUCTION

Web-browsing and surfing the web have become a highly prevalent part of all aspects of society [4]. Many people wander into the realm of the internet to browse a variety of websites daily. The use of websites is integral and goes hand in hand with the internet. Whether their purpose is for research, entertainment, or other interests they have undoubtedly crossed paths with a paywall. Many websites hinder people's access to their website by requiring some form of payment, login, and or agreement. This is known as a paywall. Popular websites might use paywalls to restrict the users entry and cater towards those that have some kind of subscription. These confining stipulations deter possible enthusiasts and might cause annoyance in those who only wish to view the website a few times and do not want to sign up for a subscription of any kind. Getting around paywalls such as this opens up more information for internet users and improves the overall internet experience. It benefits the consumer by saving time, headache, and increasing the overall spread of knowledge and information to everyone. The ability to bypass a paywall is invaluable in order to increase access to information that otherwise would be barred from the normal viewer.

David C. Wyld et al. (Eds): CMC, SOFT, MLT, SAI - 2022
pp. 37-44, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120504

The benefits of our extension, WallBreaker, opens up many different websites and minimizes the annoyance of everyday browsing. Because of how ingrained web-browsing is in our daily lives, any way to improve upon the now common activity is important.

Any internet browser has most likely encountered a paywall [5]. While certainly annoying, there are ways to get past one. A common use of paywalls is letting the user access a specific amount of pages and once that number is reached, activate the paywall preventing the user from reaching any new pages. While there are some manual ways to get by this such as using a friend's account or making a new one this is far from optimal. A much more efficient way to bypass a paywall is to do it automatically by using internal tools such as browser extensions. Our extension is not the first to attempt to fix this problem. There have been several different techniques and methods used by others in order to bypass paywalls. A popular method to bypass a paywall is to search the internet for a free version of that particular website by examining the contents of the page and searching the internet for a copy. However a common problem with this method is that oftentimes, a free copy is unable to be found on the internet. These limitations limit the amount of paywalls able to be bypassed. For example, a lesser known website with a paywall will most likely not have a free copy and even popular news articles and websites might not have any free versions available online. Most of the time free versions of websites with paywalls will be limited to scholarly articles. These methods also rely on already established databases that hold free versions of these websites.

We used a number of methods to achieve our goal of bypassing paywalls. The first step to achieving this goal is by making it easily available, we decided to create this project in the form of a google chrome extension. This extension has a brilliant UI designed to customize the user's needs. To add on, we created multiple options to make sure we can bypass all paywalls on all websites. Some examples include opening a private browser when hitting a paywall or disguising the user as a google bot beforehand. The functionality of each websites' paywall differs which results in the need for differing methods to bypass these obstacles. Some for example do not allow google bots but may allow social media links. We therefore have a method that disguises the user as coming from a social media platform to trick the website into allowing access to their content without signing up or paying for an account. Some websites' paywalls might be harder than others to traverse through. The inner workings of websites all over the web varies and their paywalls all need different requirements to bypass. This is why we required so many different solutions. Our solution has the strength of accessing the desired data directly. Other solutions such as unpaywall, while still a strong solution, reroute you to a different location where the data may possibly be stored. Our extension is also free to use, unlike some of the other solutions currently out there.. Due to these numerous features, we believe that our creation is invaluable while surfing the web. Our project is important to minimize the frustration of online browsing [6].

To prove our results we went to a series of websites and tested if we would be met with a paywall. Certain websites such as medium or wall street journal, contained paywalls with which we could run tests. To reliably test the results of our extension we tested the code manually, checking a selection of sites that offer a different variety in the types of paywalls that are used. For example, we tried our code against medium.com to test the functionality of the code. At each turn, our code reliably blocked the paywall. Our code has numerous working features, such as personally selecting certain websites to block or unblock. We tested this by using it on a variety of websites. Another feature is the ability to adapt to all kinds of paywalls by using a diverse set of methods built for a multitude of paywalls. These features are inclusive to all websites. Basically our extension is built for user customization and each function is able to work independently and as a whole. Ultimately our code works together to provide a better experience surfing the internet.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during back-end and front-end coding process; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the project we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Creating an easy to use UI inside the pop-up

Creating an easy to use UI inside the pop-up was one of the main challenges when building our extension [7]. We wanted to create a pop-up that the average user could understand and use without difficulty [8]. As Wall Breaker has multiple methods to bypass paywalls we also needed a way to easily explain the different methods to bypass different types of paywalls. We decided to incorporate elements such as sliders and a simple design to create our pop-up.

2.2. Finding methods to bypass paywall

As each paywall is different it makes sense to have multiple methods to bypass them. Some of these methods only work for specific circumstances so putting on the solutions in one location was necessary but it proved to be a challenge. We had to create a script that had all the methods in one palace in order to integrate it all in the chrome extension. We used sliders to activate the methods. Making sure to save user settings and database info was also a challenge due to the fact that the sliders naturally reset each time the popup is opened. We thus had to create a series of code to make sure the settings changed.

2.3. Testing code

One of the more difficult parts of the code was actually testing it. Due to the nature of the extension it is basically impossible to see if the code works until we actually finish coding it, meaning that if there is an error it will most likely be unknown until we actually attempt to run the code. Chrome API communication between scripts leads to limitations in certain background tasks, making the process of building the extension much more restricted and difficult.

3. SOLUTION

There are many steps and components that make up the functionality of our project [9]. When the user first opens the extension they are greeted with an easy to use, and easy to understand user interface. This interface requests for four different types of inputs, which allow the user to customize their predilections and personalize their time online [10]. With one click you can personalize your online experience by excluding or including certain websites. Once the input is entered it is saved in a database that stores the user's preferences. The preferences are frequently updated with any new changes to the inputs that are made. To bypass the paywall our program uses a content script and a background script which work together in order to avoid the paywalls. The purpose of the content script is to find the best way to traverse through a paywall by going through numerous different strategies that might be implemented in passing the paywall. To elaborate, the content script finds what kind of website is opened-because every website utilizes a paywall differently- and observes the best solution in order to dodge the paywall. The content script is constantly being updated with each click into a new website, Meanwhile the background script is refreshed only when the extension first runs. In order to continually refresh the background script, the background script is directly linked to the change of inputs from the user. The extension then returns the most optimal solution to bypass the paywall while simultaneously saving the user's settings into a database. Based on the two scripts and the input the extension customizes the user's experience on the web. Each of these individual mechanisms helps to run and produce the end result.

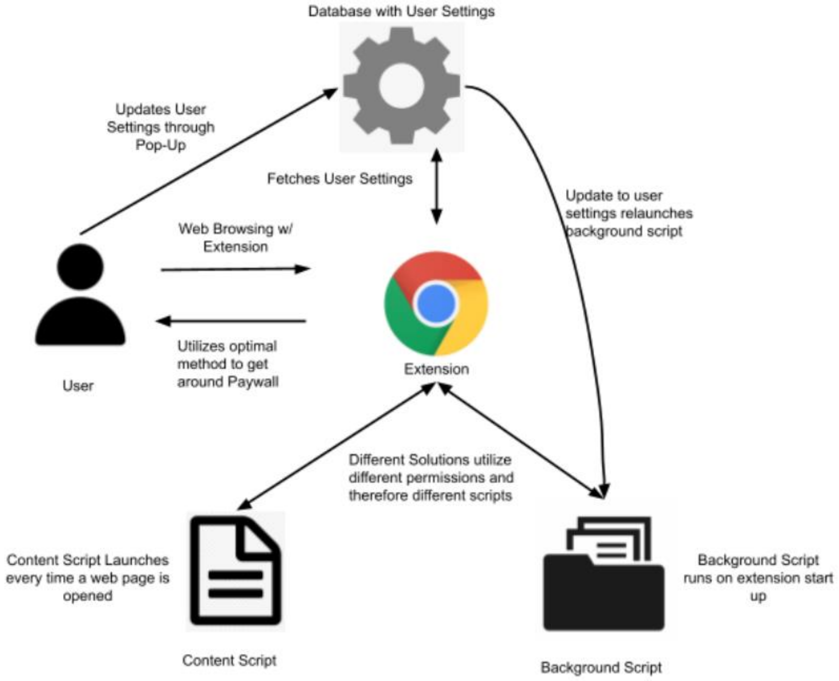
Database with User Settings
Updates User Settings through Pop-Up
Fetches User Settings
Update to user settings relaunches background script
Web Browsing w/ Extension
Utilizes optimal method to get around Paywall
User
Extension
Different Solutions utilize different permissions and therefore different scripts
Content Script Launches every time a web page is opened
Content Script
Background Script runs on extension start up
Background Script

Figure 1. An overview of the project

Google Chrome Extensions are programs that change the browser in some way, whether it's the way Google looks or the way that the search bar works. Our extension manipulates the source code of websites to bypass paywalls. Wall Breaker uses multiple scripts that communicate and rely on each other to work. The content script checks the information of a website and then gives that information to other scripts to execute the code. The background script saves the user preferences on whether the sliders remain enabled or disabled. The manifest saves information about the extension like the name and icon used. Our extension uses four methods to bypass paywalls. These methods can be switched on and off with sliders. Multiple methods are used as some might not work on specific websites. Many websites let you view their page a couple of times before enabling a paywall and making you create an account. One of our methods automatically opens the website in an incognito page, making it seem as if one is visiting the website for the first time. Some websites disable the private browsing tab however so we also have a method that disables cookies on the website and one that modifies the HTTP headers to make it seem as if one is joining from a social media site. To add on, we also have a method that takes advantage of a Google bot where it changes the website request to make the user look like a GoogleBot. All these methods are unique in their own ways and help the user in bypassing paywalls on different websites in the form of a Google Extension.

4. EXPERIMENT

4.1. Experiment 1

In order to verify that our solution can effectively solve problems at different levels and have good user feedback, we decided to select multiple experimental groups and comparison groups for several experiments. For the first experiment, we want to prove that our solution works stable and continuously, so we choose a group size of 20 different IP Address in 5 different country The goal of the first experiment is to verify if the paywall detection works good for different IP address Through sampling 5 groups of users of different IP address and letting them try the same website 10 times. Results are collected by statistics of the total time that the paywall website is detected. Experiments have shown that all IP in different IP groups show a high rate of detection. IP form North American has the most high rates, IP form China has the most Low rates This experiment could explain that the IP address do have a obvious impact on the detection results The experiment graph shows below:

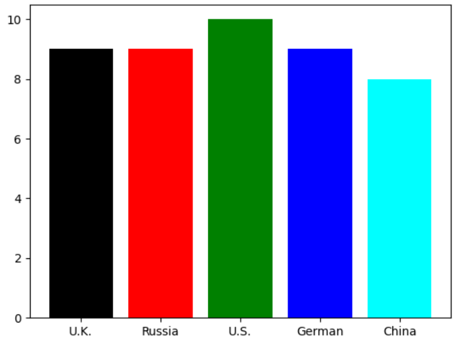

Figure 2. Survey result

4.2. Experiment 2

A good user experience is as important as a good product. So a perfect solution should have excellent user experience feedback. In order to prove that our solution has the best user feedback, we specially designed a user experience questionnaire We statistics the feedback result from 100 users, we divide those users into three different groups. The first group of users spend more time on the video games, the second group of users spend more time on the research and reading, the third group of users spend more time on working The goal of the first experiment is to verify high feedback scores shows high performance We collect the feedback scores form these 3 different group of users and analyze it. Experiments have shown that users who play games more give the highest result feedback to our app. Which may because of paywall link appears more in the game searching The experiment graph shows below:

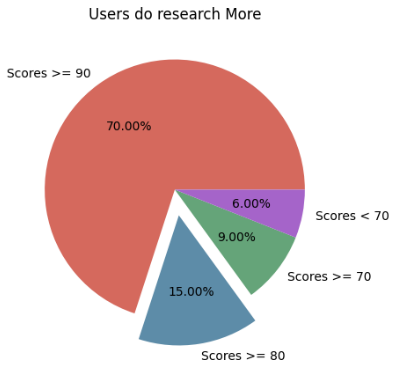

Figure 3. Survey result 2

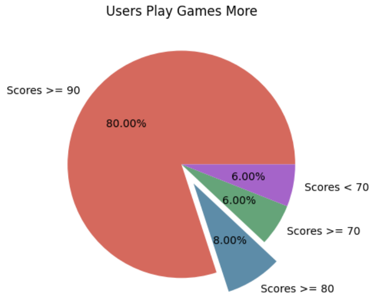

Figure 4. Survey result 3

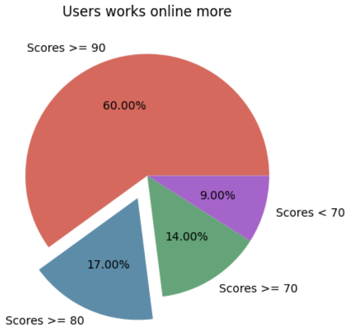

Figure 5. Survey result 4

5. Related Work

Schultz and Azadbakht have a browser extension that allows the user to use "tools" that search for an open, free version of a website [11]. When encountering a paywall there are four options that show up on the screen that link to other websites where the user can read the article for free. Similar to our extension, the project extension uses tools that users click and use to gain access to websites. However, their extension relies on other already established websites that have databases of free versions of websites. Our extension actually manipulates the code of a website and has a variety of different methods to bypass the paywall. For example, a lesser known website with a paywall might not have a free version but our extension can instead use a variety of methods to actually manipulate the code and gain access to the website.

Unpaywall is a widget that when encountering a paywall searches the internet for a free version of the website [13]. When encountering a paywall a tab will show up on the side of a screen that links to a free version or is grayed out, meaning it could not find a free version. Unpaywall is similar to our extension in that the user can interact with it. Of course, like the previous related work Unpaywall has restrictions and limitations of what it can do when encountering a paywall. Unpaywall searches for free versions of the website on the internet but if no such website is found the user has no choice but to find a different website or pay for it.

Libkey Nomad is a browser extension that finds free and full-text versions of journal articles [12]. Nomad examines the contents of an article page and searches their databases for a PDF of the article. Depending on if a PDF is found or not Nomad has several options for the user. For example, if a PDF is found the user can download the PDF and if a PDF is not found the user will be linked to a website where they can enable interlibrary loan. Libkey Nomad is limited to mostly scholarly articles but can help in accessing library subscriptions. Libkey Nomad is great for finding articles that need a library subscription to access. Our extension can access most websites and bypass the paywalls but might have trouble getting pass a subscription.

6. CONCLUSIONS

Encountering a paywall is always irritating to deal with especially when trying to do something important, like school work. Although bypassing a paywall manually is possible in some cases, such as creating a new account for a website, it is far from optimal and efficient. To effectively avoid encountering paywalls we have created Wall Breaker, a google extension that bypasses paywalls on the internet [14]. We used a variety of ways and methods to bypass paywalls. We added these methods into an easy to use popup that users can interact with. Users can turn these methods on and off and the popup will save their choices. One of the methods we use opens a page in the incognito tab. If the user does not want this to happen they can turn off this method permanently by clicking the slider. We used a variety of methods since the effectiveness of a method depends on the website. A method could work perfectly fine on a website but be completely ineffective on another [15]. The creation of multiple methods and functions lower the chance of running into a website with a paywall that can not be bypassed. The popup we created is simplistic to make sure the user can easily understand and use it. The popup also hides itself until the user interacts with it by clicking the icon on the taskbar. This makes sure that our extension is not distracting to the user. Wall Breaker is extremely effective in doing its task and makes sure that the user receives the best possible experience.

While our extension is extremely flexible there are some limitations associated with it. Our extension works on most websites but there might be a few occasions where none of our methods will work and there is no way to get by the paywall. Although our popup is easy to use it could be annoying to turn the sliders on and off depending on the website and trying to figure out which option is the best. We have multiple methods but the effectiveness of these methods vary from website to website and some can be annoying to leave on permanently such as the one that opens the website into an incognito page.

Most of the problems with our extension can be fixed by changing or adding new functions and code. We can change the code so that when encountering a paywall, a method will be automatically used without the need for user input. We can also add new and more methods to make it even less likely for there to be a paywall that can not be bypassed.

REFERENCES

[1] Estok, David. "Paywalls." Journal of Professional Communication (2012).

[2] Chiou, Lesley, and Catherine Tucker. "Paywalls and the demand for news." Information Economics and Policy 25.2 (2013): 61-69.

[3] Carlini, Nicholas, Adrienne Porter Felt, and David Wagner. "An evaluation of the google chrome extension security architecture." 21st {USENIX} Security Symposium ({USENIX} Security 12). 2012.

[4] Lieberman, Henry, Neil W. Van Dyke, and Adrian S. Vivacqua. "Let's browse: a collaborative Web browsing agent." Proceedings of the 4th international conference on Intelligent user interfaces. 1998.

[5] Muglerab, Emily, et al. "Control of an internet browser using the P300 event-related potential." International Journal of Bioelectromagnetism 10.1 (2008): 56-63.

[6] Kapoor, Ashish, Winslow Burleson, and Rosalind W. Picard. "Automatic prediction of frustration." International journal of human-computer studies 65.8 (2007): 724-736.

[7] Hoiem, Derek, Alexei A. Efros, and Martial Hebert. "Automatic photo pop-up." ACM SIGGRAPH 2005 Papers. 2005. 577-584.

[8] Greco, JoAnn. "From pop-up to permanent." Planning 78.9 (2012): 14-18.

[9] Wold, Svante, Kim Esbensen, and Paul Geladi. "Principal component analysis." Chemometrics and intelligent laboratory systems 2.1-3 (1987): 37-52.

[10] Reinhart, Tanya. "Interface strategies." OTS working papers in linguistics (1995).

[11] Azadbakht, Elena; Schultz, Teresa.Information Technology and Libraries (Online); Chicago Vol. 39, Iss. 2, (Jun 2020): 1-13. DOI:10.6017/ital.v39i2.12041

[12] Hoy, Matthew B. “LibKey Nomad.” Journal of the Medical Library Association : JMLA vol. 108,4 (2020): 672-674. doi:10.5195/jmla.2020.1017

[13] Chawla, Dalmeet Singh. "Unpaywall finds free versions of paywalled papers." Nature News (2017).

[14] Pattabhiramaiah, Adithya, S. Sriram, and Puneet Manchanda. "Paywalls: Monetizing online content." Journal of marketing 83.2 (2019): 19-36.

[15] Mintzer, Fred, Gordon W. Braudaway, and Minerva M. Yeung. "Effective and ineffective digital watermarks." Proceedings of International Conference on Image Processing. Vol. 3. IEEE, 1997.