Can Social Bookmarking Improve Web Search?

Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina
Dept. of Computer Science, Stanford University
Stanford, CA, USA
heymann@stanford.edu, koutrika@stanford.edu, hector@cs.stanford.edu

ABSTRACT

Social bookmarking is a recent phenomenon which has the potential to give us a great deal of data about pages on the web. One major question is whether that data can be used to augment systems like web search. To answer this question, over the past year we have gathered what we believe to be the largest dataset from a social bookmarking site yet analyzed by academic researchers. Our dataset represents about forty million bookmarks from the social bookmarking site del.icio.us. We contribute a characterization of posts to del.icio.us: how many bookmarks exist (about 115 million), how fast is it growing, and how active are the URLs being posted about (quite active). We also contribute a characterization of tags used by bookmarkers. We found that certain tags tend to gravitate towards certain domains, and vice versa. We also found that tags occur in over 50 percent of the pages that they annotate, and in only 20 percent of cases do they not occur in the page text, backlink page text, or forward link page text of the pages they annotate. We conclude that social bookmarking can provide search data not currently provided by other sources, though it may currently lack the size and distribution of tags necessary to make a significant impact.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.5 [**Information Storage and Retrieval**]: On-line Information Services; H.1.2 [**Models and Principles**]: User/Machine Systems—*Human information processing*

General Terms

Design, Experimentation, Human Factors, Measurement

Keywords

Social bookmarking, collaborative tagging, web search

WSDM'08, February 11–12, 2008, Palo Alto, California, USA.
Copyright 2008 ACM 978-1-59593-927-9/08/0002 ...$5.00.

1. INTRODUCTION

For most of the history of the web, search engines have only had access to three major types of data describing pages. These types are page content, link structure, and query or clickthrough log data. Today a fourth type of data is becoming available: user generated content (e.g., tags, bookmarks) describing the pages directly. Unlike the three previous types of data, this new source of information is neither well studied nor well understood. Our aim in this paper is to quantify the size of this data source, characterize what information it contains, and to determine the potential impact it may have on improving web search.

This paper centers around a series of experiments conducted on the social bookmarking site del.icio.us. In Sections 2, 3, and 4, we give the background terminology, methodology and related work for our experiments on del.icio.us. The core of our paper, Sections 5 and 6, gives two sets of results. Section 5 contains results that suggest that social bookmarking will be useful for web search, while Section 6 contains those results that suggest it will not. Both sections are divided into "URL" and "tag" subsections which focus on the two major types of data that social bookmarking provides. Finally, in Section 7 we conclude with our thoughts on the overall picture of social bookmarking and its ability to augment web search.

2. TERMINOLOGY

We differentiate *social bookmarking* from other social sites involving shared bookmarks, like *social news* sites. The two major social bookmarking sites are del.icio.us and StumbleUpon, while the three major social news sites are digg.com, reddit.com, and netscape.com.

We consider three units of data from social bookmarking sites:

Triple A triple is a $< user_i, tag_j, url_k >$ tuple, signifying that user i has tagged URL k with tag j.

Post A post is a URL bookmarked by a user and all associated metadata. A post is made up of many triples, though it may also contain information like a user comment.

Label A label is a $< tag_i, url_k >$ pair that signifies that at least one triple containing tag i and URL k exists in the system.

We use *term* to describe a unit of text, whether it is a tag or part of a query. Terms are usually words, but are also sometimes acronyms, numbers, or other tokens.

We use *host* to mean the full host part of a URL, and *domain* to mean the "effective" institutional level part of the host. For instance, in `http://i.stanford.edu/index.html`, we call `i.stanford.edu` the host, and `stanford.edu` the domain. Likewise, in `http://www.cl.cam.ac.uk/`, we call `www.cl.cam.ac.uk` the host, and `cam.ac.uk` the domain. We use the effective top level domain (TLD) list from the Mozilla Foundation to determine the effective "domain" of a particular host.[1]

3. METHODOLOGY

We chose to focus on the main social bookmarking site: del.icio.us. The companies that control social sites often run a number of internal analyses, but are usually reluctant to release specific results. This can be for competitive reasons, or perhaps simply to ensure the privacy of their users. As a result, we worked independently and through public interfaces to gather the data for this study. Doing so presented a number of challenges.

3.1 Interfaces

del.icio.us offers a variety of interfaces to interested parties, but each of these has its own caveats and potential problems. For instance, the "recent" feed provides the most recent bookmarks posted to del.icio.us in real time. However, while we found that the majority of public posts by users were present in the feed, some posts were missing (due to filtering, see Section 3.4). Interfaces also exist which show all posts of a given URL, all posts by a given user, and the most recent posts with a given tag. We believe that at least the posts-by-a-given-user interface is unfiltered, because users often share this interface with other users to give them an idea of their current bookmarks.

These interfaces allow for two different strategies in gathering datasets from del.icio.us. One can *monitor the recent feed*. The advantage of this is that the recent feed is in real time. This strategy also does not provide a mechanism for gathering older posts. Alternatively, one can *crawl* del.icio.us, treating it as a tripartite graph. One starts with some set of seeds—tags, URLs, or users. At each tag, all URLs tagged with that tag and all users who had used the tag are added to the queue. At each URL, all tags which had been annotated to the URL (e.g., all labels) and all users who had posted the URL are added to the queue. At each user, all URLs posted or tags used by the user are added to the queue. The advantage of this strategy is that it provides a relatively unfiltered view of the data. However, the disadvantage is that doing a partial crawl of a small world graph like del.icio.us can lead to data which is highly biased towards popular tags, users, and URLs. Luckily, these two methods complement each other. Monitoring is biased against popular pages, while crawling tends to be biased toward these pages (we further explore the sources of these biases in Section 3.4). As a result, we created datasets based on both strategies.

3.2 Realtime Processing Pipeline

For certain analyses (see Result 10), we need to have not just the URL being bookmarked, but also the content of the page, as well as the forward links from the page. We also wanted to have the backlinks from those pages, and the pagetext content of those backlinks. We wanted to have this page text data as soon as possible after a URL was posted.

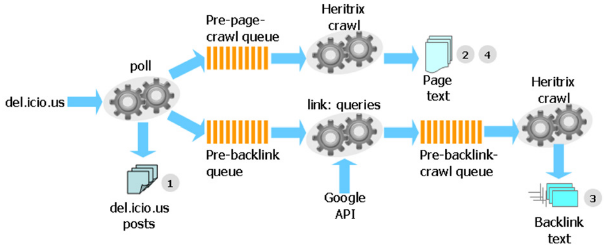


Figure 1: Realtime Processing Pipeline: (1) shows where the post metadata is acquired, (2) and (4) show where the page text and forward link page text is acquired, and (3) shows where the backlink page text is acquired.

As a result, for a one month period we set up a real time processing pipeline (shown in Figure 1). Every 20 to 40 seconds, we polled del.icio.us to see the most recently added posts. For each post, we added the URL of the post to two queues, a *pre-page-crawl queue* and a *pre-backlink queue*.

Every two hours, we ran an 80 minute Heritrix web crawl seeded with the pages in the *pre-page-crawl queue*.[2] We crawled the seeds themselves, plus pages linked from those seeds up until the 80 minute time limit elapsed.[3]

Meanwhile, we had a set of processes which periodically checked the *pre-backlink queue*. These processes got URLs from the queue and then ran between one and three `link:` queries against one of Google's internal APIs. This resulted in 0-60 backlink URLs which we then added to a *pre-backlink-crawl queue*. Finally, once every two hours, we ran a 30 minute Heritrix crawl which crawled only the pages in the *pre-backlink-crawl queue*. In terms of scale, our pipeline produced around 2GB of (compressed) data per hour in terms of crawled pages and crawled backlinks.

3.3 Datasets

Over the course of nine months starting in September 2006 and ending in July 2007, we collected three datasets from del.icio.us:

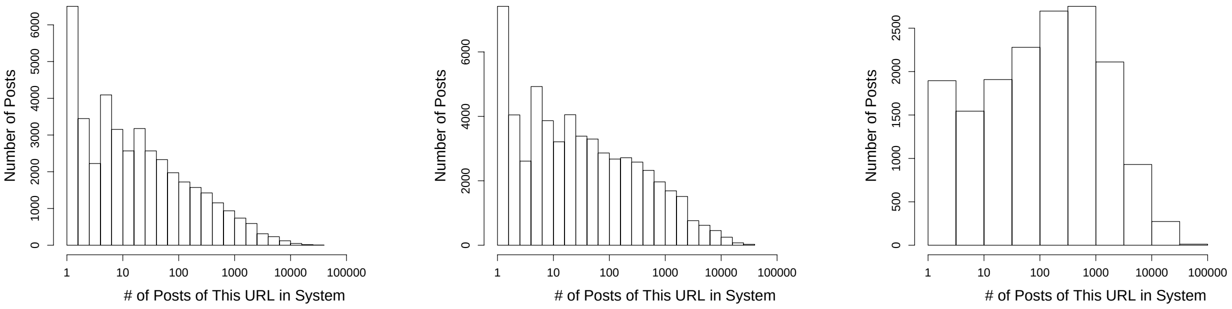
Dataset C(rawl) This dataset consists of a large scale crawl of del.icio.us in September 2006. The crawl was breadth first from the tag "web", with the crawling performed as described above. This dataset consists of 22,588,354 posts and 1,371,941 unique URLs.

Dataset R(ecent) This dataset consists of approximately 8 months of data beginning September 28th, 2006. The data was gathered from the del.icio.us recent feed. This dataset consists of 11,613,913 posts and 3,004,998 unique URLs.

[1] Available `http://publicsuffix.org/`.

[2] Heritrix software available at `http://crawler.archive.org/`.

[3] The reason for running 80 minutes every two hours is that we used a single machine for crawling. The single machine would spend 80 minutes crawling forward links, 30 minutes crawling backlinks, and we left two five minute buffers between the crawls, leading to 120 minutes.



(a) Found URLs (b) Combined (c) Missing URLs

Figure 2: Number of times URLs had been posted and whether they appeared in the recent feed or not.

Dataset M(onth) This dataset consists of one contiguous month of data starting May 25th 2007. This data was gathered from the del.icio.us recent feed. For each URL posted to the recent feed, Dataset M also contains a crawl of that URL within 2 hours of its posting, pages linked from that URL, and inlinks to the URL. This page content was acquired in the manner described in Section 3.2. Unlike Dataset R, the gathering process was enhanced so that changes in the feed were detected more quickly. As a result, we believe that Dataset M has within 1% of all of the posts that were present in the recent feed during the month long period. This dataset consists of $3,630,250$ posts, $2,549,282$ unique URLs, $301,499$ active unique usernames and about 2 TB of crawled data.

We are unaware of any analysis of del.icio.us of a similar scale either in terms of duration, size, or depth.

We also use the AOL query dataset [13] for certain analyses (Results 1, 3, and 6). The AOL query dataset consists of about 20 million search queries corresponding to about $650,000$ users. We use this dataset to represent the distribution of queries a search engine might receive.

3.4 Tradeoffs

As we will see, del.icio.us data is large and grows rapidly. The web pages del.icio.us refers to are also changing and evolving. Thus, any "snapshot" will be imprecise in one way or another. For instance, a URL in del.icio.us may refer to a deleted page, or a forward link may point to a deleted page. Some postings, users, or tags may be missing due to filtering or the crawl process. Lastly, the data may be biased, e.g., unpopular URLs or popular tags may be over-represented.

Datasets C, R, and M each have bias due to the ways in which they were gathered. Dataset C appears to be heavily biased towards popular tags, popular users, and popular URLs due to its crawling methodology. Dataset R may be missing data due to incomplete gathering of data from the recent feed. Datasets R and M are both missing data due to filtering of the recent feed. In this paper, we analyze Dataset M because we believe it is the most complete and unbiased. We use Datasets C and R to supplement Dataset M for certain analyses.

It was important for the analyses that follow not just to know that the recent feed (and thus Datasets R and M) was filtered, but also to have a rough idea of exactly how it was filtered. We analyzed over $2,000$ randomly sampled users, and came to two conclusions. First, on average, about 20% of public posts fail to appear in the recent feed (as opposed to the posts-by-user interface, for example). Second, popular URLs, URLs from popular domains (e.g., `youtube.com`), posts using automated methods (e.g., programmatic APIs), and spam will often not appear in the recent feed. Figure 2 shows this second conclusion for popular URLs. It shows three histograms of URL popularity for URLs which appeared in the recent feed ("found"), those that did not ("missing"), and the combination of the two (i.e.., the "real" distribution, "combined"). Missing posts on the whole refer to noticeably more popular URLs, but the effect of their absence seems minimal. In other words, the "combined" distribution is not substantially different from the "found" distribution.

4. RELATED WORK

Since the beginning of the web, people have used page content to aid in navigation and searching. However, almost as early—Eiron and McCurley [6] suggest as early as 1994—users were suggesting the use of anchortext and link structure to improve web search. Craswell et al. [4] also give some early justification for use of anchortext to augment web search.

Meanwhile, there has also been a current of users attempting to annotate their own pages with metadata. This began with the `<meta>` tag which allowed for keywords on a web page to aid search engines. However, due to search engine spam, this practice has lost favor. The most recent instance of this idea is Google Co-op,[4] where Google encourages site owners to label their sites with "topics." Co-op allows Google to refine search results based on this additional information. However, unlike social bookmarking, these metadata approaches require site owners to know all of the labels a user might attach to their site. This leads to the well studied "vocabulary problem" (see [8], [3]), whereby users have many different types of terminology for the same resources. Ultimately, unlike previous metadata, social bookmarking systems have the potential to overcome the vocabulary problem by presenting many terms for the same content created by many disparate users.

Independently of web search, there has been a growth of interest in tagging. This is primarily due to its usefulness

[4] See `http://www.google.com/coop/`.

as a lightweight organizational tool and as a way to increase text for video and image search. Golder and Huberman [9] were two of the earliest researchers to look at the dynamics of tagging, but many others soon followed ([12, 10, 14]). While a number of papers have looked at del.icio.us, only a few have looked at its relationship to web search. Both Bao et al. [1] and Yanbe et al. [16] propose methods to modify web search to include tagging data. However, neither looked at whether del.icio.us (or any other social bookmarking site) was producing data of a sufficient quantity, quality or variety to support their methods. Both also use relatively small datasets—Bao et al. use 1, 736, 268 web pages and 269, 566 annotations, while Yanbe et al. use several thousand unique URLs. Also, both of these papers are primarily interested in the popularity and tags of the URLs studied, rather than other possible uses of the data.

The ultimate test of whether social bookmarking can aid web search would be to implement systems like those of Bao et al. or Yanbe et al. and see if they improve search results at a major search engine. However, such a test would be expensive, time consuming, and might not really get to the root of why (or why not) social bookmarks help. Our paper aims to provide these insights.

5. POSITIVE FACTORS

Bookmarks are useful in two major ways. First, they can allow an individual to remember URLs visited. For example, if a user tags a page with their mother's name, this tag might be useful to them, but is unlikely to be useful to others. Second, tags can be made by the community to guide users to valuable content. For example, the tag "katrina" might be valuable before search engine indices update with Hurricane Katrina web sites. Non-obvious tags like "analgesic" on a page about painkillers might also help users who know content by different names locate content of interest.

In this paper, our focus is on the second use. Will bookmarks and tags really be useful in the ways described above? How often do we find "non-obvious" tags? Is del.icio.us really more up-to-date than a search engine? What coverage does del.icio.us have of the web? Sections 5 and 6 try to answer questions like these. At the beginning of each result in these sections, we highlight the main result in "capsule form" and we summarize the high level conclusion we think can be reached. In this section, we provide positive factors which suggest that social bookmarking might help with various aspects of web search.

5.1 URLs

Result 1: Pages posted to del.icio.us are often recently modified.
Conclusion: del.icio.us users post interesting pages that are actively updated or have been recently created.

Determining the approximate age of a web page is fraught with challenges. Many pages corresponding to on disk documents will return the HTTP/1.1 `Last-Modified` header accurately. However, many dynamic web sites will return a Last-Modified date which is the current time (or another similar time for caching purposes), and about $\frac{2}{3}$ of pages in Dataset M do not return the header at all! Fortunately, search engines need to solve this problem for crawl ordering. They likely use a variety of heuristics to determine if page content has changed significantly. As a result, the Yahoo! Search API gives a `ModificationDate` for all result URLs which it returns. While the specifics are unknown, ModificationDate appears to be a combination of the Last-Modified HTTP/1.1 header, the time at which a particular page was last crawled and its page content. We used this API to test the recency of five groups of pages:

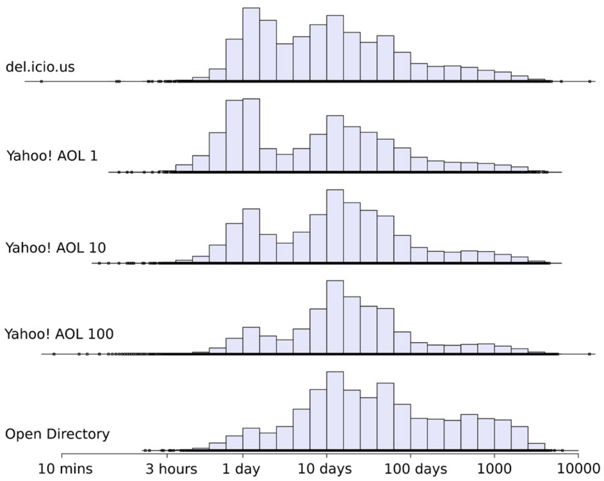


Figure 3: Histograms showing the relative distribution of ages of pages in del.icio.us, Yahoo! Search results and ODP.

del.icio.us Pages sampled from the del.icio.us recent feed as they were posted.

Yahoo! 1, 10, and 100 The top 1, 10, and 100 results (respectively) of Yahoo! searches for queries sampled from the AOL query dataset.

ODP Pages sampled from the Open Directory Project (dmoz.org).

Rather than compare the age of del.icio.us pages to random pages from the web (which would neither be possible nor meaningful), we chose the four comparison groups to represent groups of pages a user might encounter. The Yahoo 1, 10, and 100 groups represent pages a user might encounter as a result of searches. ODP represents pages a user might encounter using an Internet directory, and is also probably more representative of the web more broadly. For each URL in each set, we recorded the time since the page was last modified. In order to avoid bias by time, we ran equal proportions of queries for each set at similar times.

Figure 3 shows the results. Each bar represents the number of pages in the group with the given (x-axis) age. We found that pages from del.icio.us were usually more recently modified than ODP, which tends to have older pages. We also found that there is a correlation between a search result being ranked higher and a result having been modified more recently. However, most interestingly, we found that the top 10 results from Yahoo! Search were about the same age as the pages found bookmarked in del.icio.us. This could be interpreted in one of two ways: (i) del.icio.us is getting recent, topical bookmarks which Yahoo! Search is trying to emulate, or (ii) del.icio.us is getting bookmarks which are a result of searches, and thus have the same recency as the top 10.

Result 2: Approximately 25% of URLs posted by users are new, unindexed pages.
Conclusion: del.icio.us can serve as a (small) data source for new web pages and to help crawl ordering.

We next looked at what proportion of pages were "new" in the sense that they were not yet indexed by a search engine at the time they were posted to del.icio.us. We sampled pages from the del.icio.us recent feed as they were posted, and then ran Yahoo! searches for those pages immediately after. Of those pages, about 42.5% were not found. This could be for a variety of reasons—the pages could be indexed under another canonicalized URL, they could be spam, they could be an odd MIME-type (an image, for instance) or the page could have not been found yet. Anecdotally, all four of these causes appear to be fairly common in the set of sampled missing URLs. As a result, we next followed up by continuously searching for the missing pages over the course of the following five months. When a missing page appears in a later result, we argue that the most likely reason is that the page was not indexed but was later crawled. This methodology seems to eliminate the possibility that spam and canonicalization issues are the reason for missing URLs, but does not eliminate the possibility, for instance, that multiple datacenters give out different results.

We found that of the $5,724$ URLs which we sampled and were missing from the week beginning June 22, $3,427$ were later found and $1,750$ were found within four weeks. This implies that roughly 60% of the missing URLs were in fact new URLs, or roughly 25% of del.icio.us (i.e., $42.5\% \times 60\%$). This works out to roughly $30,000$ new pages per day.

Social bookmarking seems to be a good source of new and active pages. As a source of new pages, social bookmarking may help a search engine discover pages it might not otherwise. For instance, Dasgupta et al. [5] suggest that 25% of new pages are not discoverable using historical information about old pages. As a source of both new and active pages, social bookmarking may also help more generally with the "crawl ordering" problem—should we update old pages, or try to discover new pages? To the extent to which social bookmarks represent "interesting" changes to pages, they should be weighted in crawl ordering schemes.

Result 3: Roughly 9% of results for search queries are URLs present in del.icio.us.
Conclusion: del.icio.us URLs are disproportionately common in search results compared to their coverage.

Similarly to the recently modified pages discussion above, we used queries chosen by sampling from the AOL query dataset to check the coverage of results by del.icio.us. Specifically, we randomly sampled queries from the query dataset, ran them on Yahoo! Search, and then cross-referenced them with the millions of unique URLs present in Datasets C, M, and R. When we randomly sample, we sample over query events rather than unique query strings. This means that the query "american idol" which occurs roughly $15,000$ times, is about five times more likely to be picked than "powerball" which occurs roughly $3,000$ times.

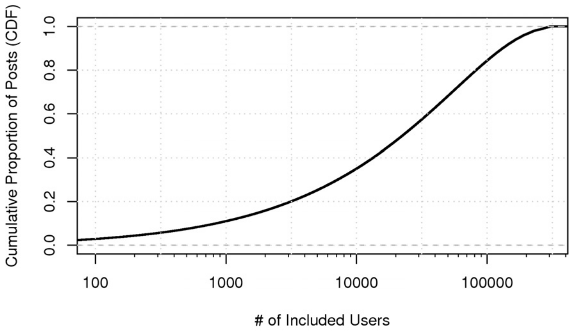


Figure 4: Cumulative Portion of del.icio.us Posts Covered by Users

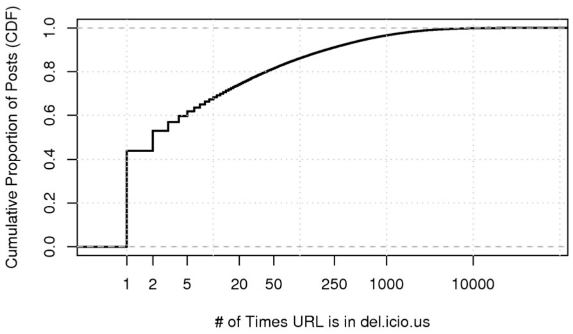


Figure 5: How many times has a URL just posted been posted to del.icio.us?

We found that despite the fact that del.icio.us covers a relatively small portion of the web (see discussion below in Result 9), it covers a disproportionately high proportion of search results. For the top 100 results of the queries, del.icio.us covers 9% of results returned for a set of over 30,000 queries. For the top 10 results, this coverage is about double: 19% of results returned are in del.icio.us. This set of queries is weighted towards more popular queries, which can explain part of this effect. By comparison, we might expect $\frac{1}{1000}$ of URLs in query results to be in del.icio.us if they were selected at random from the web (again, see Result 9). This suggests that to whatever extent del.icio.us gives us additional metadata about web pages, it may lead to result reordering for queries.

Result 4: While some users are more prolific than others, the top 10% of users only account for 56% of posts.
Conclusion: del.icio.us is not highly reliant on a relatively small group of users (e.g., $< 30,000$ users).

Figure 4 shows the extent to which the most prolific users are responsible for large numbers of posts. While there are some URLs, domains, users, and tags that cover many posts or triples, the distributions do not seem so condensed as to be problematic. For instance, on social news sites, it is commonly cited that the majority of front page posts come from a dedicated group of less than 100 users. However, the majority of posts in Dataset M instead come from tens of thousands of users. Nonetheless, the distribution is still power law shaped and there is a core group of relatively active users and a long tail of relatively inactive users.

Result 5: 30-40% of URLs and approximately one in eight domains posted were not previously in del.icio.us.
Conclusion: del.icio.us has relatively little redundancy in page information.

The recent feed states for each post how many times the URL in that post is already in del.icio.us. Figure 5 shows the distribution of this value. A new post in Dataset M is of a new URL not yet in the system about 40% of the time. This proportion might be 30% of total posts to del.icio.us if we adjust for filtering. In Dataset M, a majority of the URLs posted were only posted once during the time period.

Another way to look at new URLs being added to del.icio.us is in terms of how often a completely new domain is added (as opposed to just another URL at an existing domain). Unfortunately, we do not know the exact set of domains in del.icio.us. However, we can provide an upper-bound by comparing against the domains in Datasets C and R. We found that about 12% of posts in Dataset M were URLs whose domains were not in either Dataset C or R. This suggests that about one eighth of the time, a new URL is not just a new page to be crawled, but may also suggest an entire new domain to crawl.

This result coupled with Result 4 may impact the potential actions one might use to fight tag spam. Because of the relatively high number of new pages, it may be more difficult for those pages to determine the quality of labels placed on them. Furthermore, due to the relatively low number of label redundancies, it may be difficult to determine the trustworthiness of a user based on coincident labels with other users (as in, e.g., [11]). For instance, 85% of the labels in Dataset M are non-redundant. As a result, it may become increasingly important to use interface-based methods to keep attackers out rather than analyzing the data that they add to the system. However, on the other hand, the low level of redundancy does mean that users are relatively efficient in labeling the parts of the web that they label.

5.2 Tags

Result 6: Popular query terms and tags overlap significantly (though tags and query terms are not correlated).
Conclusion: del.icio.us may be able to help with queries where tags overlap with query terms.

One important question is whether the metadata attached to bookmarks is actually relevant to web searches. That is, if popular query terms often appear as tags, then we would expect the tags to help guide users to relevant pages. SocialSimRank [1] suggests an easy way to make use of this information. We opted to look at tag–query overlap between the tags in Dataset M and the query terms in the AOL query dataset. For this analysis, we did not attempt to remove "stop tags"—tags like "imported" that were automatically added by the system or otherwise not very meaningful. Figure 6 shows the number of times a tag occurs in Dataset M versus the number of times it occurs in the AOL query dataset. Table 1 shows the corresponding query term rank for the top 30 del.icio.us tags in Dataset M. Both show that while there was a reasonable degree of overlap between query terms and tags, there was no positive correlation between popular tags and popular query terms.

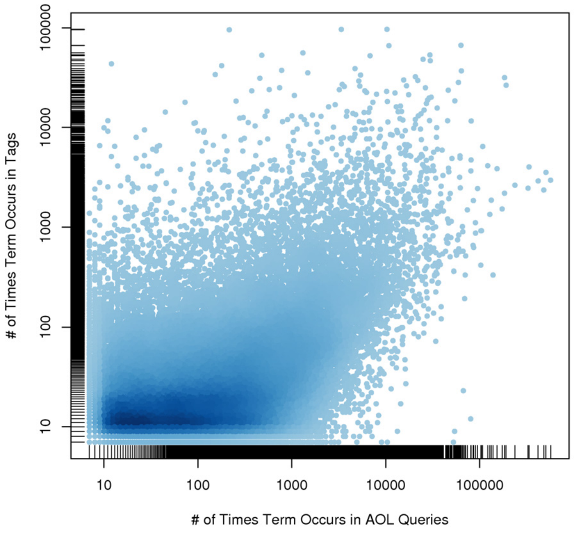


Figure 6: A scatter plot of tag count versus query count for top tags and queries in del.icio.us and the AOL query dataset. $r \approx 0.18$. For the overlap between the top 1000 tags and queries by rank, $\tau \approx 0.07$.

One likely reason the two are uncorrelated is that search queries are primarily navigational, while tags tend to be used primarily for browsing or categorizing. For instance, 21.9% of the AOL query dataset is made up of queries that look like URLs or domains, e.g., `www.google.com` or `http://i.stanford.edu/` and variations. To compute the overlap between tags and queries (but not for Figure 6), we first removed these URL or domain-like queries from consideration. We also removed certain stopword like tags, including "and", "for", "the", and "2.0" and all tags with less than three characters. We found that at least one of the top 100, 500, and 1000 tags occurred in 8.6%, 25.3% and 36.8% of these non-domain, non-URL queries.

In some sense, overlap both overstates and understates the potential coverage. On one hand, tags may correlate with but not be identical to particular query terms. However, on the other, certain tags may overlap with the least salient parts of a query. We also believe that because AOL and del.icio.us represent substantially different communities, the query terms are a priori less likely to match tags than if we had a collection of queries written by del.icio.us users.

Result 7: In our study, most tags were deemed relevant and objective by users.
Conclusion: Tags are on the whole accurate.

One concern is that tags at social bookmarking sites may be of "low quality." For example, perhaps users attach nonsensical tags (e.g., "fi32") or very subjective tags (e.g., "cool"). To get a sense of tag quality, we conducted a small user

Tag (Rank)	# Queries (Rank)	Tag (Rank)	# Queries (Rank)	Tag (Rank)	# Queries (Rank)
design (#1)	10318 (#545)	web (#11)	24992 (#184)	travel (#21)	20703 (#227)
blog (#2)	3367 (#1924)	video (#12)	29833 (#127)	free (#22)	184569 (#9)
imported (#3)	215 (#18292)	webdesign (#13)	11 (#155992)	css (#23)	456 (#10624)
music (#4)	63250 (#41)	linux (#14)	178 (#20937)	education (#24)	15546 (#335)
software (#5)	10823 (#506)	photography (#15)	4711 (#1384)	business (#25)	21970 (#212)
reference (#6)	1312 (#4655)	tutorial (#16)	779 (#7098)	flash (#26)	5170 (#1274)
art (#7)	29558 (#130)	news (#17)	63916 (#40)	games (#27)	59480 (#49)
programming (#8)	478 (#10272)	blogs (#18)	1478 (#4205)	mac (#28)	3440 (#1873)
tools (#9)	6811 (#921)	howto (#19)	152 (#23341)	google (#29)	191670 (#8)
web2.0 (#10)	0 (None)	shopping (#20)	5394 (#1222)	books (#30)	16643 (#296)

Table 1: Top tags and their rank as terms in AOL queries.

study. We had a group of ten people, a mix of graduate students and individuals associated with our department, manually evaluate posts to determine their quality. We sampled one post out of every five hundred, and then gave blocks of posts to different individuals to label. Most of the individuals labeled about 100 to 150 posts. For each tag, we asked whether the tag was "relevant," "applies to the whole domain," and/or "subjective." For each post, we asked whether the URL was "spam," "unavailable," and a few other questions. We set the bar relatively low for "relevance": whether a random person would agree that it was reasonable to say that the tag describes the page. Roughly 7% of tags were deemed "irrelevant" according to this definition. Also, remarkably few tags were deemed "subjective": less than one in twenty for all users. Lastly, there was almost no "spam" in the dataset, either due to low amounts of spam on del.icio.us, or due to the filtering described in Section 3.

6. NEGATIVE FACTORS

In this section, we present negative factors which suggest that social bookmarking might not help with various aspects of web search.

6.1 URLs

Result 8: Approximately 120,000 URLs are posted to del.icio.us each day.
Conclusion: The number of posts per day is relatively small; for instance, it represents about $\frac{1}{10}$ of the number of blog posts per day.

Figure 7 shows the posts per hour for every hour in Dataset M. The dashed lines show (where available) the independently sampled data collected by Philipp Keller.[5] Keller's data comes from sampling the recent feed every 10 minutes and extrapolating based on the difference in age between the youngest and oldest bookmark in the fixed size feed. Dataset M comes from attempting to capture every post in the recent feed. The two datasets seem to be mutually reinforcing—our data only differs from Keller's slightly, and this usually occurs at points where the feed "crashed." At these points, near June 3rd and June 15th respectively in Figure 7, the feed stopped temporarily, and then restarted, replaying past bookmarks until it caught up to the present.

There are an average of 120,087 posts per day in Dataset M. However, more relevant for extrapolation are the number of posts in a given week. On average, 92,690 posts occurred per day of each weekend, and 133,133 posts occurred each weekday. Thus, del.icio.us currently produces about 851,045 posts per week, or a little more than 44 million posts per year. For comparison, David Sifry [15] suggests that there are on the order of 1.5 million blog posts per day. This means that for every bookmark posted to del.icio.us, ten blog entries will be posted to blogs on the web.

More important than the current rate at which posts are being generated is the rate at which posts per day are accelerating. However, this rate of acceleration is harder to determine. For instance, Dataset M shows a 50% jump in posts per hour on the evening of May 30th, when del.icio.us announced a partnership with Adobe. However, we believe that this may have simply been bouncing back from a previous slump. Keller's data, shown in Figure 8 seems to tell multiple stories. From August 2005, until August 2006 (including December 2005, when del.icio.us was bought), del.icio.us seems to have been accelerating at a steady rate. However, from November 2006 to June 2007, the rate of acceleration seems to be flat. Our Dataset R, while not covering the same length of time, does not lead us to reject Keller's data. As a result, we believe that the history of social bookmarking on del.icio.us seems to be a series of increases in posting rate followed by relative stability. To the extent to which this is the case, we believe that future rates of increase in posts per day are highly dependent on external factors and are thus not easily predictable.

Result 9: There are roughly 115 million public posts, coinciding with about 30-50 million unique URLs.
Conclusion: The number of total posts is relatively small; for instance, this is a small portion (perhaps $\frac{1}{1000}$) of the web as a whole.

Relatively little is known about the size of social bookmarking sites, and in particular del.icio.us. In September 2006, del.icio.us announced that they had reached 1 million users, and in March 2007, they announced they had reached 2 million. The last official statement on the number of unique posts and URLs was in May of 2004, when del.icio.us' creator, Joshua Schacter stated that there were about 400,000 posts and 200,000 URLs.

One way to estimate the size of del.icio.us is to extrapolate from some set of URLs or tags. For instance, if the URL `http://www.cnn.com/` was posted u_m times in a one month period, there were t_m posts total during that month, and the URL had been posted to the system a total of u_s times,

[5] Available at `http://deli.ckoma.net/stats`.

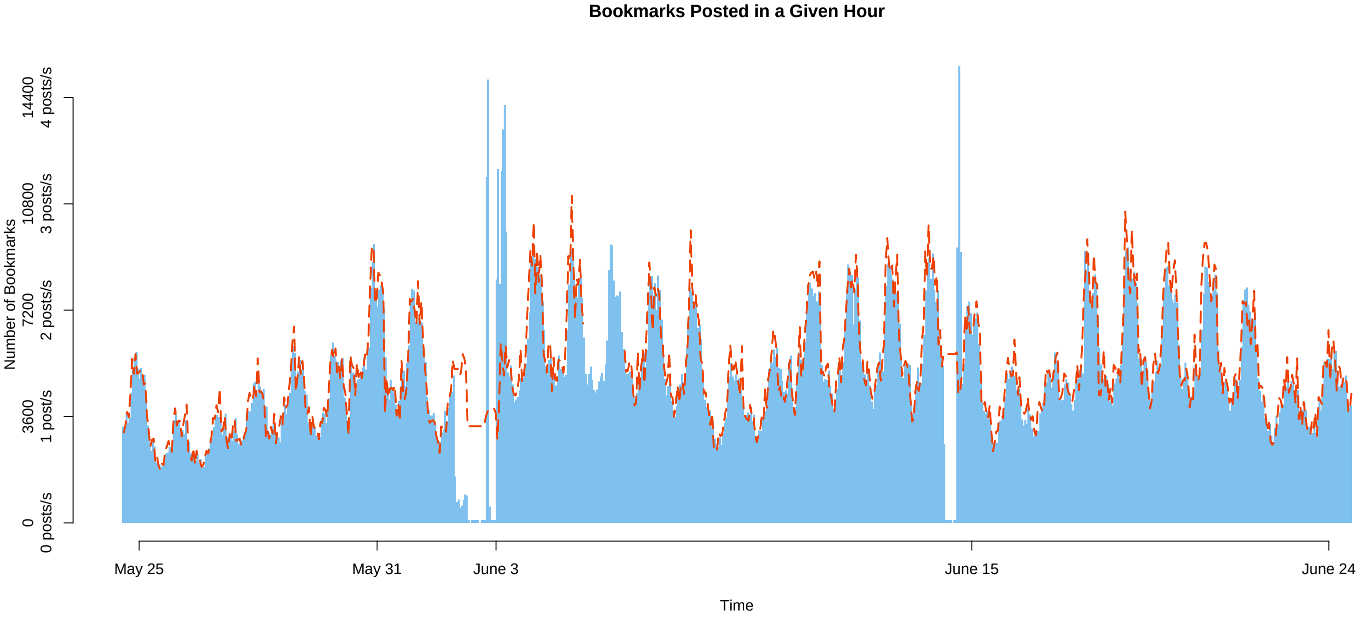


Figure 7: Posts per hour and comparison to Philipp Keller.

we might estimate the total size t_s of del.icio.us as $t_s = \frac{u_s t_m}{u_m}$ (assuming $\frac{u_m}{t_m} = \frac{u_s}{t_s}$). However, we found that this led to poor estimates—often in the billions of posts.

Instead, we assume that the rate of posting of URLs to del.icio.us has been monotonically increasing (given a sufficient time window) since its creation. We then divide the historical record of del.icio.us into three time periods. The first, t_1, is the period before Schacter's announcement on May 24th. The second, t_2, is between May 24th and the start of Keller's data gathering. The third, t_3, is from the start of Keller's data gathering to the present.

We assume that t_1 is equal to $400,000$ posts. We estimate that t_2 is equal to the time period (about $p_1 = 420$ days) times the maximum amount of posts per day in the one month period after Keller's data starts ($d_b = 44,536$) times a filtering factor ($f = 1.25$) to compensate for the filtering which we observed during our data gathering. We estimate that t_3 is equal to the posts observed by Keller (o_k), plus the posts in the gaps in Keller's data gathering (g_k). o_k is $n_k = 58,194,463$ posts, which we multiply by the filtering factor ($f = 1.25$). We estimate g_k as the number of days missing ($m_k = 104$) times the highest number of posts for a given day observed by Keller ($d_k = 161,937$) times the filtering factor ($f = 1.25$).

Putting this all together, we estimate that the number of posts in del.icio.us as of late June 2007 was:

$$\begin{aligned} & t_1 + t_2 + t_3 \\ = \; & (400000) + (p_1 \times d_b \times f) + (n_k \times f + m_k \times d_k \times f) \\ = \; & (400000) + (420 \times 44536 \times 1.25) + \\ & (58194463 \times 1.25 + 104 \times 161937 \times 1.25) \\ \approx \; & 117 \text{ million posts} \end{aligned}$$

This estimate is likely an over-estimate because we choose upper bound values for d_b and d_k. Depending on the real values of $\{d_b, d_k, f\}$, one could reasonably estimate the number of posts anywhere between about 60 and 150 million posts. It should be noted that this does not, however, include private (rather than public) posts, which we do not have any easy way to estimate. Finally, we estimate that between about 20 and 50 percent of posts are unique URLs (see discussion in Result 4 and Figure 5). This leads us to an estimate of about 12 to 75 million unique URLs.

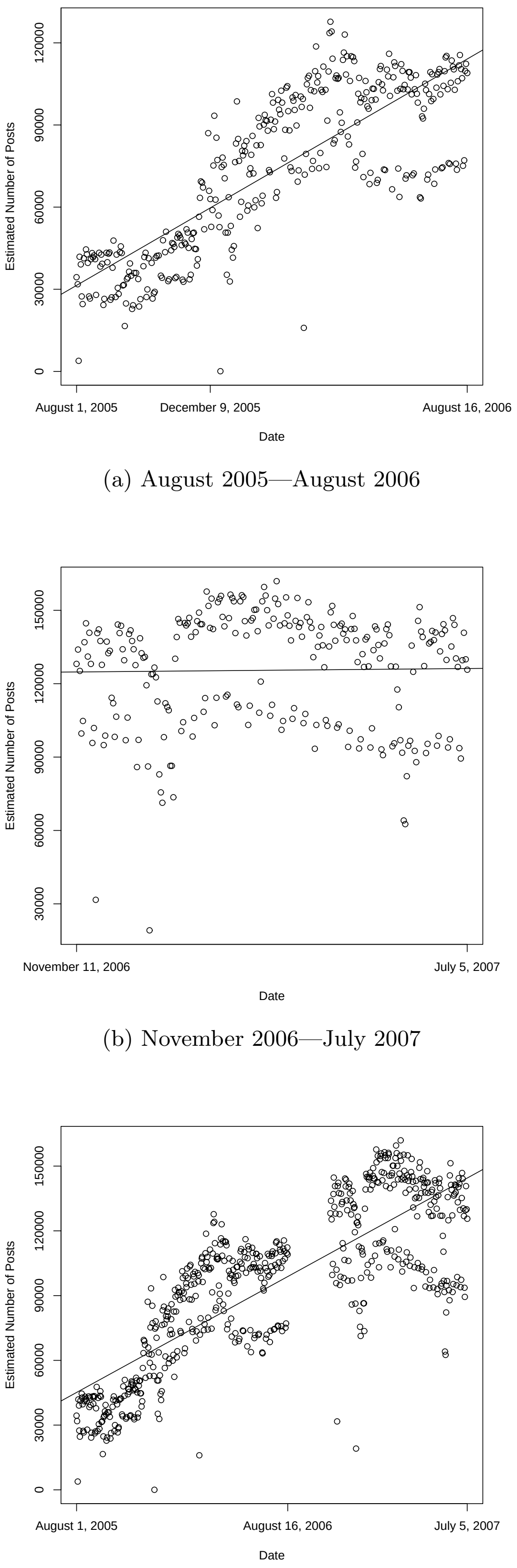
The indexes of the major search engines are now commonly believed to be in the billions to hundreds of billions of pages. For instance, Eiron et al. [7] state in 2004 that after crawling for some period of time, their crawler had explored 1 billion pages and had 4.75 billion pages remaining to be explored. Of course, as dynamic content has proliferated on the web, such estimates become increasingly subjective. Nonetheless, the number of unique URLs in del.icio.us is relatively small as a proportion of the web as a whole.

6.2 Tags

Result 10: Tags are present in the pagetext of 50% of the pages they annotate and in the titles of 16% of the pages they annotate.
Conclusion: A substantial proportion of tags are obvious in context, and many tagged pages would be discovered by a search engine.

For a random sampling of over 20,000 posts in Dataset M, we checked whether tags were in the text of the pages they annotate or related pages. To get plain text from pages, we used John Cowan's TagSoup Java package to convert from HTML.[6] To get tokens from plain text, we used the Stan-

[6] TagSoup is available at `http://ccil.org/~cowan/XML/`



(a) August 2005—August 2006

(b) November 2006—July 2007

(c) August 2005—July 2007

Figure 8: Details of Keller's post per hour data.

Host % of Tag	Tag % of Host	Host
5.0%	87.7%	java.sun.com
3.2%	81.5%	onjava.com
3.1%	82.0%	javaworld.com
1.6%	67.9%	theserverside.com
1.3%	88.7%	today.java.net

Table 2: This example lists the five hosts in Dataset C with the most URLs annotated with the tag *java* and the percentage of the URLs bookmarked at that host tagged *java*.

ford NLP Group's implementation of the Penn Treebank Tokenizer.[7] We also checked whether pages were likely to be in English or not, using Marco Olivo's lc4j Language Categorization package.[8] Finally, we lowercased all tags and all tokens before doing comparisons.

We found that 50% of the time, if a tag annotates a page, then it is present in the page text. Furthermore, 16% of the time, the tag is not just anywhere in the page text, but it is present in the title. We also, looked at the page text of pages that link to the URL in question (backlinks) and pages that are linked from the URL in question (forward links). 20% of the time, a tag annotating a particular page will appear in *three* places: the page it annotates, at least one of its backlinks, and at least one of its forward links. 80% of the time, the tag will appear in at least one of these places: the page, backlinks or forward links. Anecdotally, the tags in the missing 20% appear to be "lower quality." They tend to be mistakes of various kinds (misspellings or mistypes of tags) or confusing tagging schemes (like "food/dining"). Overall, this seems to suggest that a search engine, which is already looking at page text and particularly at titles (and sometimes at linked text), is unlikely to gain much from tag information in a significant number of cases.

Result 11: Domains are often highly correlated with particular tags and vice versa.
Conclusion: It may be more efficient to train librarians to label domains than to ask users to tag pages.

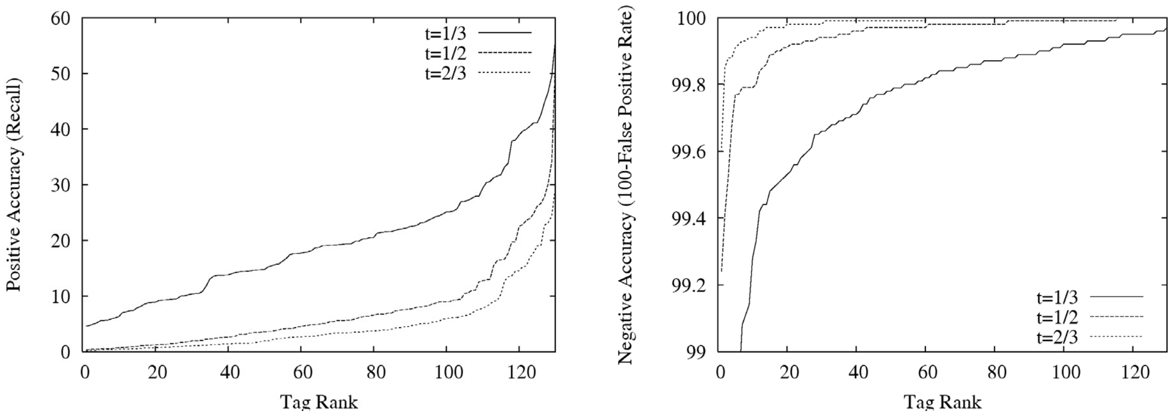
One way in which tags may be predicted is by host. Hosts tend to be created to focus on certain topics, and certain topics tend to gravitate to a few top sites focusing on them. For instance, Table 2 shows the proportion of the URLs in Dataset C labeled "java" which are on particular hosts (first column). It also shows the proportion of the URLs at those hosts which have been labeled "java" (second column). This table shows that 14 percent of the URLs that are annotated with the tag *java* come from five large topical Java sites where the majority of URLs are in turn tagged with *java*.

Unfortunately, due to the filtering discussed in Section 3.4 we could not use Dataset M for our analysis. Instead, we use Dataset C, with the caveat that based on our discussions in Section 3.4 and Result 9, Dataset C represents about 25% of the posts in del.icio.us, biased towards more popular URLs,

`tagsoup/`.

[7] The PTB Tokenizer is available at `http://nlp.stanford.edu/javanlp/`—we used the version from the Stanford NER.

[8] lc4j is available at `http://www.olivo.net/software/lc4j/` and implements algorithms from [2].



(a) Accuracy on Positive Examples

(b) Accuracy on Negative Examples

Figure 9: Host Classifier: The accuracy for the first 130 tags by rank for a host-based classifier.

	Avg Accuracy (+)	Avg Accuracy (-)
$\tau = 0.33$	19.647	99.670
$\tau = 0.5$	7.372	99.943
$\tau = 0.66$	4.704	99.984

Table 3: Average accuracy values for different τ.

users, and tags. As a result, one should not assume that the conclusions from this section apply to all of del.icio.us as opposed to the more concentrated section of Dataset C.

We denote the number of URLs tagged with a tag t_i at a given host d_j as $tagged(t_i, d_j)$, and the total number of URLs at that host in the tagging corpus as $total(d_j)$. We can construct a binary classifier for determining if a particular URL o_k having host d_j should be annotated with tag t_i with the simple rule:

$$classify(t_i, d_j) = \begin{Bmatrix} t & : & \frac{tagged(t_i,d_j)}{total(d_j)} > \tau \\ \neg t & : & \frac{tagged(t_i,d_j)}{total(d_j)} \leq \tau \end{Bmatrix}$$

where τ is some threshold. We define the positive accuracy to be the rate at which our classifier labels positive examples correctly as positives, and negative accuracy to be the rate at which our classifier correctly labels negative examples correctly as negatives. Further, we define the macro-averaged positive and negative accuracies, given in Table 3, as the mean of the positive and negative accuracies—with each tag weighted equally—for the top 130 tags, respectively.

This classifier allows us to predict (simply based on the domain) between about five and twenty percent of the tag annotations in Dataset C, with between a few false positives per $1,000$ and a few per $10,000$. We also show the accuracies on positive and negative examples in Figure 9. All experiments use leave-one-out cross validation. Our user study (described in Result 7) also supported this conclusion. About 20% of the tags which were sampled were deemed by our users to "apply to the whole domain." Because our user study and our experiments above were based on differently biased datasets, Datasets C and M, they seem to be mutually reinforcing in their conclusions. Both experiments suggest that a human librarian capable of labeling a host with a tag on a host-wide basis (for instance, "java" for `java.sun.com`) might be able to make substantial numbers of user contributed labels redundant.

7. DISCUSSION

The eleven results presented in the past two sections paint a mixed picture. We found that social bookmarking as a data source for search has URLs that are often actively updated and prominent in search results. We also found that tags were overwhelmingly relevant and objective. However, del.icio.us produces small amounts of data on the scale of the web. Furthermore, the tags which annotate URLs, while relevant, are often functionally determined by context. Nearly one in six tags are present in the title of the page they annotate, and one in two tags are present in the page text. Aside from page content, many tags are determined by the domain of the URL that they annotate, as is the case with the tag "java" for "java.sun.com." These results suggest that URLs produced by social bookmarking are unlikely to be numerous enough to impact the crawl ordering of a major search engine, and the tags produced are unlikely to be much more useful than a full text search emphasizing page titles.

All is not doom and gloom however. Specifically, if social bookmarking continues to grow at the rate it has over the past several years (rather than the past several months) then it will rapidly reach the scale of the current web. In terms of tags, we believe that user interface features could have a large impact on improving the quality of tags for search. For instance, interfaces that recommended tags not in the page, or not common for the given domain, might help alleviate those two problems. Another approach might be to have domain-specific sites (e.g., photography) which might have higher quality tags due to the shared context of the users. We believe that the challenges outlined in this paper can be met in the future, but only time will tell.

8. ACKNOWLEDGMENTS

The authors would like to thank Mark Lucovsky at Google for his invaluable assistance with the millions of automated realtime backlink queries necessary to prepare this work. Paul Heymann is supported by an NSF Graduate Research Fellowship.

9. REFERENCES

[1] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing Web Search Using Social Annotations. In *WWW '07: Proceedings of the 16th International Conference on World Wide Web*, pages 501–510, New York, NY, USA, 2007. ACM.

[2] W. Cavnar and J. Trenkle. N-Gram-Based Text Categorization. *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1994.

[3] H. Chen. Collaborative Systems: Solving the Vocabulary Problem. *IEEE Computer, Special Issue on CSCW*, 27(5):58–66, May 1994.

[4] N. Craswell, D. Hawking, and S. Robertson. Effective Site Finding Using Link Anchor Information. In *SIGIR '01: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 250–257, New York, NY, USA, 2001. ACM.

[5] A. Dasgupta, A. Ghosh, R. Kumar, C. Olston, S. Pandey, and A. Tomkins. The Discoverability of the Web. In *WWW '07: Proceedings of the 16th International Conference on World Wide Web*, pages 421–430, New York, NY, USA, 2007. ACM.

[6] N. Eiron and K. S. McCurley. Analysis of Anchor Text for Web Search. In *SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 459–460, New York, NY, USA, 2003. ACM.

[7] N. Eiron, K. S. McCurley, and J. A. Tomlin. Ranking the Web Frontier. In *WWW '04: Proceedings of the 13th International Conference on World Wide Web*, pages 309–318, New York, NY, USA, 2004. ACM.

[8] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The Vocabulary Problem in Human-System Communication. *Communications of the ACM*, 30(11):964–971, 1987.

[9] S. Golder and B. A. Huberman. Usage Patterns of Collaborative Tagging Systems. *Journal of Information Science*, 32(2):198–208, April 2006.

[10] H. Halpin, V. Robu, and H. Shepherd. The Complex Dynamics of Collaborative Tagging. In *WWW '07: Proceedings of the 16th International Conference on World Wide Web*, pages 211–220, New York, NY, USA, 2007. ACM.

[11] G. Koutrika, F. A. Effendi, Z. Gyöngyi, P. Heymann, and H. Garcia-Molina. Combating Spam in Tagging Systems. In *AIRWeb '07: Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web*, pages 57–64, New York, NY, USA, 2007. ACM.

[12] C. Marlow, M. Naaman, D. Boyd, and M. Davis. HT06, tagging paper, taxonomy, Flickr, academic article, to read. In *HYPERTEXT '06: Proceedings of the Seventeenth Conference on Hypertext and Hypermedia*, pages 31–40, New York, NY, USA, 2006. ACM.

[13] G. Pass, A. Chowdhury, and C. Torgeson. A Picture of Search. In *InfoScale '06: Proceedings of the 1st International Conference on Scalable Information Systems*, page 1, New York, NY, USA, 2006. ACM.

[14] S. Sen, S. K. Lam, A. M. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, F. M. Harper, and J. Riedl. tagging, communities, vocabulary, evolution. In *CSCW '06: Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work*, pages 181–190, New York, NY, USA, 2006. ACM.

[15] D. Sifry. State of the Live Web: April 2007. `http://www.sifry.com/stateoftheliveweb/`.

[16] Y. Yanbe, A. Jatowt, S. Nakamura, and K. Tanaka. Can Social Bookmarking Enhance Search in the Web? In *JCDL '07: Proceedings of the 2007 Conference on Digital Libraries*, pages 107–116, New York, NY, USA, 2007. ACM.