

# JabberWocky: Crowd-Sourcing Metadata for Files

Varun Bhagwan  
IBM Almaden Research Center  
650 Harry Road  
San Jose, CA 95120  
vbhagwan@us.ibm.com

Carlos Maltzahn  
University of California, Santa Cruz  
1156 High Street, M.S. SOE3  
Santa Cruz, CA 95064  
carlosm@cs.ucsc.edu

## Abstract

*Finding relevant files in a personal file system continues to be a challenge. It is still easier to find stuff on the Web with its exponential growth than in one's personal file system. Yet, the exponential growth of personal data renders the current services of personal file systems increasingly inadequate. A reason for this failure is the "cold-start" problem: algorithms that dramatically improve a user's ability to find documents on the Web become ineffective in personal file systems because there is not enough information about these documents. We propose JabberWocky, a service that allows users to manage the content of their personal file system by leveraging semantic relationships available on the Web. More specifically, JabberWocky is using keyword/resource associations of social bookmarking web sites as a basis for recommending keywords for files. We chose social bookmarking web sites because of their popularity and because the assignment of keywords (a process also referred to as "tagging") is an established and popular way to manage photos, music, movies, and audio resources on the Web – very much the kind of resources that need to be managed in personal file systems. The goal of JabberWocky is to overcome the "cold-start" problem of personal file systems and to provide recommendations in a scalable way while maintaining the user's privacy. In this work-in-progress report we describe the motivation and challenges of designing a system like JabberWocky, present the initial design of an on-going user study, and briefly discuss what we have learned so far.*

## 1 Introduction

Tagging refers to the act of associating content with appropriate keywords or phrases, and is primarily a human-powered activity. Tagging has been universally acclaimed as an effective means to classify (and subsequently find) information, be it in the forms of images (flickr), web-links

(del.icio.us), videos (youtube), or audio (last.fm). The success of tagging on the Web has led to the introduction of new solutions and services that attempt to mitigate the challenge of efficient and effective content classification and retrieval in personal file systems. The ability to tag content in personal file systems has been offered through many software components, within browsers, and through file system integration. Even so, usage of tags at the file system level has been very limited. This is due to a couple of reasons. First, there is still a significant amount of human effort required when creating tags for individual files. A great inhibitor in being able to use tags to seek information is the manual activity of tagging each and every document and the associated cognitive effort to come up with appropriate tags. Second, the social aspect of file distribution across file systems has not been leveraged. In other words, if user Alice has already tagged a file that is also present on user Bobs file system, then the existing set of tags from Alice (and possibly other users) are generally not available to Bob.

One way to reduce the cognitive overhead of tagging is to use automatic tag recommendation [11, 18, 19, 1]. Tag recommender systems depend on a large collection of existing tagging events. In environments such as personal file systems such a collection generally does not exist, making tag recommendation difficult and ineffective. We refer to this problem as the "cold-start problem" [16].

We propose JabberWocky, a service that allows users to manage the content of their personal file system by leveraging tagging information available on the Web. We do this by formulating a strategy for tag-recommendation and auto-selection of tags that overcomes the cold-start problem. In this work-in-progress report we first give an overview of JabberWocky's design, then discuss our evaluation methodology, present preliminary results of an on-going user study, and conclude with related work.

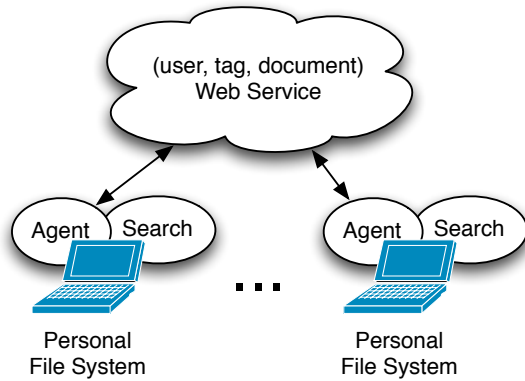


Figure 1. JabberWocky Architecture.

## 2 Approach

Our goal is not to invent another recommender system. Instead, we focus on novel ways to mine the existing metadata present on internet and intranet bookmarking services to provide an effective, personalized service for tag recommendation for files, and to provide this tag recommendation service to a large number of users. This allows us to overcome the cold-start problem associated with recommender systems in file systems by automatically adding relevant metadata to files. Our approach is completely agnostic to the standard file system hierarchies or semantics, and treats each file on an individual basis.

The JabberWocky service (Figure 1) consists of three components: (1) the *social bookmarking service* maintains a large collection of tripartite associations between users, documents, and tags (e.g. Dogear [6] or any other social bookmarking service that provides access to these associations), (2) the *agent* accesses these associations to perform tag recommendation on actively used documents and automatically tags untagged documents based on document similarity, and (3) the *search engine* that allows users to execute queries.

The social bookmarking services collects tripartite associations: the user performing the tagging ( $u$ ), the entity being tagged ( $d$ ), and the set of tags ( $T_{u,d}$ ) used by  $u$  for  $d$ . If document  $d$  is determined to be similar enough of the tagged document  $d$ , then the set of tags  $T_{u,d}$  can be recommended as tags for  $d$ .

The agent computes pair-wise similarity between documents in a personal file system and the tagged documents on a social bookmarking service and supplements untagged files with suitable tags from near-similar bookmarked documents. When working with a document, users can invoke the agent, which then suggests tags based on document similarity. At that time, the user can de-select any unwanted tags, as well as add their own tags. This newly created asso-

ciation between the document and its tags is stored locally, as well as communicated to the cloud-based web-service so as to potentially benefit others with similar documents.

For the search engine we plan to start out with search engines that are provided by contemporary file systems. But we plan to replace it with our own search engine to explore different ranking strategies.

Document similarity has been studied in great detail in the context of de-duplicating content in the World Wide Web. Popular search engines have employed techniques based on shingling algorithms [3, 10, 17] to find near-duplicate web pages. We employed shingling for document similarity in an initial version of JabberWocky’s agent.

If each document is broken down into a set of content-based features (e.g. top keywords), then additional distance metrics such as cosine distance and jaccard distance can be used to determine document similarity. It should be noted that this is a higher-order mechanism for establishing document similarity since it uses document content (as opposed to the shingling approach, which uses numeric fingerprints to determine near-similar documents).

The challenges of our approach are the following: (1) the identification of files that the user wants to tag: an informal analysis of a small sample of personal file systems showed that the over 90% of all files that are not of direct interest to the user but mainly contribute to a functioning system. (2) The comparison of a potentially large amount of data contained in bookmarked content with a typically large amount of potentially sensitive data contained in personal file systems. We expect to heavily use various forms of fingerprinting as well as methods of compression. (3) The distribution of the recommender service into a scalable global component and a personalized local component to minimize the exchange of data for scalability and privacy. (4) The evaluation of JabberWocky depends on the subjective judgment of individual users since the content of their personal file system is by definition not available for public analysis.

## 3 Methodology

Evaluating our approach is posing a significant challenge: due to the subjective nature of tagging it is difficult to quantify what is a “good” tag, especially in relation to another “good” tag. In particular, we need to be able to evaluate the following: (1) tag quality, (2) tag coverage, (3) tagging effort, (4) searching effort, and (5) tag re-use.

Unfortunately, we are not aware of evaluation methods that allow for quantifiable measurements of some of the above metrics. We believe that our ongoing user-study (next section) will allow us to evaluate the first criteria, as well strongly support the fourth criteria. We are currently seeking evaluation methodologies for the remaining items.

We are basing our experiments on two data sets, one from a popular public bookmarking service called Del.icio.us and one from IBM's enterprise-wide bookmarking service called Dogear [6].

The Del.icio.us dataset is a collection of over 1000 web-links that have been bookmarked by a single user over a period of several years. These pages have been crawled, the html converted to text using Perl utilities (non-html pages were ignored), and a docIDtags mapping created. Note that owing to web-decay, several of the pages have either gone missing, or have been modified such that the original set of tags are no longer applicable, and have been ignored as a result.

Our Dogear data was collected in late 2008, and consists of nearly half-a-million user-URL entries. The set of unique URLs (after rudimentary cleanup) was nearly quarter of a million. The total number of users who tagged at least one URL was over ten thousand. On average, each URL was assigned 4 tags. A total of 65k unique tags were used.

## 4 Results

We have conducted two initial experiments to validate our approach, and a third experiment (a user-study) is underway. In the first experiment, we took our Delicious dataset, and computed the top-5 tags for each document using standard text mining techniques. These tags were then compared against the user-created tags for the same documents. We found at least one match in a quarter of the documents. This result was promising because it showed us the possibility of tremendous improvement in tag-recommendation if document similarity was taken into account. In order to test this hypothesis, we conducted a one-off study where an enterprise users work directory was analyzed, and its documents compared against the documents from the Dogear system. To determine similarity, we employed the shingling technique described in Section 2. It became clear rather quickly that although shingling works really well for the web, it has limitations when applied to smaller datasets that are heterogeneous in nature. We are currently investigating higher-order similarity using standard text-mining approaches (e.g. bag-of-words and document summarization), and initial results have been encouraging.

Finally, in order to evaluate the potential of improved tag quality, we have designed a two-part user study (Figure 2). In the first part, we present users with a document and a set of system generated tags. The user is asked to make one of three selections for each tag: a) Spot-on, b) Somewhat relevant, and c) Discard. No tag is selected by default, so the user can choose not to make a selection. Similarly, users are allowed to skip documents. In addition to the system suggested tags, users have the option of entering their own

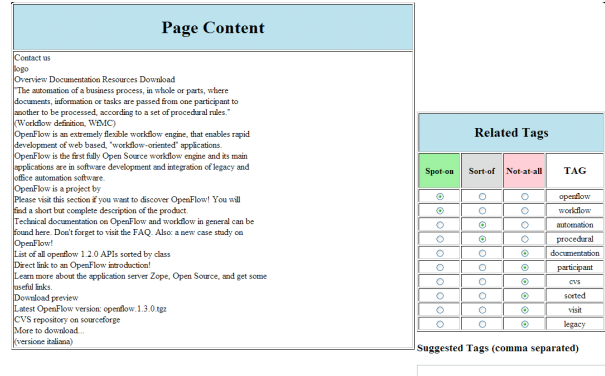


Figure 2. Evaluation Interface.

tags.

In the second part of the study, we take into account the users input during the first part, and use that to generate a new set of tags (akin to our services leveraging the “wisdom of the crowds”). The users are then presented with both the sets of tags, and asked to choose which of the two tag-sets they prefer (they are not told which tag-set is new and which one is the original tag-set).

Although we are still in the process of collecting formal results from this study, we have made a few interesting observations. First, users have reported that scanning the document by applying the search function to one or more of the suggested tags is a quick (and accurate) way to determine the key points in a document. Second, the users attitude towards a recommendation is primed by his or her initial experience: once the user finds the top 1-2 tags to be spot-on he or she tends to overly trust the rest of the recommendations provided by the system, and the converse is true as well. This provides extra challenges for our evaluation of recommendation quality.

## 5 Related Work

Our goal is to import existing human-created metadata from the Internet into individuals desktops, thus helping them better manage their information. Online websites have provided tagging services for a few years now. This has allowed users to add tags to content of all types, including images, video, text, academic publications and the like. These tags have in-turn enabled users to find, classify, relate and aggregate content like never before. Tag recommendation strategies [11, 18, 19, 1] have been proposed to aid in the tag-selection process.

In [8], the authors presented seven different functions performed by tags. They also reported the strong bias on users part towards using general tags (as opposed to more specialized tags). They showed that the first tag used had the

highest median rank, or the greatest frequency, compared to the rest of the tags which generally had a decreasing median rank. This result is very critical for systems such as ours since they auto-add tags to files.

Leveraging the wisdom of the crowds to generate appropriate keywords has been looked at in other contexts such as [7], which attempts to select and recommend appropriate keywords for targeted advertisements.

Tags have also been introduced into the file system, albeit without any remarkable sharing component. Operating systems such as Windows Vista (Metatags) and Apples OS X support tags. In addition, specialized filesystems such as TagFS [2] (now tagsistant) have also been developed. None of these however account for the tremendous bootstrap required to add tags to existing content.

The idea of the Semantic Web has also arrived at the desktop, by way of Semantic Desktops, which translate the tools and protocols developed for the Semantic Web to enable better management and integration of content and applications at the desktop level. Efforts such as MITs Haystack [15] and the IRIS project [4] have attempted to bring semantic-tooling and technology to the desktop. Although this will hopefully enable desktop applications to communicate with each other, a system such as ours is required to be able to effectively exploit Desktop content.

At the file-system and application level, the authors in [9] proposed means to create relationships and demonstrate provenance within files by tracking the coincidence of displayed content. The Universal Labeler [12] attempts to integrate information by enabling labels on a task or project specific basis. The Graffiti system [13] enables sharing of metadata across file systems and users. It aims to make the generated metadata available to applications and users alike.

Commercial enterprises developed bookmarking systems such as IBMs Dogear [14], which allows employees to bookmark and add tags to web-links (both internal and external), Mitres Onomi [5], a trial social book-marking software for the corporate intranet, and IBMs Fringe [6], which enables employees to tag other people with words describing their expertise or interests.

## References

- [1] S. Amer-Yahia, A. Galland, J. Stoyanovich, and C. Yu. From del.icio.us to x.qui.site: recommendations in social tagging sites. In *SIGMOD '08*, 2008.
- [2] S. Bloehdorn, O. Görlitz, S. Schenk, and M. Völkel. TagFS - tag semantics for hierarchical file systems. In *I-KNOW '06*, 2006.
- [3] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. *Computer Networks and ISDN Systems*, 29(8-13):1157–1166, September 1997.
- [4] A. Cheyer, J. Park, and R. Giuli. IRIS: Integrate. relate. infer. share. In *1st Workshop on The Semantic Desktop (at ISWC 2005)*, Galway, Ireland, November 2005.
- [5] L. Damianos, J. Griffith, D. Cuomo, D. Hirst, and J. Smallwood. Onomi: Social bookmarking on a corporate intranet. In *Collaborative Web Tagging Workshop at WWW'06*, Edinburgh, Scotland, May 2006.
- [6] S. Farrell and T. Lau. Fringe contacts: People-tagging for the enterprise. In *Collaborative Web Tagging Workshop at WWW'06*, Edinburgh, Scotland, May 2006.
- [7] A. Fuxman, P. Tsaparas, K. Achan, and R. Agrawal. Using the wisdom of the crowds for keyword generation. In *WWW 2008*, Beijing, China, April 2008.
- [8] S. Golder and B. A. Huberman. The structure of collaborative tagging systems. <http://arxiv.org/abs/cs.DL/0508082>, Aug 2005.
- [9] K. Gyllstrom and C. A. N. Soules. Seeing is retrieving: Building information context from what the user sees. In *IUI 2008*, Maspalomas, Gran Canaria, Spain, January 13-16 2008.
- [10] M. R. Henzinger. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In *SIGIR 2006*, Seattle, WA, August 6-11 2006.
- [11] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. In *PKDD 2007*, Warsaw, Poland, 2007.
- [12] W. Jones, C. F. Munat, H. Bruce, and A. Foxley. The universal labeler: Plan the project and let your information follow. In *Proceedings of the ASIS&T 2005 Annual Meeting*, 2005.
- [13] C. Maltzahn, N. Bobb, M. W. Storer, D. Eads, S. A. Brandt, and E. L. Miller. Graffiti: A framework for testing collaborative distributed metadata. *Proceedings in Informatics*, 21:97–111, 2007.
- [14] D. R. Millen, J. Feinberg, and B. Kerr. Dogear: Social bookmarking in the enterprise. In *CHI 2006*, 2006.
- [15] D. Quan, D. Huynh, and D. R. Karger. Haystack: A platform for authoring end user semantic web applications. In *ISWC 2003*, 2003.
- [16] A. I. Schein, A. Popescul, R. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *SIGIR 2002*, 2002.
- [17] N. Shivakumar and H. Garcia-Molina. Finding near-replicas of documents on the web. In *The World Wide Web and Databases*, Lecture Notes in Computer Science, pages 204–212. Springer, 1999.
- [18] B. Sigurbjörnsson and R. van Zwol. Flickr tag recommendation based on collective knowledge. In *WWW 2008*, 2008.
- [19] Y. Song, Z. Zhuang, H. Li, Q. Zhao, J. Li, W. Lee, and C. Giles. Real-time automatic tag recommendation. In *SIGIR 2008*, 2008.