Automatic Detection and Banning of Content Stealing Bots for E-commerce

Nicolás Poggi, Josep Lluis Berral, Toni Moreno, Ricard Gavaldà and Jordi Torres

Universitat Politècnica de Catalunya, Barcelona, Spain.

Contact email: npoggi@ac.upc.edu

1 Introduction: the problem

Content stealing in the web is becoming a serious concern for information and e-commerce websites. In the practices known as *web fetching* or *web scraping* [1], a stealer bot simulates a human web user to extract desired content off the victim's website. Stolen content is then normally stripped of copyright or authorship information and rendered as belonging to the stealer, on a different site. The incidence of *web scraping* is increasing for several reasons: the simplicity to simulate human navigation, the difficulty to tell bots apart, the grey area on its legal status [3] and, most importantly, the profitability of the business.

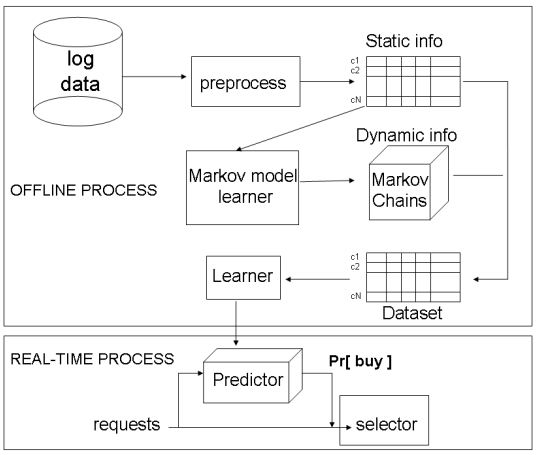
As in the case of spam and domain redirection, web scraping is part of a new breed of Internet abuses. Web scraping is shifting from simple plagiarism to profit-making using web advertising techniques. Non-profit sites such as the Wikipedia have been particularly prone to scrapping for plagiarism [1], moreover, e-commerce sites are increasingly being affected directly by their actions.

As an example, we take the online travel sales industry. Online travel agents contract services from Global Distribution Systems (GDS) [4] under specific SLAs and *look-to-book* ratios (number of searches per reservation). When a user makes a flight search, the request is sent via web services to the GDS, which in most cases forwards petitions to airline companies to produce the final flight availability result. Recently, *flight comparison* sites are appearing that operate by scraping in real-time several travel sites and combining results in their own page. Although they might be useful for users, they are becoming a problem for real travel agencies and the rest of the supply chain, as each search is resource-intensive and costly. *Flight comparison* sites also increase the *look-to-book* ratio, therefore the derived costs of travel agents [4], while operating basically at zero cost. Detection and banning of such sites is generally performed manually by administrators by blocking their IP address; nevertheless, as in the case of spammers, web scrapers are continuously introducing new techniques to bypass detection, such as IP proxies or IP pools [3].

In [2], we have recently introduced the AUGURES architecture, a system to automatically prioritize anonymous web user sessions in e-commerce sites, according to the expected revenue that will generate. As AUGURES prioritizes users according to their expected revenue, and most web scrapers never purchase, we have detected that they were systematically being assigned a very low priority, and thus the firewall could discontinue these sessions in case system resources where scarce. As a proof of concept for this NIPS workshop, we have expanded the mechanism from only detecting purchasing users, to also detect *web bots*, in a travel agency dataset presented in [2] that we use again here.

2 The AUGURES architecture

AUGURES currently has two subsystems: an offline component, that takes the historical web logfile to learn models that make predictions about each class of user future behavior by running machine learning classifiers; and a real-time component, implemented as a service that runs along the session manager of the firewall. The real-time component analyses incoming requests, runs them through a *predictor*, and outputs the priority along with other static information for the session. Further details on AUGURES can be found in [2], Figure 1 presents the general architecture.



(a) Figure 1: AUGURES architecture

	Buyers	Non-Buyers	Bots
%precision	61	91	81
%recall	54	94	74

(b) Table 1: Percentage of recall and precision.

3 Experiments

We wanted to test whether it was possible to identify with reasonable certainty bots accessing a web site for automated banning. For this we have classified each session in the training dataset, as either a *buying* (human) user, a *non-buyer* user, or a *bot*. The first two classes were those considered in [2], while the *bot* class is new to this work. To classify a session as content stealing bot we have used the following criteria: the total number of searches, time average between searches, and the number of origins/destinations in all searches, for each web session or IP addresses. The approach has been manually sampled to validate bot classifications to minimize false positives; once training models are generated further classification can be performed automatically. In this way, about 15% of the traffic in the training set ended up being marked as due to content stealing bots.

The recall and precision obtained for each class by AUGURES are presented in Table 1. In particular, 74% of bot accesses were predicted as such, and among all accesses predicted as bots, 81% truly corresponded to bots. We regard these figures as quite good. In particular, they indicate that the system could potentially be used to filter out at least 10% of total traffic by banning bots.

4 Conclusions

Content stealing on the web is proliferating rapidly as it becomes more profitable, affecting not only content sites, but also e-commerce sites that rely on paid transactions for product availability. We have extended our AUGURES system in [2] to show promising results on the applicability of machine learning techniques for automatic detection and banning of bots. Results obtained can free administrators from manually performing these operations while leveraging revenue loss from the spurious transactions.

References

[1] Hepp, M., D. Bachlechner, and K. Siorpaes. *Harvesting Wiki Consensus - Using Wikipedia Entries as Ontology Elements.* Proceedings of the ESWC2006, Budva, Montenegro, 2006.

[2] N.Poggi, T. Moreno, J. Berral, R. Gavalda, J. Torres. *Web Customer Modeling for Automated Session Prioritization on High Traffic Sites.* Proceedings of the 11th International Conference on User Modeling. Corfu, Greece, June 25-29, 2007.

[3] WebPage: Web Scrapping, `http://en.wikipedia.org/wiki/Web_scraping`, Wikipedia, 2007.

[4] WebPage: Computer Reservations System, `http://en.wikipedia.org/wiki/Computer_reservations_system`, Wikipedia, 2007.