Social Tag Prediction

Paul Heymann, Daniel Ramage, and Hector Garcia-Molina
Department of Computer Science
Stanford University, Stanford, CA, USA
{heymann, dramage, hector}@cs.stanford.edu

ABSTRACT

In this paper, we look at the "social tag prediction" problem. Given a set of objects, and a set of tags applied to those objects by users, can we predict whether a given tag could/should be applied to a particular object? We investigated this question using one of the largest crawls of the social bookmarking system del.icio.us gathered to date. For URLs in del.icio.us, we predicted tags based on page text, anchor text, surrounding hosts, and other tags applied to the URL. We found an entropy-based metric which captures the generality of a particular tag and informs an analysis of how well that tag can be predicted. We also found that tag-based association rules can produce very high-precision predictions as well as giving deeper understanding into the relationships between tags. Our results have implications for both the study of tagging systems as potential information retrieval tools, and for the design of such systems.

Categories and Subject Descriptors

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

General Terms

Design, Experimentation, Human Factors, Measurement

1. INTRODUCTION

Social tags (keyword annotations) have recently emerged as a popular way to allow users to contribute metadata to large and dynamic corpora. Standard taxonomies force objects into predefined categories. By contrast, tags have no such requirement. This makes tags appropriate for corpora like the web and user contributed video and photo collections where the distribution or type of content may change rapidly. However, despite increased interest in tagging, tags are still poorly understood. In particular, little is known about the predictability of tags. Given a set of objects, and a set of tags applied to those objects by users, can we predict whether a given tag could/should be applied to a particular object? We call this the "social tag prediction" problem. In this paper, we look at how effective different types of data are at predicting tags in a tagging system.

Solving the social tag prediction problem has two benefits. At a fundamental level, we gain insights into the "information content" of tags: that is, if tags are easy to predict from other content, they add little value. At a practical level, we can use a tag predictor to enhance a social tagging site. These enhancements can take a variety of forms:

Increase Recall of Single Tag Queries/Feeds Many, if not most, queries in tagging systems are for objects labeled with a particular tag. Similarly, many tagging systems allow users to monitor a feed of items tagged with a particular tag. For example, a user of a social bookmarking site might set up a feed of all "photography" related web pages. Tag prediction could serve as a recall enhancing device for such queries and feeds. In Section 4.2, we set up such a recall enhancing tag prediction task.

Inter-User Agreement Many users have similar interests, but different vocabularies. Tag prediction would ease sharing of objects despite vocabulary differences.

Tag Disambiguation Many tags are polysemous, that is, they have different meanings. For example, "apple" might mean the fruit, or the computer company. Predicting additional tags (like "macos" or "computer") might aid in disambiguating what a user meant when annotating an object. Past work by Aurnhammer et al. [2] looks at similar issues in photo tagging.

Bootstrapping Sen et al. [16] find that the way users use tags is determined by previous experience with tags in the system. For example, in systems with low tag usage, fewer users will apply tags. If tag usage in the system is mostly personal tags, users tend to apply more personal tags. Using tag prediction, a system designer could pre-seed a system with appropriate tags to encourage quality contributions from users.

System Suggestion Some tagging systems provide tag suggestions when a user is annotating an object (see for example, Xu et al. [18]). Predicted tags might be reasonable to suggest to users in such a system. However, unlike the other applications in this list, it might be more informative for the system to suggest tags that it is unsure of to see if the user selects them.

In this paper, we use one of the largest tagging datasets available, the Stanford Tag Crawl 2007 dataset based on the

SIGIR'08, July 20–24, 2008, Singapore.

del.icio.us social bookmarking site. We examine whether tags are predictable based on the page text, anchor text, and surrounding domains of pages they annotate. We find that there is a high variance in the predictability of tags, and we look at metrics associated with predictability. One such metric, a novel entropy measure, captures a notion of generality that we think might be helpful for other tasks in tagging systems. Next, we look at how to predict tags based on other tags annotating a URL. We find that we can expand a small set of tags with high confidence. We conclude with a summary of our findings and their broader implications for tagging systems and web search.

2. PRELIMINARIES

A social tagging system consists of users $u \in U$, tags $t \in T$, and objects $o \in O$. We call an annotation of a set of tags to an object by a user a *post*. A post is made up of one or more (t_i, u_j, o_k) *triples*. We imagine that every object o has a vast set of tags that do not describe it, a smaller set of tags which do describe it, and an even smaller set of tags which users have actually chosen to input into the system as applicable to the object. We say that the first set of tags *negatively describes* the object, the second set of tags *positively describes* the object, and the last set of tags currently *annotates* the object. We model each of these three relationships as relations or tables:

R_p A set of (t, o) pairs where each pair means that tag t positively describes object o.

R_n A set of (t, o) pairs where each pair means that tag t negatively describes object o.

R_a A set of (t, u, o) triples where each triple means that user u annotated object o with tag t.

In practice, the system owner only has access to R_a.

We manipulate the relations R_p, R_n, and R_a using two standard relational algebra operators with set semantics. Selection, or σ_c selects tuples from a relation where a particular condition c holds. Projection, or π_p projects a relation into a smaller number of attributes. σ_c is equivalent to the `WHERE c` clause in SQL whereas π_p is equivalent to the `SELECT p` clause in SQL. σ_c can be read as "select all tuples satisfying c." π_p can be read as "show only the attributes in p from each tuple."

Suppose a tagging system had only two objects, a web page o_{bagels} about a downtown bagel shop and a web page o_{pizza} about a pizzeria next door. We might have:

$$\begin{aligned} R_p &= (t_{bagels}, o_{bagels}), (t_{shop}, o_{bagels}), (t_{downtown}, o_{bagels}), \\ &\quad (t_{pizza}, o_{pizza}), (t_{pizzeria}, o_{pizza}) \\ R_n &= (t_{pizzeria}, o_{bagels}), (t_{pizza}, o_{bagels}), (t_{bagels}, o_{pizza}) \dots \end{aligned}$$

If we want to know all of the tags which positively describe o_{bagel}, we would write $\pi_t(\sigma_{o_{bagel}}(R_p))$ and the result would be $(t_{bagels}, t_{shop}, t_{downtown})$. If we want all (t, o) pairs which do not describe o_{pizza}, we would write $\pi_{(t,o)}(\sigma_{o_{pizza}}(R_n))$. Suppose also that a user u_{sally} has annotated the pizzeria web page with the tag $t_{pizzeria}$:

$$R_a = (t_{pizzeria}, u_{sally}, o_{pizza})$$

If we want to know all users who have tagged o_{pizza}, we would write $\pi_u(\sigma_{o_{pizza}}(R_a))$ and the result would be (u_{sally}).

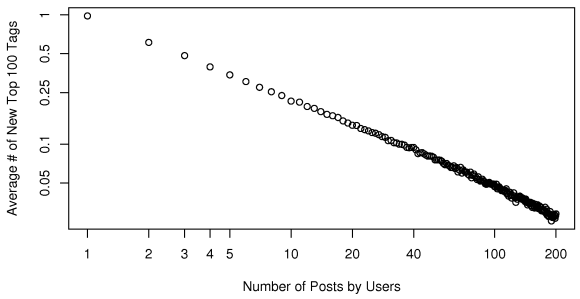

Figure 1: Average new tags versus number of posts.

3. DATASET

The Stanford Tag Crawl Dataset consists of one contiguous month of data starting May 25th 2007. This data was gathered from the del.icio.us recent feed. The recent feed is filtered by del.icio.us, so it is likely that the recent feed is only a significant portion of all posts during the period. For each URL posted to the recent feed, the dataset also contains a crawl of that URL within 2 hours of its posting, pages linked from that URL, and inlinks to the URL. The dataset is likely to have 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 call the set of the top 100 tags in the dataset by frequency T_{100} for short (shown in Figure 2). For more details about the dataset, how it was gathered, and tradeoffs in gathering it, see Heymann et al. [8].

3.1 Tradeoffs

Based on the Stanford Tag Crawl dataset, we wanted to construct a dataset approximating R_p and R_n for our prediction experiments. However, we only know R_a. In previous work, Heymann et al. [8] suggest that if $(t_i, o_k) \in \pi_{(t,o)}(R_a)$ then $(t_i, o_k) \in R_p$. In other words, annotated tags tend to be accurate. However, the reverse is not true. The case where $(t_i, o_k) \notin \pi_{(t,o)}(R_a)$ and $(t_i, o_k) \in R_p$ occurs sufficiently often that measures of precision, recall, and accuracy can be heavily skewed. In early experiments on a naively created dataset, we found that as many as $\frac{3}{4}$ of false positives were erroneous according to manual reviews we conducted. Our classifiers had accurately predicted for a given (t_i, o_k) pair that $(t_i, o_k) \in R_p$, but $(t_i, o_k) \notin \pi_{(t,o)}(R_a)$.

When comparing systems, it is reasonable to use a partially labeled dataset, because the true relative ranking of the systems is likely to be preserved. Pooling [11], for example, makes this assumption. However, for this work, we wanted to give absolute numbers for how accurately tags can be predicted, rather than comparing systems.

We decided to filter our dataset by looking at the total number of posts for a given URL:

$$\text{postcount}(o_k) = |\pi_u(\sigma_{o_k}(R_a))|$$

As $\text{postcount}(o_k)$ increases, we expect the probability for any given t_i that $(t_i, o_k) \notin \pi_{(t,o)}(R_a)$ and $(t_i, o_k) \in R_p$ to decrease.[1] We chose a cutoff of 100, which leads us to ap-

[1]Using $\text{postcount}(o_k)$ for filtering relies to a certain extent on evaluating popular tags. While we do not examine it here, for lower frequency tags we suggest $\text{vocabcount}(t_i, o_k) = \sum_{u_j \in \pi_u(\sigma_{o_k}(R_a))} min(|\sigma_{(t_i,u_j)}(R_a)|, 1)$

#	Tag	#	Tag	#	Tag
4225	reference	3469	technology	3012	web
3794	toread	3366	tools	2996	web2.0
3788	resources	3365	internet	2879	online
3677	cool	3205	computer	2759	free
3593	work	3016	blog	2661	software

#	Tag	#	Tag	#	Tag
396	politics	328	mp3	226	fashion
396	mobile	326	health	216	rails
351	game	310	environment	135	food
343	jobs	266	finance	74	recipes
341	wordpress	233	ruby	5	fic

Table 1: The top 15 tags account for more than $\frac{1}{3}$ of top 100 tags added to URLs after the 100th bookmark. Most are relatively ambiguous and personal. The bottom 15 tags account for very few of the top 100 tags added to URLs after the 100th bookmark. Most are relatively unambiguous and impersonal.

proximate R_p and R_n as:

$$\begin{aligned}(t_i, o_k) \in R_p \quad &\text{iff} \quad 100 \leq \text{postcount}(o_k) < 3000 \\ &\text{and} \quad |\pi_u(\sigma_{t_i,o_k}(R_a))| \geq \frac{\text{postcount}(o_k)}{100} \\ (t_i, o_k) \in R_n \quad &\text{iff} \quad 100 \leq \text{postcount}(o_k) < 3000 \\ &\text{and} \quad \sigma_{t_i,o_k}(R_a) = \emptyset\end{aligned}$$

This results in a filtered set of $|\pi_o(R_p \cup R_n)| \approx 62,000$ URLs and their corresponding tags.

Our reasoning for the 100 post minimum is based on the rate at which new unique tags from the top 100 tags, T_{100}, are added to a URL. Figure 1 shows the average number of new tags $t_i \in T_{100}$ that are added to a URL by the nth post. This information is computed over all URLs which occur at least 200 times in our dataset. On average, the first person to post a URL adds one tag from T_{100}, the second adds 0.6 of a tag from T_{100}, and so on. By the 100th post, the probability of a post adding a new tag from T_{100} is less than 5% and remains relatively flat. Furthermore, the top tags which are added later tend to be much more ambiguous or personal. Table 1 shows the fifteen tags which most and least commonly get added after the 100th post. Tags like "mp3" and "food" are relatively clear in meaning, whereas tags like "internet" and "toread" are much more ambiguous and personal. While we cannot completely eliminate the possibility of erroneous tuples in R_p and R_n, our approach is most accurate for unambiguous or impersonal tags and does not require creating a gold standard based on human annotation. Such a gold standard would be especially difficult to create for subjective tags like "cool" or "toread."

4. TAG PREDICTION

In this section, we look at the predictability of tags given two broad types of data. In Section 4.1 we look at the predictability of the tags in T_{100} given information we have about the web pages in our dataset. We look at page text, anchor text, and surrounding hosts to try to determine whether particular tags apply to objects in our dataset. This task is specific to social bookmarking systems because the data we use for prediction is specific to web pages. However, the predictability of tags for web pages may also be important for web search, which may want to determine if tags provide information above and beyond page text, anchor text, and surrounding hosts, and to vertical (web) search, which may want to categorize parts of the web by tags. Heymann et al. [8] provides some initial answers to these questions, but does not address predictability directly, nor does it look specifically at anchor text. "Predictability" is approximated by the predictive power of a support vector machine. While classifiers differ, we believe our results enable qualitative conclusions about the machine predictability of tags for state of the art text classifiers.

In Section 4.2 we look at the predictability of tags based on other tags already annotating an object. This task has many potential applications within tagging systems, as discussed in Section 1. Unlike the task in Section 4.1, our work in Section 4.2 is applicable to tagging systems in general (including video, photo and other tagging systems) rather than solely social bookmarking systems because it does not rely on any particular type of object (e.g., web pages).

4.1 Using Page Information

We chose to evaluate prediction accuracy using page information on the top 100 tags in our dataset (i.e., T_{100}). These tags collectively represent $2,145,593$ of $9,414,275$ triples, meaning they make up about 22.7% of the user contributed tags in the full Stanford Tag Crawl dataset. The dataset contains crawled page text and additional information for about $60,000$ of the URLs in $\pi_o(R_p) \cup \pi_o(R_n)$ (about 95%).

We treated the prediction of each tag $t_i \in T_{100}$ as a binary classification task. For each tag $t_i \in T_{100}$, our positive examples were all $o_k \in \pi_o(\sigma_{t_i}(R_p))$ and our negative examples were all $o_k \in \pi_o(\sigma_{t_i}(R_n))$. For each task, we defined two different divisions of the data into train/test splits. In the first division, which we call Full/Full, we randomly select $\frac{11}{16}$ of the positive examples and $\frac{11}{16}$ of the negative examples to be our training set. The other $\frac{5}{16}$ of each is our test set. For each Full/Full task, the number of training, test, positive, and negative examples varies depending on the tag. However, usually the training set is between $30,000$ and $35,000$ examples and the test set is about $15,000$ examples. The proportion of positive examples can vary between 1% and 60% with a median of 14% and a mean of 9%. In the second division, which we call 200/200, we randomly select 200 positive and 200 negative examples for our training set, and the same for our test set.

How well we do on the Full/Full split implies how well we can predict tags on the naturally occurring distribution of tagged pages. (We call it Full/Full because the union of positive and negative examples is the full set of URLs in R_p and R_n.) However, we can get high accuracy (if not high precision) on Full/Full by biasing towards guessing negative examples for rare tags. For example, because "recipes" only naturally occurs on 1.2% of pages, we could achieve 98.8% accuracy by predicting all negative on the "recipes" binary classification task. One solution to this problem is to change metrics to precision-recall break even point (PRBEP) or F1 (we report the former later). However, these measures are still highly impacted by the proportion of positive examples. We provide 200/200 as an imperfect indication of how predictable a tag is due to its "information content" rather than the distribution of examples in the system.

which should behave similarly to postcount(o_k) but relies on users' vocabularies rather than raw number of posts.

Each example represented one URL and had one of three different feature representations depending on whether we were predicting tags based on page text, anchor text, or surrounding hosts. Page text means all text present at the URL. Anchor text means all text within fifteen words of inlinks to the URL (similar to Haveliwala et al. [7]). Surrounding hosts means the sites linked to and from the URL, as well as the site of the URL itself. For both page text and anchor text, our feature representation was a bag of words. We tokenized pages and anchor text using the Penn TreeBank Tokenizer, dropped infrequent tokens (less frequent than the top 10 million tokens) and then converted tokens to token ids. For anchor text tasks, we only used URLs as examples which had at least 100 inlinks.[2] The value of each feature was the number of times the token occurred. For surrounding hosts, we constructed six types of features. These features were: the hosts of backlinks, the domains of backlinks, the host of the URL of the example, the domain of the URL of the example, the hosts of the forward links, and the domains of the forward links. The value of each feature was one if the domain or host in question was a backlink/forwardlink/current domain/host and zero if not.

We chose to evaluate page text, anchor text, and host structure rather than just combining all text of pages linked to or from the URL of each example because Yang et al. [20] state that including all surrounding text may reduce accuracy. For all representations (page text, anchor text, and surrounding hosts), we engineered our features by applying Term Frequency Inverse Document Frequency (TFIDF), normalizing to unit length, and then feature selected down to the top 1000 features by mutual information. We chose mutual information due to discussion in Yang and Pedersen [19]. In previous experiments, we found that the impact of more features was negligible, and reducing the feature space helped simplify and speed up the training process.[3]

For our experiments, we used support vector machines for classification. Specifically we used Thorsten Joachims' SVMlight package with a linear kernel and the default regularization parameter (see [10]) and his SVMperf package with a linear kernel and regularization parameters of 4 and 150 (see [9]). With SVMlight, we trained to minimize average error, with SVMperf, we trained to minimize PRBEP.

Given that we had 100 tags, 2 splits (200/200 and Full/Full), and 3 feature types for examples (page text, anchor text, and surrounding hosts), we conducted 600 binary classification tasks total. Assuming only a few evaluation metrics for each binary classification task, we could have thousands of numbers to report. Instead, in the rest of this section, we ask several questions intended to give an idea of the highlights of our analysis. Apart from the questions answered below, Figure 2 gives a quick at-a-glance view of which tags are more or less predictable in T_{100} ranked by the sum of PRBEP (Full/Full), Prec@10% (Full/Full) and Accuracy (200/200).[4] See discussion below for description of each metric. In the analysis below, when we give the mean of the values of tags, we mean the macro-averaged value.

cool, online, resources, community, work, culture, portfolio, social, technology, history, advertising, writing, architecture, flash, inspiration, humor, search, funny, tools, fun, internet, home, media, free, illustration, fashion, library, research, ajax, marketing, books, computer, environment, firefox, art, jobs, productivity, freeware, business, download, education, news, web2.0, language, tips, wiki, wordpress, graphics, mobile, video, google, php, article, blogs, mp3, travel, security, science, shopping, hardware, photography, games, reference, tutorials, toread, audio, photos, movies, javascript, tv, maps, blog, mac, howto, game, health, photo, design, music, opensource, osx, politics, photoshop, java, web, windows, finance, tutorial, webdesign, css, software, apple, development, food, linux, ruby, programming, rails, recipes

Figure 2: Tags in T_{100} in increasing order of predictability from left to right. "cool" is the least predictable tag, "recipes" is the most predictable tag.

What precision can we get at the PRBEP?

For applications like vertical search (or search enhanced by topics), one natural question is what our precision-recall curve looks like at reasonably high recall. PRBEP gives a good single number measurement of how we can tradeoff precision for recall. For the Full/Full split, we calculated the PRBEP for each of the 600 binary classification tasks. On average, the PRBEP for page text was about 60%, for anchor text was about 58%, and for surrounding hosts was about 51% with a standard deviation of between 8% and 10%. This suggests that on realistic data, we can get about $\frac{2}{3}$ of the URLs labeled with a particular tag with about $\frac{1}{3}$ erroneous URLs in our resulting set. This is pretty good—we are doing much better than chance given that a majority of tags in T_{100} occur on less than 15% of documents.

What precision can we get with low recall?

For applications like bootstrapping or single tag queries (see Section 1), we may care less about overall recall (because the web is huge), but we may want high precision. We used the Full/Full split to look at this question. For each binary classification task, we calculated the precision at 10% recall (e.g., Prec@10%). With all of our feature types (page text, anchor text, and surrounding hosts), we were able to get a mean Prec@10% value of over 90%. The page text Prec@10% was slightly higher, at 92.5%, and all feature types had a standard deviation of between 7% and 9%. This suggests that whatever our feature representation, if we have many more examples than we need for our system, we can get high precision by reducing the recall. Furthermore, it suggests that there are some examples of most tags that our classifiers are much more certain about, rather than a relatively uniform distribution of certainty.

Which page information is best for predicting tags?

According to all evaluation metrics, we found a strict ordering among our feature types. Page text was strictly more informative than anchor text which was strictly more informative than surrounding hosts. For example, for PRBEP,

[2]We found that the difference between 10 and 100 inlinks as the cutoff was negligible. More data about a particular URL improves classification accuracy for that URL, but more URLs improves classification accuracy for in general.

[3]Gabrilovich and Markovitch [5] actually find that aggressive feature selection is necessary for SVM to be competitive with decision trees for certain types of hypertext data.

[4]Two tags are missing, "system:imported" (a system generated tag) and "fic" (which is common in the full dataset but uncommon for top URLs and was removed as an outlier).

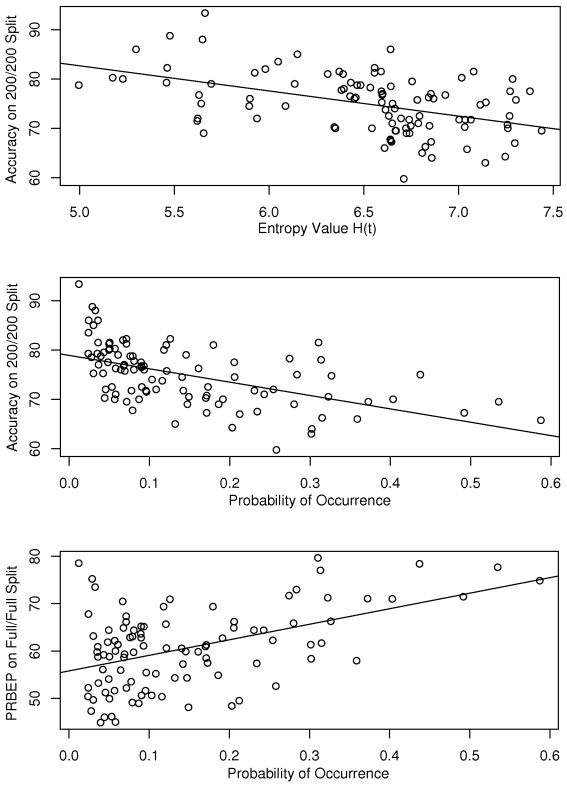

Figure 3: When the rarity of a tag is controlled in 200/200, entropy and occurrence rate are negatively correlated with predicability (first/second graphs). However, in Full/Full, additional examples are more important than the vagueness of a tag, and more common tags are more predictable (third graph).

the ordering is $(60, 58, 51)$, for Prec@10% it is $(92.5, 90, 90)$, for accuracy on the 200/200 split, it is $(75, 73, 66)$. Usually, page text was incrementally better than anchor text, while both were much better than surrounding hosts. This may have been due to our representation or usage of our surrounding hosts, or it could simply be that text is a particularly strong predictor of the topic of a page.

Is anchor text particularly predictive of tags?

One common complaint about tags is that they should be highly predictable based on anchor text, because both serve as commentary on a particular URL. While both page text and anchor text are predictive of tags, we did not find anchor text to be more predictive on average than page text for any of our split/evaluation metric combinations.

What makes a tag predictable?

A more general question than those above is what makes a tag predictable. Predictability may give clues as to the "information content" of a tag, but it may also be practically useful for tasks like deciding which tags to suggest to users. In order to try to quantify this, we defined an entropy measure to try to mirror the "generality" of a tag. Specifically, we call the distribution of tag co-occurrence events with a given tag t_i, $P(T|t_i)$. Given this number, we define the entropy of a tag t_i to be:

$$H(t_i) = -\sum_{t_j \in T, t_j \neq t_i} P(t_j|t_i) \log P(t_j|t_i)$$

For example, if the tag t_{car} co-occurs with t_{auto} 3 times, with $t_{vehicle}$ 1 time, and with $t_{automobile}$ 1 time, we would say its entropy was equal to:

$$H(t_{car}) = -\frac{3}{5}\log\frac{3}{5} - \frac{1}{5}\log\frac{1}{5} - \frac{1}{5}\log\frac{1}{5} \approx 1.37$$

Because the relative rarity of a tag heavily impacts its predictability, we used the 200/200 split to try to evaluate predictability of tags in the abstract. For this split, we found a significant correlation between our entropy measure $H(t_i)$ and accuracy of a classifier on 200/200 (see Figure 3 for the plot of accuracy versus entropy and accuracy versus occurrence rate which follows). For page text, we had a Pearson product-moment correlation coefficient of $r = -0.46$, for anchor text $r = -0.51$, and for surrounding hosts $r = -0.54$. All p-values were less than 10^{-5}.[5] However, for the same split, we also found that the popularity of a tag was highly negatively correlated with our accuracy. Specifically, for page text, we had $r = -0.53$, for anchor text $r = -0.51$, and for domains $r = -0.27$. In other words, the popularity of a tag seems to be as good a proxy for "generality" as a more complex entropy measure. The two are not exclusive—a linear model fit to accuracy based on both popularity and entropy does better than a model trained on either one alone.

For the Full/Full split, we found that the commonality of a tag (and hence the commonality of positive examples) was highly positively correlated with high PRBEP. However, perhaps because the recall was relatively low, we found no correlation between the commonality of a tag and our performance on Prec@10% (though we did find some low but significant correlation between PRBEP and Prec@10%). The entropy measure was uncorrelated with PRBEP or Prec@10% for the Full/Full split.

4.2 Using Tags

Between about 30 and 50 percent of URLs posted to del.icio.us have only been bookmarked once or twice. Given that the average bookmark has about 2.5 tags, the odds that a query for a particular tag will return a bookmark only posted once or twice are low. In other words, our recall for single tag queries is heavily limited by the high number of rare URLs with few tags. For example, a user labeling a new software tool for Apple's Mac OS X operating system might annotate it with "software," "tool," and "osx." A second user looking for this content with the single tag query (or feed) "mac" would miss this content, even though a human might easily realize that "osx" implies "mac." The question in this section is given a small number of tags, how much can we expand this set of tags in a high precision manner? The better we do at this task, the less likely we are to have situations like the "osx"/"mac" case because we will be able to expand tags like "osx" into implied tags like "mac."

A natural approach to this problem is market-basket data mining. In the market-basket model, there are a large set of items and a large set of baskets each of which contains a small set of items. The goal is to find correlations between

[5]Though we do not quote them here, we also computed Kendall's τ and Spearman's ρ values which gave similarly strong p-values.

Int.	Conf.	Supp.	Rule
0.59	0.994	634	graphic-design → design
0.69	0.992	644	oop → programming
0.56	0.992	654	macsoftware → software
0.89	0.990	605	photographer → photography
0.44	0.990	1780	webstandards → web
0.44	0.990	786	w3c → web
0.58	0.989	2144	designer → design
0.85	0.987	669	windowsxp → windows
0.44	0.987	1891	dhtml → web
0.85	0.986	872	debian → linux
0.58	0.986	1092	illustrator → design
0.56	0.986	707	sourceforge → software
0.85	0.985	1146	gnu/linux → linux
0.61	0.985	539	bloggers → blog
0.58	0.985	597	ilustracion → design
0.44	0.985	1794	web-development → web
0.44	0.985	3366	xhtml → web
0.68	0.984	730	disenoweb → webdesign
0.87	0.983	648	macsoftware → mac

Table 2: Association Rules: A selection of the top 30 tag pair association rules. All of the top 30 rules appear to be valid, these rules are representative.

sets of items in the baskets. Market-basket data mining produces association rules of the form $X \rightarrow Y$. Association rules commonly have three values associated with them:

Support The number of baskets containing both X and Y.

Confidence $P(Y|X)$. (How likely is Y given X?)

Interest $P(Y|X) - P(Y)$, alternatively $\frac{P(Y|X)}{P(Y)}$. (How much more common is X&Y than expected by chance?)

Given a minimum support, Agrawal et al. [1] provide an algorithm for computing association rules from a dataset.

In our case, the baskets are URLs, and the items are tags. Specifically, for each $o_k \in \pi_o(R_p)$, we construct a basket $\pi_t(\sigma_{o_k}(R_p))$. We constructed three sets of rules: rules with support > 500 and length 2, rules with support > 1000 and length 3, and rules with support > 2000 of any length. We merged these three rule sets into a single association rule set for our experiments.

Found Association Rules

We found a surprising number of high quality association rules in our data. Table 2 shows some of the top association rules of length two. The rules capture a number of different relationships between tags in the data. Some rules correspond to a "type-of" style of relationship, for example, "graphic-design" is a type of "design." Others correspond to different word forms, for example, "photographer" and "photography." Some association rules correspond to translations of a tag, for example, "disenoweb" is Spanish for "webdesign." Some of the relationships are surprisingly deep, for example, the "w3c" is a consortium that develops "web" standards. Arguably, one might suggest that if both $t_i \rightarrow t_j$ and $t_j \rightarrow t_i$ with high confidence and high interest, t_i and t_j are probably synonymous.

Depending on computational resources, numbers of association rules in the millions or billions can be generated with reasonable support. However, in practice, the most intuitive rules seem to be rules of length four or less. In order to give an idea of the rules in general, rather than picking the top rules, we give a random sampling of the top 8000 rules of length three or less. This information is shown in Table 3. There is sometimes redundancy in longer rules, for example, one might suggest that rather than "webdesign & websites → web" we should instead have "webdesign → web" and "websites → web". This is a minor issue however, and it is relatively rare for a rule with high confidence to be outright incorrect. Furthermore, given their ease of interpretation, it would not be unreasonable to have human moderators look over low length, high support and high confidence rules.

Int.	Conf.	Supp.	Rule
0.81	0.989	1097	open_source & source → opensource
0.55	0.979	1003	downloads & os → software
0.42	0.967	1686	free & webservice → web
0.73	0.964	1134	accessibility & css → webdev
0.84	0.952	1305	app & osx → mac
0.40	0.950	2162	webdesign & websites → web
0.47	0.947	2269	technology & webtools → tools
0.40	0.945	1662	php & resources → web
0.63	0.937	2754	html & tips → webdesign
0.50	0.934	1914	xp → software
0.45	0.928	1332	freeware & system → tools
0.62	0.919	1513	cool & socialsoftware → web2.0
0.61	0.915	1165	business & css → webdesign
0.61	0.912	2231	tips & webdevelopment → development
0.35	0.900	6337	toread & web2.0 → web
0.69	0.897	1010	fotografia & inspiration → art
0.33	0.895	1723	help & useful → reference

Table 3: Association Rules: A random sample of association rules of length ≤ 3 and support > 1000.

Tag Application Simulation

For our evaluation, we simulate rare URLs with few bookmarks. We separated $\pi_o(R_p)$ into a training set of about $50,000$ URLs and a test set of about $10,000$ URLs. We generated our three sets of association rules based only on baskets from the training set. We then sampled n bookmarks from each of the the URLs in the test set, pretending these were the only bookmarks available. Given this set of sampled bookmarks, we attempted to apply association rules in decreasing order of confidence to expand the set of known tags. We stopped applying association rules once we had reached a particular minimum confidence c.

For example, suppose we have a URL which has a recipe for oven-cooked pizza bagels with three bookmarks corresponding to three sets of tags:

$$\{(\text{food}, \text{recipe}), (\text{food}, \text{recipes}), (\text{pizza}, \text{bagels})\}$$

For $n = 1$, we might sample the bookmark *(pizza, bagels)*. Assuming we had two association rules:

- pizza → food (confidence = 0.9)
- bagels → bagel (confidence = 0.8)

We would first apply the confidence 0.9 rule and then the confidence 0.8 rule. Applying both rules (i.e., two applications), would result in *(pizza, bagels, food, bagel)*.

Number and Precision of Tag Expansions

We ran a simulation as described above for each number of sampled bookmarks $n \in \{1, 2, 3, 5\}$ and for each minimum confidence $c \in \{0.5, 0.75, 0.9\}$. Our results are shown in Table 4. Each row represents one setting of n and c. We asked two initial questions: "How many tags were added?" and

# B-marks	Min. Conf.	# Tag Expansions						Exp. Precision		Mean Recall (T_{100})		New Precision (T_{100})	
		0	1	2	3	4	5+	Est.	Actual	Orig.	Expd.	Mean	Median
1	0.50	2096	100	153	435	486	7667	0.650	0.633	0.099	0.271	0.629	0.677
1	0.75	4015	1717	1422	1263	866	1654	0.844	0.854	0.099	0.153	0.929	0.963
1	0.90	7898	1845	709	291	116	78	0.941	0.954	0.100	0.113	0.993	1.000
2	0.50	545	78	115	283	300	9616	0.652	0.590	0.160	0.386	0.585	0.626
2	0.75	2067	1630	1582	1664	1145	2849	0.844	0.811	0.164	0.237	0.909	0.949
2	0.90	6520	2491	1113	473	208	132	0.942	0.931	0.161	0.180	0.989	1.000
3	0.50	216	61	62	172	164	10262	0.653	0.559	0.205	0.451	0.550	0.577
3	0.75	1397	1415	1545	1558	1363	3659	0.844	0.779	0.207	0.289	0.900	0.942
3	0.90	5913	2746	1265	596	249	168	0.943	0.917	0.204	0.226	0.988	1.000
5	0.50	71	31	29	101	80	10625	0.654	0.509	0.265	0.524	0.497	0.496
5	0.75	810	1070	1360	1507	1485	4705	0.842	0.732	0.268	0.358	0.881	0.931
5	0.90	5145	3065	1427	692	366	242	0.943	0.873	0.271	0.294	0.983	0.996

Table 4: Association Rules: Tradeoffs between number of original bookmarks, minimum confidence, resulting tag expansions, recall, and precision.

"How accurate were our applications of tags?" The column labeled "# Tag Expansions" shows, for each simulation, the number of URLs to which we were able to add 0, 1, 2, 3, 4 or 5+ tags. The column labeled "Actual Precision" shows the percentage of tag applications which were correct (given the other information we had about each URL in R_p). For each simulation, we also compute our estimate ("Estimated Precision") of what our precision should have been based on the confidence values of applied rules. Our estimate is the average of the confidence of all applied rules. For example, in our oven-cooked pizza bagel example above (assuming the URL was the only URL in our simulation), we would have an actual precision of 0.5 because "food" is a tag which appears in other bookmarks annotating the URL, whereas "bagel" is not. Our estimate of our precision would be $\frac{0.9+0.8}{2} = 0.85$. We would also increment the "2" column of "# Tag Expansions" because the URL was expanded twice.

Thus, the first ten columns of the first row of Table 4 say that we ran a simulation with $n = 1$, $c = 0.5$ and $10,937$ URLs. In $2,096$ cases, we were not able to add any tags (anecdotally, this usually happens when a bookmark only has one tag). In $7,667$ cases, we were able to add five or more tags. Our estimate of our precision was 0.650 while our actual precision was a little lower, 0.633.

The results in the first ten columns of Table 4 show that with only a single bookmark, we can expand anywhere from 10 to 80 percent of our URLs by at least one tag depending on our desired precision. With larger numbers of bookmarks, we can do better, though the most pertinent tags for a URL are applied quickly. Also, as the number of bookmarks increases, the difference between estimated and actual precision increases. This means that as a URL receives more and more annotations, we become increasingly unsure of the effectiveness of association rules for unapplied tags.

How Useful are Predicted Tags?

As we argued in Section 1, predicted tags can be used by a system in many ways. Here we briefly explore one such use: increasing recall for single tag queries. For instance, if the user searches for "food," the system can return objects annotated with "food" as well as objects which we predict "food" annotates. Using term co-occurrence to expand query results is a well known IR technique; here we want to know how well it works for tags.

For evaluation, we consider each tag $t_i \in T_{100}$ to be a query q_{t_i}. For each query q_{t_i}, the result set s contains the URLs annotated with the tag, and the result set s' contains the URLs annotated with the tag plus URLs which we predict are annotated with the tag using association rules. We then compare the recall and precision achieved by s and s'. For example, suppose five objects are positively described by "food." In our simulation suppose only two of the objects are known to have "food." Suppose that we correctly predict that one of the remaining three objects is labeled "food" (perhaps using our "bagels" $\rightarrow$ "food" rule above), and we incorrectly predict that two other objects are labeled "food." Without expansion, query q retrieves s which has two known bagel objects, so recall is 2/5. With expansion, s' returns three additional objects, one of which was correct, for a recall of $\frac{2+1}{5}$ and a precision of $\frac{2+1}{2+3}$.

The rightmost 4 columns of Table 4 show the results for the experiment. For each simulation (row), we give (a) the mean recall before expansion (macro average over all tags in T_{100}); (b) the mean recall after expansion; and (c) the mean and median of the precision after expansion. (Note that without expansion precision is always 1.) For instance, if we sample one bookmark per URL ($n = 1$) and use 50% confidence ($c = 0.5$), we see that tag expansion improves mean recall from 0.099 to 0.271, a factor of 3 improvement! Of course, our average precision drops from 1 to 0.629. For both the one and two tag cases (i.e., $n = 1$ and $n = 2$) with confidence $c = 0.75$ we can increase recall by 50% while keeping precision above 90%.

5. RELATED WORK

Previous work has looked at the nature of tags chosen by users [6, 16]. We do not know of any work explicitly looking at how to construct a reasonable dataset for prediction in tagging systems as we do in Section 3. While our hypertext classification task in Section 4.1 is inspired by a long line of work, usefully surveyed by Yang et al. [20], we believe the application to tags is new. Chakrabarti et al. [3] suggest a different way to use local link information for classification that might prove more effective than our domain features, however, we do not evaluate this possibility here. Our use of an entropy measure for tagging systems is inspired by Chi and Mytkowicz [4]. Other work has looked at tag suggestion, usually from a collaborative filtering and UI perspective, for example with URLs [18] and blog posts [13, 17].

Our work in Section 4.2 is similar to work by Schmitz et al. [14]. However, Schmitz et al. is primarily concerned with theoretical properties of mining association rules in tripartite graphs. Schwarzkopf et al. [15] extend Schmitz's association rules work to build full ontologies. However, neither Schmitz et al. nor Schwarzkopf et al. appear to evaluate the quality of the rules themselves aside from generating ontologies. Lastly, there is also much previous work in IR studying query expansion and relevance feedback trying to address similar questions of cross-language and cross-vocabulary queries (see for example a general reference such as Manning et al. [12]). However, we believe that association rules may be the most natural approach to these problems in tagging systems due to user interface issues (for example, feeds, browsing).

6. DISCUSSION

We are in the midst of a large scale experiment to determine what kind of metadata scales to millions of users, what kind gives the most information for IR, and how to maximally leverage that metadata for IR. In this paper, we looked at tags, which make several tradeoffs to try to scale the web. Our tag prediction results suggest three insights.

First, many tags on the web do not contribute substantial additional information beyond page text, anchor text, and surrounding hosts. All three types of data can be quite predictive of different tags in our dataset, and if we only want a small recall (e.g., 10%) we can have a precision above 90%. The predictability of social bookmarking tags may influence web search (by suggesting ways to use tagging information or whether to use it at all), as well as to system designers who might bootstrap tagging systems with initial quality data (by making it possible to predict such initial data).

Second, the predictability of a tag when our classifiers are given balanced training data is negatively correlated with its occurrence rate and with its entropy. More popular tags are harder to predict and higher entropy tags are harder to predict. When considering tags in their natural (skewed) distributions, data sparsity issues tend to dominate, so each further example of a tag improves classifier performance. To the extent to which predictability is correlated with the "generality" of a tag, these measures may serve as building blocks for tagging system designers to produce new features that rely upon understanding the specificity of tags (for example, system suggestion and tag browsing). Both of our measures of tag predictability are object type independent. This suggests that they may be applicable to tagging systems based on photos or video rather than only social bookmarking systems.

Third, association rules can increase recall on the single tag queries and feeds which are common in tagging systems today. This suggests that they may serve as a way to link disparate vocabularies among users. We found association rules linking languages, super/subconcepts, and other relationships. These rules may also indicate synonymy and polysemy, two issues that have plagued tagging systems since Golder and Huberman's seminal work [6].

7. ACKNOWLEDGEMENTS

The authors acknowledge Varun Ganapathi's help in experimental design. Heymann is supported by an NSF GRFP Fellowship. Ramage is supported by an NDSEG Fellowship.

8. REFERENCES

[1] R. Agrawal, T. Imieliński, and A. Swami. Mining Association Rules Between Sets of Items in Large Databases. *SIGMOD Record*, 22(2), 1993.

[2] M. Aurnhammer, P. Hanappe, and L. Steels. Integrating Collaborative Tagging and Emergent Semantics for Image Retrieval. *Collaborative Web Tagging Workshop (WWW'06)*.

[3] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced Hypertext Categorization Using Hyperlinks. *SIGMOD'98*.

[4] E. Chi and T. Mytkowicz. Understanding the Efficiency of Social Tagging Systems using Information Theory. *HT'08*.

[5] E. Gabrilovich and S. Markovitch. Text Categorization with Many Redundant Features: Using Aggressive Feature Selection to Make SVMs Competitive with C4.5. *ICML'04*.

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

[7] T. Haveliwala, A. Gionis, D. Klein, and P. Indyk. Evaluating Strategies for Similarity Search on the Web. *WWW'02*.

[8] P. Heymann, G. Koutrika, and H. Garcia-Molina. Can Social Bookmarking Improve Web Search. *WSDM'08*.

[9] T. Joachims. A Support Vector Method for Multivariate Performance Measures. *ICML'05*.

[10] T. Joachims. Making Large-scale Support Vector Machine Learning Practical. *Advances in Kernel Methods: Support Vector Learning*, 1999.

[11] K. Jones and C. van Rijsbergen. Information Retrieval Test Collections. *Journal of Documentation*, 32(1):59–75, 1976.

[12] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[13] G. Mishne. AutoTag: a collaborative approach to automated tag assignment for weblog posts. *WWW'06*.

[14] C. Schmitz, A. Hotho, R. Jaschke, and G. Stumme. Mining Association Rules in Folksonomies. *IFCS'06*.

[15] E. Schwarzkopf, D. Heckmann, D. Dengler, and A. Kroner. Mining the Structure of Tag Spaces for User Modeling. *Workshop on Data Mining for User Modeling (ICUM'07)*.

[16] S. Sen, S. K. Lam, A. M. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, F. M. Harper, and J. Riedl. tagging, communities, vocabulary, evolution. *CSCW'06*.

[17] S. Sood, K. Hammond, S. Owsley, and L. Birnbaum. TagAssist: Automatic Tag Suggestion for Blog Posts. *ICWSM'07*.

[18] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the Semantic Web: Collaborative Tag Suggestions. *Collaborative Web Tagging Workshop (WWW'06)*.

[19] Y. Yang and J. O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. *ICML'97*.

[20] Y. Yang, S. Slattery, and R. Ghani. A Study of Approaches to Hypertext Categorization. *Journal of Intelligent Information Systems*, 18(2-3), 2002.