Towards the Semantic Web: Collaborative Tag Suggestions

Zhichen Xu, Yun Fu, Jianchang Mao, and Difu Su
Yahoo! Inc
2821 Mission College Blvd., Santa Clara, CA 95054
{zhichen, yfu, jmao, difu}@yahoo-inc.com

ABSTRACT

Content organization over the Internet went through several interesting phases of evolution: from structured directories to unstructured Web search engines and more recently, to tagging as a way for aggregating information, a step towards the semantic web vision. Tagging allows ranking and data organization to directly utilize inputs from end users, enabling machine processing of Web content. Since tags are created by individual users in a free form, one important problem facing tagging is to identify most appropriate tags, while eliminating noise and spam. For this purpose, we define a set of general criteria for a good tagging system. These criteria include high coverage of multiple facets to ensure good recall, least effort to reduce the cost involved in browsing, and high popularity to ensure tag quality. We propose a collaborative tag suggestion algorithm using these criteria to spot high-quality tags. The proposed algorithm employs a goodness measure for tags derived from collective user authorities to combat spam. The goodness measure is iteratively adjusted by a reward-penalty algorithm, which also incorporates other sources of tags, e.g., content-based auto-generated tags. Our experiments based on My Web 2.0 show that the algorithm is effective.

Keywords

Classification, tagging, information retrieval, collaborative filtering, Web 2.0.

1. INTRODUCTION

Effectively organizing information over the World Wide Web has been a challenging problem since the beginning. In the early days of the Internet, portal services organized Web content into hierarchical directories, assuming that the Web can be organized by strict structures of topics. However, the manually supervised directories have been gradually predominated by crawler-based search engines for at least two reasons: data explosion and the unstructured nature of Web content. While search engines work well for users to access Web information by issuing *ad hoc* queries, they use very limited semantic information of the Web content by parsing content and exploiting the hyperlink structure established by Web masters. The pull model used by search engines makes it hard to discover new and dynamic content. According to Brightplanet, the deep Web can be 500 times larger than the surface Web. In addition, personalization and spam detection require human inputs. Furthermore, it is difficult for people to share massive unstructured Web pages among each other or recover them later. A push model that directly takes inputs from users solves these problems. Tagging is a process by which users assign labels (in the form of keywords) to Web objects with a purpose to share, discover and recover them. Discovery enables users to find new content of their interest shared by other users. Recovery enables a user to recall content that was discovered before. Further, tagging allows ranking and data organization to utilize metadata from individual users directly. It brings some benefits of semantic Web into the current HTML dominated Web.

We are witnessing an increasing number of tagging services on the web, such as Flickr [11], Delicious [10], My Web 2.0 [12], Rawsugar [14], and Shadows [15]. Flickr enables users to tag photos and share them with others. Delicious users can tag URLs and share their bookmarks with the public. My Web 2.0 provides a Web-scale social search engine to enable users to find, use, share, and expand human knowledge. It allows users to save and tag Web pages so that they can easily browse and search for the content again. It also enables users to share Web pages within a personalized community or to the public by setting access privileges. Further, My Web 2.0 provides scoped search within user's trusted social networks, e.g., friends or friends of friends. Consequently, the search results are personalized and spam-filtered by the trusted networks.

Tagging advocates a grass root approach to form a so-called "*folksonomy*", which is neither hierarchical nor exclusive. With tagging, a user can enter labels in a free form to tag any object; it therefore relieves users much burden of fitting objects into a universal ontology. Meanwhile, a user can use a certain tag combination to express the interest in objects tagged by other users, e.g., tags (`renewable`, `energy`) for objects tagged by both the keywords `renewable` and `energy`.

Ontology works well when the corpus is small or in a constrained domain, the objects to be categorized are stable, and the users are experts [8]. A universal ontology is difficult and expensive to construct and maintain when there involve hundreds of millions of users with diverse background. When used to organize Web objects, ontology faces two hard problems: unlike physical objects, digital content is seldom semantically pure to fit in a specific category; and it is difficult to predict the paths, through which a user would explore to discover a digital object [8]. Taking Yahoo directory as an example, a recipe book belongs to both the categories `Shopping` and `Health`,

since it is hard to predict which category an end user would perceive to be the best fit.

Tagging bridges some gap between browsing and search. Browsing enumerates all objects and finds the desirable one by exerting the *recognition* aspect of human brain, whereas search uses *association* and dives directly to the interested objects, and thus is mentally less obnoxious [9].

The benefits of tagging do not come without a cost. For instance, the number of tags in a social network multiples like rabbits [13]. The structure in traditional hierarchy disappears: Tagging relates to faceted classification, which uses clearly defined, mutually exclusive, and collectively exhaustive aspects to describe objects. For instance, a music piece can be identified by facets such as artist, album, genre, and composer. Faceted systems fail to dictate a linear order in which to experience the facets, a step crucial for guiding the users to explore this system. Since tags are created by end-users in a free form, they can be chaotic when compared with a faceted system constructed by experts. This lack of order and depth can result in a disaster, leaving the users muddled in a "hodgepodge" [13].

To remedy the shortcomings of tagging, we advocate using collaboratively filtering to automatically identify high-quality tags for users, leveraging the collective wisdom of Web users. Specifically, this paper makes the following contributions:

- We discuss the desirable properties of a good tagging system, which include: (a) *high coverage of multiple facets*, (b) *high popularity*, and (c) *least-effort*. Faceted and generic tags can facilitate the aggregation of objects entered by different users. It makes discovery and recovery of tagged content easier. Tags used by a large number of people for a given object are less likely to be spam and more likely to be used by a new user for the same object. Least-effort has two meanings: The number of objects identified by the suggested tags should be small, and the number of tags for identifying an object should be minimized as well. This enables efficient recovery of the tagged objects.
- We propose collaborative tagging techniques that suggest tags for an object based on what other users use to tag the object. This not only addresses the vocabulary divergence problem, but also relieves users the obnoxious task of having to come up with a good set of tags.
- We propose a reputation score for each user based on the quality of the tags contributed by the user.
- By introducing the notion of "virtual" users, our tag suggestion algorithm incorporates not only user-generated tags but also other sources of tags, such as tags auto-generated via content-based or context-based analysis.
- We have implemented a simplified tag suggestion scheme in My Web 2.0. Our experience shows that this simple scheme is quite effective in suggesting appropriate tags that possess the properties proposed by us for a good tagging system.

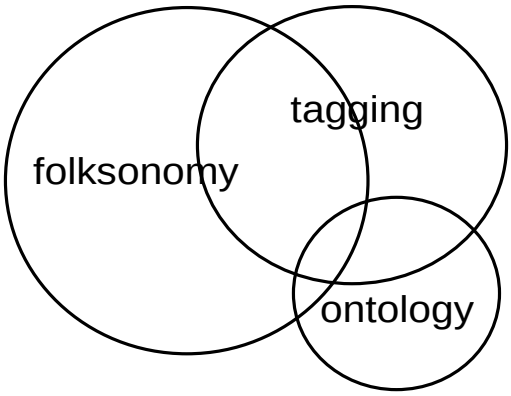

Figure 1. Tag browsing via filtering. The objects tagged by the tag "*folksonomy*" intersect with those tagged by the tags "*tagging*" and "*ontology*." Therefore, the tags "*tagging*" and "*ontology*" are related to the tag "*folksonomy*."

The rest of the paper is organized as follows: Section 2 discusses an important usage of tags for relational browsing. Section 3 describes a set of criteria for selecting high quality tags and proposes an algorithm for tag suggestion. In section 4, we illustrate our algorithm with a few examples. We conclude in Section 5.

2. RELATIONAL TAG BROWSING

Tagging is a tool to organize objects for the purposes of recovery and discovery. Unlike scientific classification, which forces a hierarchical structure on objects, tagging organizes objects in a network structure, thus making it suitable to organize Web objects, which lack a clear hierarchical structure by nature. Tagging, when combined with search technology, becomes a powerful tool to discover interesting Web objects. With the help of search technology, tagged objects can be browsed or searched for. The way tags work is analogous to filters. They are treated as logical constraints to filter the objects. Refinement of results is done through strengthening the constraints whereas generalization is done by weakening them. E.g., tag combination (`2006`, `calendar`) strengthens tag (`2006`) and tag (`calendar`).

Figure 1 illustrates how tags can be used as a filtering mechanism for browsing and searching for objects. In My Web 2.0, we explore the co-occurrence of tags to enable tag browsing through progressive refinement. When a user

selects a tag combination, the system returns the set of objects tagged with the combination. Meanwhile, it also returns the tags that relate to the selected tags, which are those co-occur with the selected tags. In Figure 1, the tags (`tagging`) and (`ontology`) relate to the tag `folksonomy`.

In the next section, we describe our collaborative tag suggestion algorithm.

3. COLLABORATIVE TAG SUGGESTION

3.1 A taxonomy of tags

Before presenting the algorithm, we first describe the categories of tags that we observe on My Web 2.0.

1. Content-based tags: Tags that describe the content of an object or the categories that the object belongs to, e.g., `Autos`, `Honda Odyssey`, `batman`, `open source`, `Lucene`, and `German Embassy`. These tags are usually specific terms and are common in My Web 2.0.
2. Context-based tags: Tags that provide the context of an object in which the object was created or saved, e.g., tags describing locations and time such as `San Francisco`, `Golden Gate Bridge`, and `2005-10-19`.
3. Attribute tags: Tags that are inherent attributes of an object but may not be able to be derived from the content directly, e.g., author of a piece of content such as `Jeremy's Blog` and `Clay Shirky`.
4. Subjective tags: Tags that express user's opinion and emotion, e.g., `funny` or `cool`.
5. Organizational tags: Tags that identify personal stuff, e.g., `my paper` or `my work`, and tags that serve as a reminder of certain tasks such as `to-read` or `to-review`. This type of tags is usually not useful for global tag aggregation with other user's tags.

Golder and Huberman have also discussed tag categorization [3].

3.2 Criteria for good tags

In a large scale tagging system like My Web 2.0, an object is usually identified by a group of tags. A specific tag is efficient to identify an object but less useful for other people to discover new objects. In contrast, a generic tag is useful for discovery but not effective to narrow down objects. Tagging an object with a good set of tags helps both discovery and recovery. We argue that a good tag combination should have the following properties.

High coverage of multiple facets. A good tag combination should include multiple facets of the tagged objects. For example, tags for a URL to a travel attraction site may include generic tags such as category (`travel`), location (`San Francisco`), time (`2005`), specific tag (`Golden Gate Bridge`), and subjective tag (`cool`). Generic tags facilitate the aggregation of the content entered by different users and thus are often used for a large number of objects. The larger the number of facets the more likely a user is able to recall the tagged content.

High popularity. If a set of tags are used by a large number of people for a particular object, these tags are less likely to be a spam. They are more likely to uniquely identify the tagged content and the more likely to be used by a new user for the given object. This is analogous to the term frequency in traditional information retrieval.

Least-effort. The number of tags for identifying an object should be minimized, and the number of objects identified by the tag combination should be small. As a result, a user can reach any tagged objects in a small number of steps via tag browsing.

Uniformity (normalization). Since there is no universal ontology, tags can diverge dramatically. Different people can use different terms for the same concept. In general, we have observed two general types of divergence: those due to syntactic variance, e.g., `blogs`, `blogging`, and `bog`; and those due to synonym, e.g., `cell-phone` and `mobile-phone`, which are different syntactic terms that refer to the same underlying concept. These kinds of divergence are a double-edged sword. On the one hand, they introduce noises to the system; on the other hand it can increase recall. The right thing to do is to allow the users to use whatever form they like but to collapse the variances to an internal canonical representation.

Exclusion of certain types of tags. For example, personally used organizational tags are less likely to be shared by different users**.** Thus, they should be excluded from public usage. Rather than ignoring these tags, My Web 2.0 includes a feature that auto-completes tags as they are being typed by matching the prefixes of the tags entered by the user before. This not only improves the usability of the system but also enables the convergence of tags.

Our criteria are based on study of tag usage by real users in My Web 2.0. Figure 2 shows the rank of a tag versus the number of URLs labeled by the tag in a log-log scale, which demonstrates a Zipf-like distribution. The figure only shows a subset of data publicly shared by users. We excluded three system introduced tags, which are automatically generated for Web objects imported from other services. Our data shows that people naturally select some popular and generic tags to label their interested Web objects. The most popular tags include *music*, *news*, *software*, *blog*, *rss*, *web*, *programming*, and *design*. These tags are convenient for users to recover and share with other users.

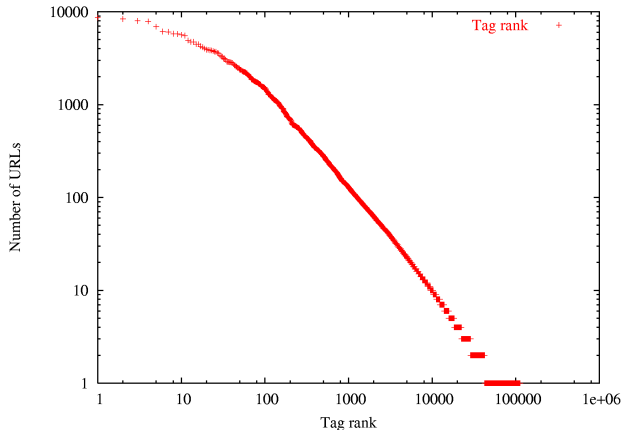

Figure 2. Tag popularity

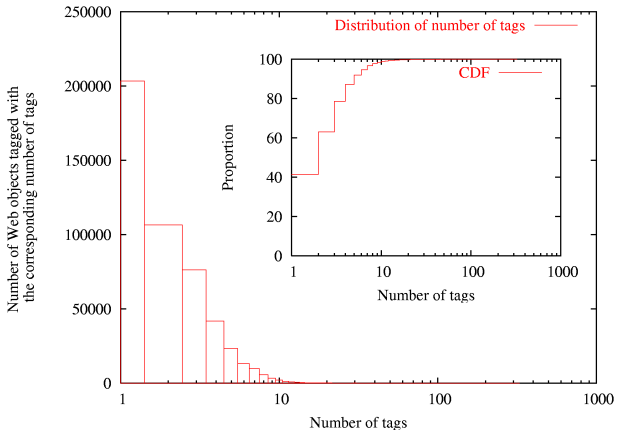

Figure 3. Distribution of the number of Web objects tagged with the corresponding number of tags

Figure 3 shows the distribution of the number of tags versus the number of Web objects tagged with the corresponding number of tags. From the figure, we can observe that *92%* Web objects are labeled with equal or less than *5* tags, *79%* Web objects with equal or less than *3* tags. The figure demonstrates that our least-effort criteria will be acceptable by most users.

3.3 Collaborative Tag Suggestions

Our tag suggestion algorithm takes the above criteria into consideration. First, it favors tags that are used by a large number of people (with good reputation). Second, it aims to minimize the overlap of concepts among the suggested tags to allow for high coverage of multiple facets. Third, it honors the high correlation among tags, e.g., if tags `ajax` and `javascript` tend to be used together by most users for a given object, they should co-occur in our suggested tags. We first introduce some basic concepts and notations before presenting our tag suggestion algorithm:

- $P_s(t_i|t_j;o)$ --- the probability that an object o is tagged with t_i given it is already tagged with t_j by the *same* user. For the given object o, one way to measure such correlation between t_i and t_j is to divide the number of people who have tagged o with both t_i and t_j by the number of people who have tagged it by t_j. Our algorithm honors such correlation when suggesting tags.

- $P_a(t_i|t_j)$ --- the probability that *any* object is tagged with t_i, given it is already tagged with t_j by *any* user. Such correlation can be measured as the number of people who have used both t_i and t_j over the number of people who have used with t_j. This probability indicates the overlap in terms of the concepts between t_i and t_j. To ensure that the suggested tags cover multiple facets, our algorithm attempts to minimize the overlap of the concepts identified by the suggested tags.

- $S(t,o)$ --- Goodness measure (score) of the tag t to an object o. We use the sum of the authority scores of all users who have assigned tag t to the object o. In a simple case where we assign uniform authority score of 1.0 for every user.

- $C(t)$ --- The coverage of tag t, defined as the number of different objects tagged by t with some dampening. In practice, the goodness measure can be enhanced by accounting for the coverage of a tag. The wider the coverage, the less specific the tag is to a given object. This is analogous to TF*IDF used in traditional information retrieval.

The basic idea of our algorithm is to iteratively select the tags with the highest additional contribution measured by $S(t,o)$ to the already selected tag set. $S(t,o)$ is initialized to the sum of the authority scores (of all users who have assigned tag t to object o) multiplied by the inverse of $C(t)$. In the remainder of the paper, we ignore $C(t)$ for simplicity of presentation. At each step, after a tag t_i is selected, we adjust the score for each remaining tag t' as follows:

- Penalize tag t' by removing the redundant information, e.g., by subtracting $P_a(t'|t_i)*S(t_i,o)$ from $S(t',o)$, i.e.,

  $$S(t',o) = S(t',o) - P_a(t'|t_i)*S(t_i,o)$$

  This minimizes the overlap of the concepts identified by the suggested tags.

- Reward tag t' if it co-occurs with the selected tag t_i when users tag object o.

  $$S(t',o) = S(t',o) + P_s(t'|t_i;o)*S(t_i,o)$$

  Since, a user is not likely to tag a given URL using tags that are syntactic variances, e.g., `blogs`, `blogging`,

and `blog`. This rewarding mechanism also improves the uniformity of the suggested tags.

This simple principle ensures that the suggested tag combination has a good balance between coverage and popularity.

The algorithm is summarized in Table 1. *T* is the set of tags assigned to a given object *o* by all users. The algorithm suggests a pre-specified number of *K* tags for object *o* to users based on the tags in *T*. The suggested tags are stored in *R*.

Table 1. Basic Algorithm

```
R = {}; // result tag set
T = all the tags assigned to object o by all users;
X = a set of excluded tags
K = pre-specified maximum number of suggested tags;
T = T – X;
Compute S(t,o) for each t in T;

While (T ≠ empty AND |R| < K) {

        //find the tag with the highest additional contribution
        t_i∈T AND S(t_i,o)≥S(t_j,o) for t_j∈T
        AND j≠i

        //remove the chosen tag from T
        T=T-{t_i};

        //adjust the additional contribution of the remaining tags
        foreach tag t'∈T {
                S(t',o)=S(t',o)–
                P_a(t'|t_i)*S(t_i,o)+
                P_s(t'|t_i;o)*S(t_i,o);
        }
        //record the chosen tag
        R = R ∪ {t_i};
}
```

Note that we have adopted a greedy approach to penalize and reward the tag score because of its efficiency, which is important for dealing with Web-scale data. Other more sophisticated algorithms are under investigation.

3.4 Tag Spam Elimination

As tagging becomes more and more popular, tag spam could become a serious problem. In order to combat tag spam, we introduce an authority score (or reputation score) for each user. The authority score measures how well each user has tagged in the past. This can be modeled as a voting problem. Each time, a user votes correctly (consistent with the majority of other users), the user gets a higher authority score; the user gets a lower score with more bad votes.

Let *a(u)* be the authority score of a given user *u*. As we have mentioned before, the goodness measure of a (tag, object) pair is the sum of the authority scores of all users who have tagged the object with the tag, that is

$$S(t,o) = \sum_{u \in user(t,o)} a(u) \qquad (1)$$

Here *user(t,o)* denotes the set of users who have tagged a given object *o* with the tag *t*.

One simple way to measure the authority of a user is to assign authority score of the user according to the average quality of this user's tags (see Equation (2)).

$$a(u) = \frac{\sum_{o \in object(u),\, t \in tag(o,u)} S(t,o)}{\sum_{o \in object(u)} |tag(o,u)|} \qquad (2)$$

In Equation (2), *object(u)* is the set of objects tagged by the user *u*, and *tag(o, u)* denotes the set of tags assigned to object *o* by user *u*. Equation (2) measures the average quality of a given user's tags. The authority score *a(u)* can be computed via an iterative algorithm similar to HITs [7]. Initially, we can set the weight of each user to be the same, e.g., 1.0.

The above formula treats heavy users the same way as light users. It does not distinguish people who introduce original tags from those who follow the steps of others. People who introduce original and high quality tags should be assigned higher authority than those who follow, and similarly for people who are heavy users of the system. One way to handle this is to give the user who introduces an original tag some bonus credit each time the tag is reinforced by another user.

If a tagging application also allows users to rate other users or tagged objects as in many open rating systems [4][5], the authority score from such open rating systems can be incorporated into our collaborative tag suggestion algorithm.

3.5 Content-based Tag Suggestions

In addition to using tags entered by the real end-users as a source for tag suggestion, we can also suggest content-based (and context-based) tags based on analysis and classification of the tagged content and context. This not only solves the cold start problem, but also increases the tag quality of those objects that are less popular.

One simple way to incorporate auto-generated tags is to introduce a virtual user and assign an authority score to this user. The auto-generated tags are than attributed to this virtual user. The algorithm described in Table 1 remains

Table 2. Suggested Tags for the URL http://wiki.osfoundation.org/bin/view/Projects/AjaxLibraries

	Base case	P_a	P_s	P_a AND P_s	P_a AND P_s AND Syntactic Variance Elimination
1 2 3 4	ajax, javascript, library, ajax library,	ajax, library, ajax library, development,	ajax, **javascript**, **programming**, **webdev**,	ajax, javascript, library, **programming**,	ajax, javascript, library, programming,
5 6 7 8	development, programming, wedev, Reference	**javascript**, programming, reference, **webdev**	Development **reference**, library, ajax library	**ajax library,** development, webdev, Reference	development, reference, webdev, Ajax library

intact. This mechanism allows us to incorporate multiple sources of tag suggestions under the same framework.

3.6 Tag Normalization

Collapsing syntactic variances of the same term can fit in the same algorithmic framework, for instance, by computing the bi-grams (shingles of two characters [1]) of the tags in the currently chosen tag set *C*. To adjust the additional contribution of another tag, we compute the set of bi-grams (*S*) of the tag. The additional contribution of the tag can be computed by multiplying its current value with the following factor, 1- $|S \cap C|/|S|$. Other techniques for improving tag uniformity include stemming, edit distance, thesauri, etc.

3.7 Temporal Tags

Tags introduced are often time sensitive, e.g., due to recent events such as Katrina, shifting user interests, or announcement of new products. In My Web 2.0 we have seen a lot of such tags like `iTune` and `ajax`. Thus, a higher weight can be assigned to more recent tags than those introduced long time ago.

3.8 Adjustments

Our algorithm considers a variety of factors simultaneously. Ideally, we would like to train our algorithm by adjusting the parameters, e.g., by dampening tag coverage score, and (ii) by adding coefficients to the penalizing and rewarding forces. What is interesting to speculate is that as an object is being tagged by more people, the penalizing and rewarding forces start to reflect more in the goodness measure.

4. EXAMPLES

To see how effective our algorithm is, we use the URL http://wiki.osfoundation.org/bin/view/Projects/AjaxLibraries (saved in My Web 2.0) as an example. We compare several cases and show how the forces of penalty and reward interact. As a base case, we suggest tags by using the *S* score alone without penalty and reward adjustments. The suggested tags are listed in the first column in Table 2.

In the second case, we consider the penalty adjustment in the column labeled by P_a. In this case, `javascript` and `webdev` are pushed down in the list. This is due to the relative big overlap between `ajax` and `javascript` and the overlap between `ajax` and `webdev.` In our system, P_a(`javascript`|`ajax`)=0.37, and P_a (`webdev`|`ajax`) = 0.22.

In the third case (see the third column of Table 2), we consider the rewarding mechanism without factoring in penalties. As a result, the tags `programming` and `webdev` are pulled higher up in the list due to high P_s values, where P_s(`programming`|`ajax`)=0.31 and P_s(`webdev`|`ajax`)=0.26 respectively. Users who have tagged `ajax` for the URL also tagged the URL with tags `programming` or `webdev`.

The next experiment shows the results of the interaction between the forces of penalty and reward. The results are shown in the fourth column of Table 2. We observe that the joint force pulls the tag `programming` up but pushes the tag `ajax library` down.

If we need to suggest four tags to users, these tags would be `ajax`, `javascript`, `library`, and `programming`. We can see that this tag combination includes three fairly orthogonal facets; *JavaScript*, *library*, and *programming*. At the same time, it also honors the popular demand of users to include `ajax` along with `javascript`.

In the last column of Table 2, we show results with syntactic variance elimination, which pushes the redundant phrase `ajax library` to the bottom of our list. The order of the tags being suggested is also meaningful. What is more important to note is the intricate balance between the forces of reward and penalty.

Table 3 shows more examples of tag suggestions for URLs with variable popularity. We observe that the tags suggested by our algorithm both have good facet mix and are fairly indicative of the target objects.

Table 3. Tags suggested for URLs with varying popularity

URLs	Suggested Tags
http://maps.yahoo.com/	maps, yahoo, directions, reference, map
http://www.php.net/	php, programming, opensource, php home page, development
http://sourceforge.net/	open source, download, applications, programming, projects
http://code.google.com/	google, api, code, opensource, programming
http://delicious.mozdev.org/	firefox, del.icio.us, extension, tags, tools
http://www.apple.com/	apple, mac, computer, ipod, itunes
http://azureus.sourceforge.net/	bittorrent, software, p2p, java, windows
http://blogs.law.harvard.edu/tech/rss	rss, specification, xml, rss-learning, web design
http://eventful.com/	calendar, events, web2.0, community, tags
http://hymn-project.org/	itunes, ipod, aac, mp3, kickass
http://hype.non-standard.net/	music, mp3, blog, audio, aggregator
http://del.icio.us/	bookmark, del.icio.us, tagging, social, blog
http://digg.com/	digg, news, daily, aggregator, rss
http://en.wikipedia.org/wiki/Main_Page	encyclopedia, reference, wiki, knowledge, research
http://johnvey.com/features/deliciousdirector/	del.icio.us, ajax, javascript, tools, xml
http://maps.google.com/	maps, google, satellite, directions, search
http://myweb2.search.yahoo.com/	my web, yahoo, bookmarks, search, beta
http://next.yahoo.com/	yahoo, betas, next, 1 varios technologia, search

5. CONCLUSIONS

The pull model widely adopted by search engines uses limited semantic information of Web content. This makes it hard to personalize search results, detect spam, and discover new or dynamic content. A push model that directly takes inputs from end users has the potential to address these problems. Tagging allows users to assign keywords to Web objects for sharing, discovering and recovering them. It allows ranking and data organization to utilize metadata from individual users directly, and brings some benefits of semantic Web into the current HTML dominated Web.

Since tags are created by individual users in a free form, one important problem facing tagging is to identify most appropriate tags, while eliminating noise and spam. We advocate using the collective wisdom of the Web users to suggest tags for Web objects. We discussed the basic criteria for a good tagging system and proposed a collaborative algorithm for suggesting tags that meet these criteria. Our preliminary experience shows that a simple embodiment of such an algorithm is effective. In the future, we plan to make the following improvements.

- Improve tag browsing experience by applying the same principles in constructing tag cloud, e.g., by presenting tags with good facet mix while considering popularity and user interests. At a high-level, we will investigate how to bridge the gap between taxonomy and faceted systems to get the best of both worlds.
- Develop metrics to quantitatively measure the quality of suggested tags, and study how tag suggestion can help to facilitate convergence of tag vocabulary.
- Introduce automatically generated content-based tags and also consider the time-sensitivity of tags. This addresses the cold start problem as well as the evolution of concepts and user interests over time.
- Improve tag uniformity by normalizing semantically similar tags that are not similar in letters. The bi-gram method cannot achieve this. This would require incorporating certain linguistic analysis features.
- Using voting and existing tags alone may prevent new high-quality tags from emerging. It subsequently can make content discovery harder. In practice, we can do the following to avoid such limitation. (i) We could give new users bootstrapping time to establish their reputation. (ii) Rather than only relying on the tags assigned to a given object, we should also consider the tags across similar objects identified by clustering. (iii) We should allow tags assigned with low score by the algorithm to have opportunity to be judged by users. To do so, we can separate tags into buckets with different score ranges and display tags from each bucket. Thus, we get user's feedback on tags that are identified by the algorithm as having low quality.
- We are in the process of incorporating the full algorithm into My Web 2.0. Part of the challenge is to handle Internet-scale data and Yahoo-scale users.

6. ACKNOWLEDGMENTS

Many thanks to Caterina Fake, Hao Xu, Adrienne Basset, Tom Chi, Chung-Man Tam, Ken Norton, Nathan Arnold, Chad Norwood, and David Rout for many helpful discussions.

7. REFERENCES

[1] Broder, A. Z. "On the resemblance and containment of documents." *In Proceedings of the Compression and Complexity of Sequences,* June, 1997.

[2] Dvorak, John C. "To Tag or Not To Tag, That Is the Question." *PC Magazine, (http://www.pcmag.com/article2/0,1759,1819101,00.asp),* 2005.

[3] Golder, Scott A., Huberman, Bernardo A. "The Structure of Collaborative Tagging Systems." *HPL Technical Report.* 2005.

[4] Guha, R. "Open Rating Systems." *Proceedings of the 1st workshop on Friends of a Friend, Social Networking and the Semantic Web*, 2004.

[5] Guha, R., Kumar R., Raghavan P., and Tomkins A. "Propagation of trust and distrust." *In Proceedings of the Thirteenth International World Wide Web Conference*, 2004

[6] "Interview on tagging with Jon Lebkowsky and Clay Shirky."*http://adam.easyjournal.com/entry.aspx?eid=2632426*, July 28, 2005.

[7] Kleinberg, J. "Authoritative sources in a hyperlinked environment." *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[8] Shirky, C. "Ontology is Overrated: Categories, Links, and Tags." *In Economics & Culture, Media & Community. (http://www.shirky.com/writings/ontology_overrated.html),* 2005.

[9] Xu, Z., Karlsson, M., Tang, C., and Karamanolis C. "Towards a Semantic-Aware File Store." *9th Workshop on Hot Topics in Operating Systems (HotOS IX).* May 18-21, 2003.

[10] Delicious. *http://del.icio.us/*

[11] Flickr. *http://www.flickr.com/*

[12] My Web 2.0. *http://myweb2.search.yahoo.com/*

[13] "OSAF wiki.Journal.HierarchyVersusFacets." *http://wiki.osafoundation.org/bin/view/Journal/HierarchyVersusFacetsVersusTags?skin=print,* 2005.

[14] Rawsugar. *http://www.rawsugar.com/*

[15] Shadows. *http://www.shadows.com/*