Determining Bias to Search Engines from Robots.txt

Yang Sun, Ziming Zhuang, Isaac G. Councill, and C. Lee Giles
Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16801, USA
{ysun, zzhuang, icouncill, giles}@ist.psu.edu

Abstract

Search engines largely rely on robots (i.e., crawlers or spiders) to collect information from the Web. Such crawling activities can be regulated from the server side by deploying the Robots Exclusion Protocol in a file called robots.txt. Ethical robots will follow the rules specified in robots.txt. Websites can explicitly specify an access preference for each robot by name. Such biases may lead to a "rich get richer" situation, in which a few popular search engines ultimately dominate the Web because they have preferred access to resources that are inaccessible to others. This issue is seldom addressed, although the robots.txt convention has become a de facto standard for robot regulation and search engines have become an indispensable tool for information access. We propose a metric to evaluate the degree of bias to which specific robots are subjected. We have investigated 7,593 websites covering education, government, news, and business domains, and collected 2,925 distinct robots.txt files. Results of content and statistical analysis of the data confirm that the robots of popular search engines and information portals, such as Google, Yahoo, and MSN, are generally favored by most of the websites we have sampled. The results also show a strong correlation between the search engine market share and the bias toward particular search engine robots.

1   Introduction

[1] Without robots, there would probably be no search engines. Web search engines, digital libraries, and many other web applications such as offline browsers, internet marketing software and intelligent searching agents heavily depend on robots to acquire documents. Robots, also called "spiders", "crawlers", "bots" or "harvesters", are self-acting agents that navigate around-the-clock through the hyperlinks of the Web, harvesting topical resources at zero costs of human management [3, 4, 14]. Because of the highly automated nature of the robots, rules must be made to regulate such crawling activities in order to prevent undesired impact to the server workload or access to non-public information.

The Robots Exclusion Protocol has been proposed [12] to provide advisory regulations for robots to follow. A file called robots.txt, which contains robot access policies, is deployed at the root directory of a website and accessible to all robots. Ethical robots read this file and obey the rules during their visit to the website. The robots.txt convention has been adopted by the community since the late 1990s, and has continued to serve as one of the predominant means of robot regulation. However, despite the criticality of the robots.txt convention for both content providers and harvesters, little work has been done to investigate its usage in detail, especially at the scale of the Web.

More importantly, as websites may favor or disfavor certain robots by assigning to them different access policies, this bias can lead to a *"rich get richer"* situation whereby some popular search engines are granted exclusive access to certain resources, which in turn could make them even more popular. Considering the fact that users often prefer a search engine with broad (if not exhaustive) information coverage, this *"rich get richer"* phenomenon may introduce a strong influence on users' choice of search engines, which will eventually be reflected in the search engine market share. On the other hand, since it is often believed (although this is an exaggeration) that *"what is not searchable does not exist,"* this phenomenon may also introduce a biased view of the information on the Web.

1.1   Related Work and Contributions

A 1999 study of the usage of robots.txt[10] in UK universities and colleges investigated 163 websites and 53 robots.txt. Robots.txt files were examined in terms of file size and the use of Robots Exclusion Protocol within the UK university domains. In 2002, Drott [7] studied the usage of robots.txt as an aid for indexing to protect information. 60 samples from Fortune Global 500 company web-

---

[1]A greatly abbreviated version of this paper appeared as a poster in the Proceedings of the 16th International World Wide Web Conference, 2007.

sites were manually examined in this work concluding that *"robots.txt files are not widely used by the sampled group and for most of the sites on which they appear, they are redundant. ...they exclude robots from directories which are locked anyway."* Our investigation shows a contrary result which may be due to the difference in sample size, domain and time. Other work addresses the legal aspects of obeying robots.txt [2, 8] and an overview of Web robots and robots.txt usage is given in [5].

None of the aforementioned work investigates the content of robots.txt in terms of biases towards different robots. In addition, the sample sizes of previous studies have tended to be relatively small considering the size of the Web. In this paper, we present the first quantitative study of such biases, and conduct a more comprehensive survey of robots.txt usage on the Web. By implementing our own specialized "robots.txt" crawler, we collect real-world data from a considerable amount of unique websites with different functionalities, covering the domains of education, government, news, and business. We investigate the following questions:

- Does a robot bias exist?

- How should such a bias be measured quantitatively?

- What is the implication for such a bias?

Our contributions are:

- We propose a quantitative metric to automatically measure robot biases.

- By applying the metric to a large sample of websites, we present our findings about the most favored and disfavored robots.

The rest of the paper is organized as follows. In Section 2 we briefly introduce the Robots Exclusion Protocol. In Section 3, We propose a bias metric and demonstrate how it is applied to measure the degree of robot bias. In Section 4 we present our data collection for this study. In Section 5 we present our observations on robots.txt usage and discuss the implications. In Section 6 we conclude our paper with plans for future work.

2 Robots Exclusion Protocol

The Robots Exclusion Protocol[2] is a convention that allows website administrators to indicate to visiting robots which parts of their site should not be visited. If there is no robots.txt file on a website, robots are free to crawl all content.

The format of Robots Exclusion Protocol is described in [12]. A file named *"robots.txt"* with Internet Media

Type "text/plain" is placed under the root directory of a Web server. Each line in the robots.txt file has the format: $< field >:< optionalspace >< value >< optionalspace >$. There are three types of case-insensitive tags for the $< field >$ to specify the rules: *User-Agent*, *Allow* and *Disallow*. Another unofficial directive *Crawl-Delay* is also used by many websites to limit the frequency of robot visits.

The *robots.txt* file starts with one or more $User - Agent$ fields, specifying which robots the rules apply to, followed by a number of $Disallow :$ and/or $Allow :$ fields indicating the actual rules to regulate the robot. Comments are allowed anywhere in the file, and consist of optional whitespaces. Comments are started with a comment character '#' and terminated by the linkbreak.

A sample *robots.txt* is listed below (this robots.txt file is from *BotSeer*[3]):

```
User-Agent: *
Disallow: /robots/
Disallow: /src/
Disallow: /botseer
Disallow: /uastring
Disallow: /srcseer
Disallow: /robotstxtanalysis
Disallow: /whois

User-Agent: googlebot
Disallow: /robotstxtanalysis
Disallow: /uastring

User-Agent: botseer
Disallow:
```

It shows that *Googlebot* cannot visit *"/robotstxtanalysis"* and *"/uastring"*. *BotSeer* can visit any directory and file on the server. All the other robots should follow the rules under $User - Agent : *$ and cannot visit the directories and files matching *"/robots/"*, *"/src/"*, *"/botseer"*, *"/uastring"*, *"/srcseer"*, *"/robotstxtanalysis"*, *"/whois"*.

3 Robot Bias

We propose $\Delta P(r)$, a measure of the favoribility of robots across a sample of robots.txt files, to measure the degree to which specific robots are favored (or disfavored) by a set of websites. A formal definition of the robot bias (favored or disfavored) is described below.

3.1 The *GetBias* Algorithm

Our definition of a favored robot is a robot allowed to access more directories than the universal robot according to

the robots.txt file in the website. The universal robot is any robot that has not matched any of the specific User-Agent names in the robots.txt file. In other words, the universal robot represents all the robots that do not appear by name in the robots.txt file.

Let F be the set of robots.txt files in our dataset. Given a robots.txt file $f \in F$, let R denote the set of named robots for a given robots.txt file f. For each named robot $r \in R$, We define the $GetBias(r, f)$ algorithm as specified in *Algorithm 1*. *GetBias* measures the degree to which a named robot r is favored or disfavored in a given robots.txt file f.

Algorithm 1 $GetBias(r, f)$

1: **if** r is $*$ **then**
2:     return 0
3: **end if**
4: Construct *DIR* for f;
5: $bias = 0$
6: **for all** $d \in DIR$ **do**
7:     **if** d is allowed for $*$ **then**
8:         $D_u \leftarrow d$
9:     **end if**
10: **end for**
11: **for all** $d \in DIR$ **do**
12:     **if** d is allowed for r **then**
13:         $D_r \leftarrow d$
14:     **end if**
15: **end for**
16: $bias = |D_r| - |D_u|$
17: return $bias$

Let *DIR* be the set of all directories that appear in a robots.txt file f of a specific website. *DIR* is used as an estimation of the actual directory structure in the website because the Robot Exclusion Protocol considers any directory in the website that does not match the directories in the robots.txt as an allowed directory by default. $D_u \in DIR$ is the set of directories that the universal robot "*" is allowed to visit. If there are no rules specified for $User - Agent : *$, the universal robot can access everything by default. $D_r \in DIR$ is the set of directories that a given robot r is allowed to visit. $|D_u|$ and $|D_r|$ are the number of directories in D_u and D_r.

For a given robot r, the algorithm first counts how many directories in *DIR* is allowed for r. Then it calculates the bias score for robot r as the difference between the number of directories in *DIR* that are allowed for the robot r and the number of directories that are allowed for the universal robot. In the $GetBias$ algorithm, the bias of the universal robot is treated as the reference point 0 ($GetBias$ returns 0). The bias scores of favored robots returned by $GetBias$ are positive values. Higher score of a robot means the robot is more favored. On the contrary, the bias scores of disfavored

robots returned by $GetBias$ are negative values, which is consistent with our bias definition. Thus, the bias of a robot in a robots.txt file can be represented by a categorical variable with three categories: *favored*, *disfavored*, and *no bias*.

As an example, consider the robots.txt file in *http://BotSeer.ist.psu.edu* from Section 2: $DIR = \{$"/robots/", "/src/", "/botseer", "/uastring", "/srcseer", "/robotstxtanalysis", "/whois"$\}$. According to the algorithm we have $D_u = \{$null$\}$, $D_{botseer} = \{$"/robots/", "/src/", "/botseer", "/uastring", "/srcseer", "/robotstxtanalysis", "/whois"$\}$ and $D_{google} = \{$"/robots/", "/src/", "/botseer", "/srcseer", "/whois"$\}$. Thus, $|D_u| = 0$, $|D_{botseer}|=7$, and $|D_{google}|=5$. According to Algorithm 1, $bias_u = |D_u| - |D_u| = 0$, $bias_{botseer} = |D_{botseer}| - |D_u| = 7$ and $bias_{googlebot} = |D_{google}| - |D_u| = 5$. Thus, the robots "googlebot" and "botseer" are favored by this website, and they are categorized as *favored*. All other robots will be categorized as *no bias*.

3.2 Measuring Overall Bias

Based on the bias score for each file, we propose $\Delta P(r)$ favorability in order to evaluate the degree to which a specific robot is favored or disfavored on a set of robots.txt files. Let $N = |F|$ be the total number of robots.txt files in the dataset. The $\Delta P(r)$ favorability of a robot r can be defined as below:

$$
\begin{aligned}
\Delta P(r) &= P_{favor}(r) - P_{disfavor}(r) \\
&= \frac{N_{favor}(r) - N_{disfavor}(r)}{N}.
\end{aligned}
\tag{1}
$$

where $N_{favor}(r)$ and $N_{disfavor}(r)$ are the number of times a robot is favored and disfavored respectively. $P_{favor}(r)$ is the proportion of the robots.txt files in which a robot r is favored; $P_{disfavor}(r)$ is the proportion of the robots.txt files in which a robot r is disfavored.

The proportions of robots.txt files that favor or disfavor a specific robot are simple measures for survey statistics; however, in our dataset the two proportions in isolation are not very accurate in reflecting the overall biases in our sample since there are more than two events (favor, disfavor and no bias). This means that $P_{favor}(r) + P_{disfavor}(r) < 1$. Each event only reflects one aspect of the bias. For example, a robot named "ia_archiver" is favored by 0.24% of the websites in our dataset and the proportion of sites that favor "momspider" is 0.21%. Alternatively, the proportions of sites that disfavor "ia_archiver" and "momspider" are 1.9% and 0%, respectively. If we only consider the favored proportion, we will reach the conclusion that "ia_archiver" is more favored than "momspider".

$\Delta P(r)$ is the difference of the proportions of sites that favor and disfavor a specific robot, and thus treats both cases

in unison. For the above example $\Delta P(ia\_archiver)$ is -1.66% and $\Delta P(momspider)$ is 0.21%. Thus, "momspider" is more favored than "ia_archiver". For any no-bias robot r, $\Delta P(r)$ is 0. The bias measure can eliminate the misleading cases and still be intuitively understandable (favored robots have positive numbers and disfavored robots have negative numbers).

3.3 Examining Favorability

The favorability is actually a ranking function of robots. To evaluate accuracy of this ranking function, we run a ranking performance test based on Kendall's rank correlation method [11]. The rank correlation method is briefly described below. The details of the ranking performance evaluation using partial order can be found in [9].

For a robots.txt file f, let m_a be a bias measure function for all robots R appearing in f. Let r_i and $r_j \in R$ be two named robots in f. We denote $r_i <_{m_a} r_j$ if r_i is ranked higher than r_j for measure m_a. Thus, for any two measure functions m_a and m_b, Kendall's τ can be defined based on the number P_f of concordant pairs and the number Q_f of discordant pairs. A pair $r_i \neq r_j$ is concordant if both m_a and m_b agree in how they order r_i and r_j. It is discordant if they disagree. In this case, Kendall's τ can be defined as:

$$\tau_f(m_a, m_b) = \frac{P_f - Q_f}{P_f + Q_f} \qquad (2)$$

For any given measure m_a and m_b, the $\tau_f(m_a, m_b)$ represents how well the two ranking measures agree with each other in a file f. Let m_a represent the actual ranking function of robots. Although we do not know the actual ranking function, we have the partial ranking of robots for each robots.txt file based on the bias score defined previously. Thus, computing the $\tau_f(m_a, m_b)$ for all robots.txt files will show how well the measure m_b agrees with m_a for the actual ranking of robots in file f.

We calculate $\tau_f(m_a, m_b)$ for each robots.txt files f in our dataset. If $\tau_f(m_a, m_b) = 1$ for a given robots.txt file, we consider that the file f is a concordant file for m_a and m_b. Otherwise, the file f is a discordant file. By counting the concordant files P and discordant files Q in the dataset, we can compute the average $\overline{\tau}(m_a, m_b)$. Note that $P + Q = N$, thus,

$$\overline{\tau}(m_a, m_b) = \frac{P - Q}{P + Q} = 1 - \frac{2Q}{N}. \qquad (3)$$

We rank the robots using the δP favorability. The ranked lists are then compared with the actual ranking using the method introduced above. The average $\overline{\tau}$ value is 0.957 which we believe is accurate enough to measure the overall bias of a robot.

4 Data Collection

To observe the potential robot bias on the Web, our work studies a wide range of websites with different domains and from different physical locations. The data collection is described below in detail.

4.1 Data Sources

The Open Directory Project [6] is the largest and most comprehensive human-maintained Web directory. Our primary source to collect the initial URLs to feed our crawler is DMOZ because the Open Directory Project classifies a large URL collection into different categories. It enables us to collect data from different domains and physical locations. Our collection from the Open Directory Project covers three domains: education, news, and government. The university domain is further broken down into the American, European, and Asian university domains.

Since the directory structure of the business domain in DMOZ is complicated and has a significant amount of overlaps, we use the 2005 Fortune Top 1000 Company List [1] as our data source.

There are certain limitations inherent in our data collection. First, because the website collection in DMOZ is limited for other countries especially for non-English websites, the majority of the websites are from the USA. Second, because the DMOZ entries are organized by human editors, there might be errors. Finally, we collect business websites from the Fortune 1000 list which contains data mostly of large corporations, so that the data in that domain may not be representative of small businesses. We intend to address these limitations in future research.

4.2 Crawling for Robots.txt

We have implemented a specialized focused crawler for this study. The crawler starts by crawling the metadata of a website obtained from DMOZ including the functional classification, the name of the website, and the physical location of its affiliated organization. Then, the crawler checks the existence of robots.txt for that domain and downloads existing robots.txt files for offline analysis. A parsing and filtering module is also integrated into our crawler to eliminate duplicates and for ensuring that retrieved pages are within the target domain.

Besides the root level directory, our crawler also examines other possible locations of the robots.txt file. The subdirectories of a website (up to level 3) are inspected. Results show that there are few cases where robots.txt is not placed under the root directory where it should be according to the Robots Exclusion Protocol. Misspelled filenames are also examined by our crawler. In rare cases the filename "robot.txt" (which will be ignored by robots) is used instead of "robots.txt".

In order to observe the temporal properties, the crawler has performed 5 crawls for the same set of websites from Dec. 2005 to Oct. 2006. In order to analyze the temporal properties, the downloaded robots.txt files are archived according to the date of the crawl.

4.3 Statistics

We crawled and investigated 7,593 unique websites including 600 government websites, 2047 newspaper websites, 1487 USA university websites, 1420 European university websites, 1039 Asian university websites, and 1000 company websites. The number of websites that have robots.txt files in each domain from the 5 crawls are shown in Table 1.
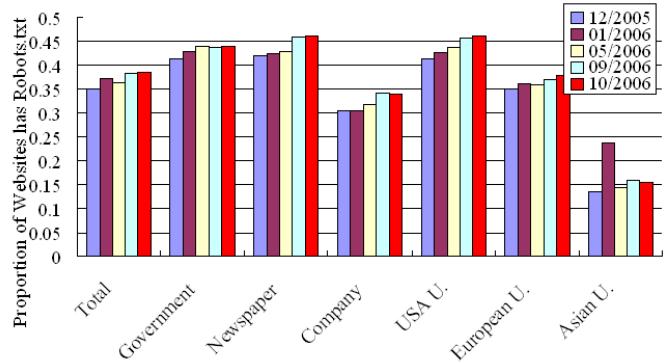
| | Websites | Collected robots.txt files | | | | |
		Dec. 2005	Jan. 2006	May. 2006	Sep. 2006	Oct. 2006
Government	600	248	257	263	262	264
Newspaper	2047	859	868	876	937	942
USA Univ.	1487	615	634	650	678	683
European Univ.	1420	497	510	508	524	537
Company	1000	303	306	319	341	339
Asian Univ.	1039	140	248	149	165	160
Total	7593	2662	2823	2765	2907	2925

Table 1. Number of robots.txt found in each domain for each crawl.

To better describe the usage of the robots.txt in websites in different domains, Figure 1 illustrates the proportion of websites having robots.txt in each domain. Overall, except for in the case of Asian university websites, the usage of robots.txt has increased. 46.02% of newspaper websites currently have implemented robots.txt files and the newspaper domain is the domain in which the Robots Exclusion Protocol is most frequently adopted. 45.93% of the USA university websites in our sample adopt the Robots Exclusion Protocol, significantly more than European (37.8%) and Asian (15.4%) sites. Since search engines and intelligent searching agents become more important for accessing web information, this result is expected. The Robots Exclusion Protocol is more frequently adopted by government, newspaper and university websites in the USA. It is used extensively to protect information not to be offered to the public and balance workload for these websites. A detailed robots.txt usage report can be found in [16].

5 Results

There are 1056 named robots found in our dataset. The universal robot "*" is the most frequently used robot in the

Figure 1. Probability of a website that has robots.txt in each domain.

User-Agent field and used 2744 times, which means 93.8% of robots.txt files have rules for the universal robots. 72.4% of the named robots appeared only once or twice. The most frequently appearing robots in our dataset are shown in Figure 2.
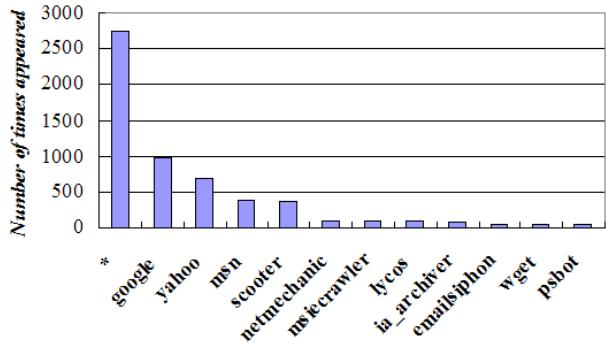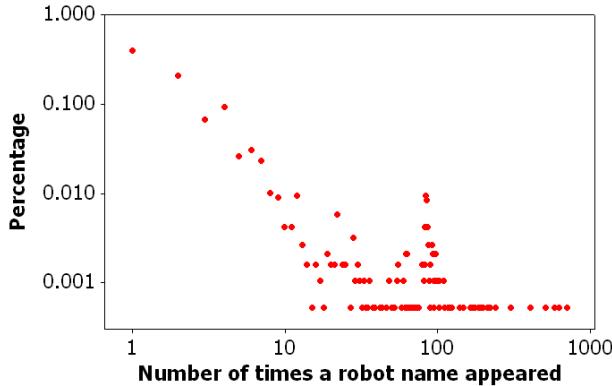

Figure 2. Most frequently used robot names in robots.txt files. The height of the bar represents the number of times a robot appeared in our dataset.

5.1 History of Bias

The distribution of how many times a robot is used (see Figure 3) did not change significantly over the past 11 months. Thus, we show the bias results from the latest crawl since not much has changed.

Since most of the robots appeared only once or twice in the dataset, their ranking scores are ranked in the middle of the list and are almost indistinguishable. We consider only the top ranked (favored) and bottom ranked (disfavored) robots. The 10 most favored robots and 10 most

Figure 3. The distribution of a robot being used.

Favored Robots (Sample size $N = 2925$)				
robot name	N_{favor}	N_{disf}	$\Delta P(r)$	σ
google	877	25	0.2913	0.0084
yahoo	631	34	0.2041	0.0075
msn	349	9	0.1162	0.0059
scooter	341	15	0.1104	0.0058
lycos	91	5	0.0294	0.0031
netmechanic	84	10	0.0253	0.0029
htdig	15	3	0.0041	0.0012
teoma	13	3	0.0034	0.0011
oodlebot*	8	0	0.0027	0.0010
momspider	6	0	0.0021	0.0008
Disfavored Robots (Sample size $N = 2925$)				
robot name	N_{favor}	N_{disf}	$\Delta P(r)$	σ
msiecrawler	0	85	-0.0291	0.0031
ia_archiver	7	55	-0.0164	0.0023
cherrypicker	0	37	-0.0126	0.0021
emailsiphon	3	34	-0.0106	0.0019
roverbot	2	27	-0.0085	0.0017
psbot	0	23	-0.0079	0.0016
webzip	0	21	-0.0072	0.0016
wget	1	22	-0.0072	0.0016
linkwalker	2	20	-0.0062	0.0015
asterias	0	18	-0.0062	0.0015

Table 2. Top 10 favored and disfavored robots. σ is the standard deviation of $\Delta P(r)$.

disfavored robots are listed in Table 2 where N is the sample size, N_{favor} is the number of times the robot is favored, $N_{disfavor}$ is the number of times the robot is disfavored and $\sigma = \sqrt{\frac{\Delta P(r)(1-\Delta P(r))}{N}}$ is the categorical standard deviation[13] of $\Delta P(r)$. The categorical standard deviation σ gives the variance when using ΔP to estimate the favorability of robots on the Web.

Our bias measure shows that the most highly favored robots are from well-known search engines and organizations, e.g., "Google", "Yahoo" and "MSN" are favored much more than the remaining robots. Please note that for some robots in the *disfavored* category, their ΔP favorability does not show a significant difference due to their rare appearances in the sampled robots.txt files.

On the other hand, most of the disfavored robots are email collectors ("CherryPicker" and "emailsiphon") and off-line browsers ("Wget" and "webzip"). From the privacy perspective, it is reasonable for webmasters to exclude robots whose major purpose is to collect private information. Also, webmasters typically do not want their websites to be copied entirely by others. However, even robots from well-known companies can be disfavored e.g., "MSIECrawler" (Microsoft) and "ia_archiver" (Alexa). "MSIECrawler" is a robot embedded in Internet Explorer (IE). When IE users bookmark a page while offline, MSIECrawler downloads the page and all links related to it, including links, images, JavaScript and Style sheets, when the user is next online. "ia_archiver" is the crawler from archive.org and Alexa.com. A list of detailed description of known robots appeared in this paper can be found on the web[4].

We also find that robot biases in different domains vary

significantly. Google is always the most favored robot. Other top favored robots vary in different domains. Yahoo ("slurp" is a Yahoo robot) and MSN are also favored in most domains, but they are not significantly favored over other robots. Other top favored robots are mostly open source crawlers and crawlers from well-known organizations. Disfavored robot lists vary widely for different domains. Most of these robots are still email collectors and offline browsers. The differences could be due to the different behaviors of robots in different domains (e.g., emailsiphon may crawl business websites more often than others to collect business contacts).

5.2 Search Engine Market vs. Robot Bias

In order to study the impact of the "rich get richer" effect, we calculate the correlation between the robot bias and the search engine market share for specific companies. The market share of Google, Yahoo, MSN and Ask in the past 11 months and the ΔP favorability for the corresponding robots are considered two independent variables. The Pearson product-moment correlation coefficient[15] (PMCC)

between the two variables is a measure of the tendency of two variables X and Y measured on the same object or organism to increase or decrease together. For our dataset, the Pearson correlation of the market share of the four companies and the $\Delta P(r)$ of their corresponding robots is 0.930 with P-Value < 0.001. The search engine market share [5] and robot bias in September, 2006 is shown in Figure 4.
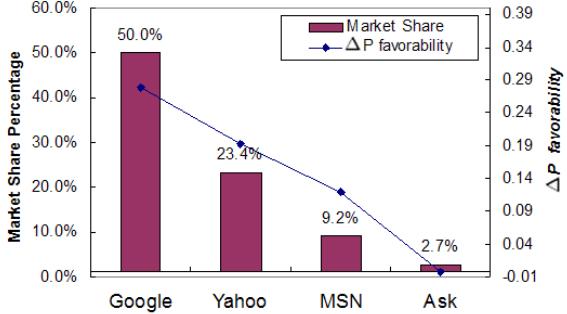
Figure 4. Search engine market share vs. robot bias.

6 Conclusions

We have presented a comprehensive survey of robot biases on the Web through careful content and statistical analysis of a large sample of robots.txt files. Results show that the robots of popular search engines and information portals, such as Google, Yahoo, and MSN, are generally favored by most of the websites we have sampled. This implies a "rich get richer" bias toward popular search engines. We also shows a strong correlation between the search engine market share and the bias toward corresponding robots. Our study indicates that the usage of robots.txt has increased over the past 11 months in which 2,662 robots.txt files were found in the first crawl and 2,925 files were found for the last crawl. We observe that 46.02% of newspaper websites currently have implemented robots.txt files and the newspaper domain is the domain in which the Robots Exclusion Protocol is most frequently adopted. 45.93% of the USA university websites in our sample adopt the Robots Exclusion Protocol, significantly more than European (37.8%) and Asian (15.4%) sites. Our future work will try to further break down the analysis by geographical region to investigate the robot favorability in each country.

Future work will pursue a deeper and larger scale analysis of robots' behavior and regulations. We are investigating other metrics for robot bias. Experimental investigation of web robots' behavior will be undertaken in order to better understand how live robots interpret the Robots Exclusion Protocol.

References

[1] Fortune magazine. http://money.cnn.com/magazines/fortune/-fortune500, 2005.

[2] M. L. Boonk, D. R. A. d. Groot, F. M. T. Brazier, and A. Oskamp. Agent exclusion on websites. In *Proceedings of The 4th Workshop on the Law and Electronic Agents*, 2005.

[3] S. Chakrabarti, M. Van den Berg, , and B. Dom. Focused crawling: A new approach to topic-specific web resource discovery. In *Proc. of the 8th WWW Conference*, pages 545–562, 1999.

[4] J. Cho, H. Garcia-Molina, and L. Page. Efficient crawling through url ordering. In *Proceedings of the 7th International World Wide Web Conference*, 1998.

[5] T. Y. Chun. World wide web robots: An overview. *Online Information Review*, 23(3):135–142, 1999.

[6] DMOZ. The open directory project. http://dmoz.org, 2005.

[7] M. Drott. Indexing aids at corporate websites: The use of robots.txt and meta tags. *Information Processing and Management*, 38(2):209–219, 2002.

[8] D. Eichmann. Ethical web agents. *Computer Networks and ISDN Systems*, 28(1-2):127–136, 1995.

[9] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, New York, NY, USA, 2002. ACM Press.

[10] B. Kelly and I. Peacock. Webwatching uk web communities: Final report for the webwatch project. *British Library Research and Innovation Report*, 1999.

[11] M. Kendall. *Rank Correlation Methods*. Hafner, 1955.

[12] M. Koster. A method for web robots control. In *the Internet Draft, The Internet Engineering Task Force (IETF)*, 1996.

[13] R. L. Ott and M. T. Longnecker. *An Introduction to Statistical Methods and Data Analysis*. Duxbury Press; 5th edition, 2000.

[14] G. Pant, P. Srinivasan, and F. Menczer. *Crawling the Web*, chapter Web Dynamics. Springer-Verlag, 2004.

[15] R. R. Sokal and F. J. Rohlf. *Biometry*. Freeman New York, 2001.

[16] Y. Sun, Z. Zhuang, and C. L. Giles. A large-scale study of robots.txt. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1123–1124, New York, NY, USA, 2007. ACM Press.

---

[5]http://www.netratings.com