Tag Data and Personalized Information Retrieval

Mark J. Carman
University of Lugano
Department of Informatics
Lugano, Switzerland
mark.carman@lu.unisi.ch

Mark Baillie
University of Strathclyde
Department Computer and
Information Sciences
Glasgow, UK
mb@cis.strath.ac.uk

Fabio Crestani
University of Lugano
Department of Informatics
Lugano, Switzerland
fabio.crestani@unisi.ch

ABSTRACT

Researchers investigating personalization techniques for Web Information Retrieval face a challenge; that the data required to perform evaluations, namely query logs and click-through data, is not readily available due to valid privacy concerns. One option for researchers is to perform a user study, however, such experiments are often limited to small (and sometimes biased) samples of users, restricting somewhat the conclusions that can be drawn. Alternatively, researchers can look for publicly available data that can be used to approximate query logs and click-through data. Recently it has been shown that the information contained in social bookmarking (tagging) systems may be useful for improving Web search.

We investigate the use of tag data for evaluating personalized retrieval systems involving thousands of users. Using data from the social bookmarking site del.icio.us, we demonstrate how one can rate the quality of personalized retrieval results. Furthermore, we conduct experiments involving various smoothing techniques and profile settings, which show that a user's "bookmark history" can be used to improve search results via personalization. Analogously to studies involving implicit feedback mechanisms in IR, which have found that profiles based on the content of clicked URLs outperform those based on past queries alone, we find that profiles based on the content of bookmarked URLs are generally superior to those based on tags alone.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Experimentation, Performance

Keywords

Personalized Information Retrieval, Folksonomy, Tagging

SSM'08, October 30, 2008, Napa Valley, California, USA.

1. INTRODUCTION

A challenge for researchers developing techniques for personalizing search results is that in order to evaluate their systems, they require personalized relevance judgments; that is, judgments stating which documents are deemed relevant for a particular query by a particular individual. An excellent source of such information is personal query logs and click-through data [5]. Query logs are, however, not readily available to the wider research community due primarily to privacy and monetary concerns. Moreover, the standard test collection in IR, namely the TREC datasets [13], cannot be used for evaluating personalized IR systems, since the topics (queries) and corresponding relevance judgments are not associated with particular users, but are consensus judgments. In short, in order to evaluate personalized retrieval systems one needs personalized relevance judgments. One public source of personalized ratings is tag data, which we discuss below.

1.1 Tag Data

Social bookmarking systems such as del.icio.us[1], StumbleUpon[2] and Bibsonomy[3] are a recent and popular phenomenon. In these systems, users label interesting webpages (or in the case of Bibsonomy research articles) with primarily short and unstructured annotations in natural language called *tags*. These sites offer an alternative model for discovering information online. Rather than following the traditional model of submitting queries to a Web search engine, users can browse tags as though they were directories looking for popular pages that have been tagged by a number of different users. Since tags are chosen by users from an unrestricted vocabulary, these systems can be seen to provide consensus categorizations of interesting websites.

Given this wealth of user-generated annotations, academic interest in these sites has been growing. Various researchers have investigated the applicability of social bookmarking data to the problem of improving Web search results [3, 14]. In a recent analysis of the social bookmarking site del.icio.us, Heymann et al. [7] found that the bookmark data had a good coverage of interesting pages on the Web, in the sense that bookmarked URLs were disproportionately common in search results given the small relative size of the del.icio.us index. Over a set of 30,000 popular queries, they found that 19% of the top 10 results and 9% of the top 100 results were present in the index.

[1] `http://del.icio.us`
[2] `http://www.stumbleupon.com`
[3] `http://www.bibsonomy.org`

In this paper, we investigate the problem of whether social bookmarking data can be used to improve Web search (a particular case of IR) from the perspective of personalization. In particular, we investigate the following questions:

- Can tag data be used to approximate actual user queries to a search engine?
- How can we evaluate personalized IR systems using information contained in social bookmarks (tag data)?
- Is there enough information in (i.e. a strong enough correlation between) the tags/bookmarks in a user's history in order to build a profile of the user that will be useful for personalizing search engine results?

The paper is structured as follows. We first discuss some related work before introducing our approach to evaluating personalized IR using tag data. Next we describe a test collection that we have created by accessing the bookmarking site del.icio.us. We then discuss different models for personalizing search engine results using the data available in our test collection. Finally, we describe the experiments we performed to test whether tag data is indeed useful for personalizing IR, before concluding the paper.

2. RELATED WORK

Heymann et al. [7] recently performed a large-scale exploratory analysis of the utility of tag data obtained from the social bookmarking site del.icio.us. Important findings included the observation that approximately 12.5% of URL's posted by users are new pages (unindexed by a search engine). Therefore, social bookmarking data may be a useful supplementary resource for indexing and crawling. It was also discovered that a large overlap existed between popular query terms and tags, indicating that tagging may be a useful resource to promote popularly tagged urls within a ranking. Another interesting observation was that tags were present in the content of only 50% of pages and 16% in titles of pages annotated by users. Bookmarking data may be a useful resource to address problems such as vocabulary mismatch [6] i.e. the divergence between the language used by searchers when looking for content from that used by authors when creating the content.

There have been a limited number of studies attempting to exploit social bookmark data in order to improve a document rankings [14, 3]. Yanbe et al. [14] implemented a search engine using data extracted from delicious such as tag date and the popularity of a URL. These features supplemented a standard retrieval model through a linear combination of features. Unfortunately, no adequate evaluation was performed in order to determine the utility of these features when included into a ranking algorithm. Bao et al [3] also implemented two ranking algorithms that integrated social bookmarking data into the retrieval model; one exploiting the overlap between query terms and tags, and the second using the popularity of web pages as indicated by the tag data (i.e. popularity increases as the frequency of tags associated with a url increases). An evaluation using 50 computer science related queries indicated improvements over a BM25 baseline (i.e no social history).

Agichtein et al. [2] investigated whether user interaction and behaviour data could be used to classify the top ranked result of a given query. A large sample of historical user data was used to train a classifier using the RankNet algorithm [4] with promising results. In a companion paper, Agichtein et al. [1], investigated how implicit user behaviour, such as click, timing and query features, can be incorporated into ranking process i.e. relevance feedback. Again RankNet was used to determine optimal parameter settings for the ranking algorithm (BM25) incorporating these features.

None of these approaches investigated the use of social bookmarking data to improve a results ranking tailored to the user but were more inspired by a collaborative model to improve web search accuracy as a whole. In the context of personalized rankings to specific users, Shen et al. [10] examined the use of user query and click-through history incorporated into a relevance feedback model. They used statistical language models to improve document rankings by exploiting such data. Evaluation was performed on the TREC AP collection supplemented with click-through data from three users annotating 30 topics.

Other work investigating aspects of personalization in information retrieval include a study by Teevan et al. [12] who analyze the implications of personalization on aspects such as system design and retrieval models with specific focus on the task of re-finding. Although the study indicated that document re-finding may be improved by promoting the rank of a document, a study of click-through data indicated that users were less likely to access the same page if it had moved in the ranking. The time to click on the same page also increased during a change in ranking. These changes, however, may be an indicator of the decrease in the utility of a document for an information need as new information is indexed over time.

Work on using click-through data directly for personalization by Dou et al. [5] found that the utility of personalization is highly dependent on the ambiguity of the query. If the query is highly specific (unambiguous) then personalization is likely to have a negative effect on the results. This suggests that any deployed personalization system will need to estimate the ambiguity in the query so as to apply personalization only when it is likely to improve the results.

In this paper we propose a rudimentary evaluation of user profile features for improving personalized search. Traditionally in IR, large-scale test collections such as TREC are used to offset the need for user-studies, in particular for exploratory problems [13]. However, personalized test collections are not available freely at present. Modifying original test collections for evaluating personalized information retrieval may also be problematic under the TREC evaluation model as a number of key assumptions are made to ensure fair system comparisons, such as that (i) the topics are independent, (ii) all documents are judged for relevance (completeness), (iii) these judgments are representative of the target user population, and (iv) each document is equally important in satisfying the users information need. A number of these assumptions may not be realistic for a personalized test collection.

Evaluating personalized information retrievals is a complex and difficult task. As we have discussed, current approaches use either click through data (which is often not available due to privacy concerns), or are based on small-scale user-studies. In the following section we describe how one can use tag data to evaluate personalized information retrieval systems.

3. EVALUATING PERSONALIZED IR

We propose that an initial evaluation of personalized IR techniques can be achieved using tag data. To do so, we take a document d and the set of tags τ that a particular user has used to annotate that document (or webpage). We assume that the tags, being a summary of the user's thoughts about the page, can be used as a query to a search engine to find the document d. We note here that this assumption is not unusual given that users principally tag pages in order to aid themselves in *re-finding* the page within their often times very large bookmark collection. The re-finding task is a highly investigated topic in Web information retrieval and according to some estimates, re-finding queries account for up to 40% of queries to major search engines [12].

To then test a personalization system, we simply run the personalized and non-personalized systems side-by-side and look at the resulting ranks. If the particular document that the user has tagged ranks higher on the personalized list than the non-personalized one, we consider the system to be working correctly. If the opposite is true (and the tagged document has moved down on the list), we consider the personalization to be having a negative effect. If the two rankings are the same with respect to the position of the tagged document then we cannot infer anything about the workings of the personalization system. The document may have remained in the same position because the initial ranking was optimal with respect to the user's preferences or because the personalization system was not working, but we have no way of telling which occurred, and since the personalization system did not worsen the result, we do not penalize it.

To formalize this test, we denote the ranking of document d in the list of results from a personalized system as $r_p(d)$ and from the not-personalized system as $r_{\neg p}(d)$. We have a positive result if $r_p(d) < r_{\neg p}(d)$ and a negative result if $r_p(d) > r_{\neg p}(d)$.

Thus we compare a personalized system against a non-personalized one to see if a *known relevant document* moves up or down on the list. While it is true that we do not know the ground truth as to what the actual personalized rank $r_p(d)$ should be, it is reasonable to assume that on average the correct value should be equal to or higher than the non-personalized rank $r_{\neg p}(d)$, given that we know this document is deemed to be relevant to the query by the user.

If on average a relevant document should move up the list as the result of personalization, we can measure the number of times this occurs to test the efficacy of each personalization approach. Specifically, we count and compare the number of times the document moves up the list (a positive result, denoted R^+) and the number of times it moves down the list (a negative result, R^-):

$$R^+ = |\{d \in D \mid r_p(d) < r_{\neg p}(d)\}| \quad (1)$$

$$R^- = |\{d \in D \mid r_p(d) > r_{\neg p}(d)\}| \quad (2)$$

where D represents a set of tagged documents.

We can also measure the amount of change in the rankings caused by personalization. This can be important if the amount of movement in one direction is usually larger than movements in the other direction. A measure of retrieval performance that is commonly used for evaluating re-finding tasks is the Mean Reciprocal Rank (MRR):

$$MRR = \frac{1}{|D|} \sum_{d \in D} \frac{1}{r(d)} \quad (3)$$

number of del.icio.us users	14,006
average # of bookmarks/user	28.1
total number of bookmarks	393,739
number of distinct webpages	289,951
average age of bookmarks	74.1 days
average time between bookmarks	4.26 days
standard deviation of time difference	23.8 days
average number of tags/page	2.80
total # of tag occurrences	1,102,042
total # of distinct tags	83,011

Table 1: Statistics of Del.icio.us Dataset Sample

We calculate the MRR over a set of queries before and after personalization and compare the values. The change in MRR is a particularly useful measure, because it takes into account the fact that moving from position 100 to 99 in the rankings is less important than moving from 2 to 1.

Finally, since users of Web search engines are often interested only in the top N results, we also calculate *success*@N values, which count the number of times the document appears in the top N personalized (/non-personalized) results. The *success*@N is given by:

$$success@N = \frac{1}{|D|} |\{d \in D \mid r(d) < N\}| \quad (4)$$

The combination of these metrics: the simple positive/negative counts, the MRR and *success*@N values, provide a characterization of the performance of a personalized information retrieval system with respect to a particular tag-data test collection.

4. TEST COLLECTION

In order to create a test collection, we downloaded data from the social bookmarking site del.icio.us during December 2007. All accesses to the site were made via its public RSS feeds.

In order to download a sufficient amount of data (bookmarks per user) for studying personalization over a large number of users, we proceeded as follows. We first accessed a "recent" bookmark feed which lists bookmarks as they are added to delicious where the number of people who have bookmarked the site is above a given threshold (a parameter which we set to 50). We then accessed the del.icio.us feed for each of the bookmarked documents, which contained a list of all users who had bookmarked the page. After a number of iterations, we had quickly built a list of over 190,000 del.icio.us usernames. In the interest of ensuring user anonymity, we obscure all usernames in our dataset and perform all tests in a randomized fashion.

We then proceeded to randomly sample usernames from this list and download the corresponding RSS feed for each user. This feed contains the most recent bookmarks created by the user and is limited to the 31 bookmarks[4]. We note that the user profiles were *current* (as of December 2007) in the sense that they are the last documents tagged by the user when we accessed the feed, providing a snapshot of the current profile/interests of each individual user.

[4]This threshold is a limit set by del.icio.us. We assumed, however, that 31 bookmarks would be a sufficient sample for evaluation.

Table 1 provides descriptive statistics describing the bookmarking dataset. There are a number of observations to be be made about the test collection. Firstly the average time between bookmarks is on the order of days and the average age of a bookmark (with respect to the user's most recent bookmark) is in the order of months. This is very different from Web search engine query logs where the user may submit multiple queries in the order of minutes. The large time interval between bookmarks does not bode well for personalization, which relies heavily on correlation between topics in the bookmarked documents. Over the space of days or weeks, users are likely to research vastly different topics or even change interests altogether. This phenomenon, sometimes referred to as topic drift, is not as important an issue in query log analysis as the logs can often be divided into fairly homogenous user sessions [11].

Examining the tag data, there are a relatively high number of tags per bookmark (almost 3) indicating that it may well be possible to use this data for approximating queries. In comparison, Silverstein et al. [11] estimated that the average query length in a large Web Search engine query log was 2.35 words/query.

Also interesting is the fact that the vocabulary of tags is not extremely large for our modestly sized test collection, given that tags can be chosen at random by users from an unrestricted vocabulary. (There is no requirement that tags correspond to words from natural language). We note that tag statistics are calculated after punctuation removal, thus some hyphenated tags may be considered the same term.

4.1 Document Collection

After generating a bookmark collection, we proceeded to download the content of the bookmarked webpages, resulting in a collection of 257,955 documents (after badly-formed URLs, binary data, non-existant webpages, etc. had been removed from the list), equating to 11GB of data. Obviously, since each webpage was downloaded after the bookmark was created, we cannot guarantee that the content was not updated after the date of the annotation. However, by concentrating on each user's most recent annotations, we minimize this effect while constructing a realistic environment for testing personalization.

We note that the size of the document collection (around 250,000 documents) is a relatively small snapshot of the entire Web. We contend, however, that the collection is useful for evaluation for a number of reasons; (i) each document is associated with a bookmarked occurrence, (ii) by indexing only annotated pages from del.icio.us, we create a snapshot of what our sample of del.icio.us users find to be of interest on the Web; and (iii) we are interested only in the relative change in ranking of a known *relevant* document not the absolute document rank. This relative change between the baseline and personalized systems would not be affected by an increase in collection size.

5. PERSONALIZATION STRATEGIES

We now discuss various models for personalizing information retrieval results using the information available in social bookmarking data. This is an exploratory study not intended to be comprehensive, but sufficient to support the claim that social bookmarking data is useful (informative enough) for evaluating personalized IR systems. In section 6.3, we show that it is indeed possible to improve search results using some of the profiles discussed below. Before discussing the techniques, however, we need to first introduce some notation. Let $c(w, Q)$ denote the number of times the term w appears in query Q. We denote the relative (normalized) frequency of a term w in a query Q by:

$$p(w|Q) = \frac{c(w, Q)}{|Q|} \tag{5}$$

where $|Q|$ is the number of terms in the query (i.e. $|Q| = \sum_{w \in Q} c(w, Q)$). We note that queries don't generally contain multiple occurrences of the same term, but we use the general notation $c(w, Q)$ so as to keep notation consistent throughout the paper. We use the conditional probability notation $p(w|Q)$ to represent the relative frequency of terms in the query, indicating that it is a Maximum Likelihood estimate for the parameter of a unigram query language model [15].

Similarly, we represent a user by a profile P which is a multiset of terms, and use $p(w|P)$ to denote the relative frequency of term w in P.

5.1 Smoothing

The first issue that needs to be dealt with is how we can combine the query frequency with the user profile. We follow Shen et al. [10], and investigate two smoothing strategies for combining profile information with information from the query itself. These models are *fixed interpolation* and *Dirichlet priors*, and are commonly employed in the Language Modeling approach to Information Retrieval [15].

Using *fixed* interpolation, we perform a simple linear addition of the two sources of information into one expanded (mixture) query model:

$$\hat{p}(w|\theta_{fixed}) = (1 - \lambda)p(w|Q) + \lambda p(w|P) \tag{6}$$

One problem with this simple mixture model is that the effect of the profile is independent of the size of the query Q. It is reasonable to assume, that the longer the query, the more precise it is, and the less useful profile information may be - or inversely, the more likely it is to have a negative effect on the ranking. The second smoothing method takes the length of the query into account and weights the profile less the longer the query. Using *Dirichlet* priors, we have :

$$\hat{p}(w|\theta_{Dirichlet}) = \frac{|Q|}{|Q| + \lambda} p(w|Q) + \frac{\lambda}{|Q| + \lambda} p(w|P) \tag{7}$$

where $|Q|$ denotes the number of terms in the query Q.

5.2 User Profiles

We now discuss various models for generating a profile of the user. There are two sources of profile information that we investigate. The first is the tags a user chooses to label a particular website in his bookmark history. The second is the content of the bookmarked page. There are a number of different ways in which this historical user information can be interpreted in a user profile, leading to a number of different profile types, which we outline below.

We denote the (time ordered) stream of webpages that have been bookmarked by a particular user as $\{d_1, ..., d_n\}$, where d_n is the most recent. The set of tags associated with each document d_i is denoted τ_i and the time at which the user created the bookmark is denoted T_i. In the following we will refer to d_i as the i^{th} bookmark and τ_i as its tags.

5.2.1 Tag-based Profiles

The first *simple* profile involves counting the occurrences of terms in the tags of any of the known bookmarks. The equation for calculating this profile is:

$$p(w|P_{simple}) = \frac{\sum_{i \neq n} c(w, \tau_i)}{\sum_{i \neq n} |\tau_i|} \quad (8)$$

where $|\tau_i|$ denotes the number of terms (tags) in τ_i.

An obvious problem with the first profile is that users often have multiple interests and their many bookmarks cover a range of topics. Thus some bookmarks may be completely unrelated to the n^{th} bookmark (and thus the tags being used as the current query). A simple way to try to force the profile to include only related and therefore relevant tags is to look for *common* tags between the query (the tags τ_n) and the tags used to describe the remaining bookmarks, $\{\tau_1, .., \tau_{n-1}\}$. If we require at least one tag to be common we have the following profile:

$$p(w|P_{common}) = \frac{\sum_{i \neq n, \tau_i \cap \tau_n \neq \emptyset} c(w, \tau_i)}{\sum_{i \neq n, \tau_i \cap \tau_n \neq \emptyset} |\tau_i|} \quad (9)$$

(In practice we require that the common tag be at least two characters long so as to avoid dubious matches between tags.)

An alternative method for trying to ensure that the bookmarks are relevant to the current query is to include only *recent* ones, such as the last k documents:

$$p(w|P_{recent}) = \frac{\sum_{i=1}^{k} c(w, \tau_{n-i})}{\sum_{i=1}^{k} |\tau_{n-i}|} \quad (10)$$

We can also weight older bookmarks less by multiplying their tag counts by a discounting factor δ. This *decaying* profile is calculated as:

$$p(w|P_{decaying}) = \frac{\sum_{i=1}^{n-1} \delta^{n-i} c(w, \tau_i)}{\sum_{i=1}^{n-1} \delta^{n-i} |\tau_i|} \quad (11)$$

Finally, since we know the *time* at which each bookmark was created we can try to discount the profile by an amount proportional to the (relative) age of each bookmark:

$$p(w|P_{time}) = \frac{\sum_{i \neq n} c(w, \tau_i)/(T_n - T_i)}{\sum_{i \neq n} |\tau_i|/(T_n - T_i)} \quad (12)$$

5.2.2 Content-based Profiles

The second source of useful information in the bookmarks is the content of the bookmarked pages themselves. One would expect given the much larger vocabulary of Web pages compared to tag data, that content may prove more useful for personalized search than tags. Indeed previous studies analyzing click-stream data have shown that content-based profiles are more useful than query-based ones [10]. Moreover, since a user spends more time deliberating over which pages to bookmark than they do deciding which search results to click on, one should expect that the content of bookmarked documents to be particularly useful for personalization. Stated in a different way, since a user will only bookmark sites that they find particularly useful or interesting, these documents should contain a lot of useful information about the user.

Once again we start with a *simple* profile that concatenates all documents together and counts term frequencies:

$$p(w|P_{simple}) = \frac{\sum_{i \neq n} c(w, d_i)}{\sum_{i \neq n} |d_i|} \quad (13)$$

where $|d_i|$ denotes the length of (total number of words in) the document d_i.

As was the case for profiles based on tag data, we create a profile called *same-tag*, that includes only bookmarks containing common terms (to the query) in their tag sets, but uses the content of the bookmarked document rather than the tags to populate the profile.

$$p(w|P_{same-tag}) = \frac{\sum_{\tau_i \cap \tau_n \neq \emptyset} c(w, d_i)}{\sum_{\tau_i \cap \tau_n \neq \emptyset} |d_i|} \quad (14)$$

The previous profile is somewhat adhoc in its decision which documents to include and which not to include. In theory, we would like to include all documents that the user has bookmarked, but weight them according to their expected usefulness for resolving ambiguity in the current query. Our first attempt to estimate the distance between two bookmarks is to count the number of common terms in their respective sets of tags (the current query τ_n and the bookmark tags τ_i). This leads to the following *similar-tag* profile:

$$p(w|P_{similar-tag}) = \frac{\sum_{i \neq n} |\tau_n \cap \tau_i| c(w, d_i)}{\sum_{i \neq n} |\tau_n \cap \tau_i||d_i|} \quad (15)$$

We also follow Neubauer et al. [8] and investigate whether calculating the cosine similarity between the current bookmark's tags (τ_n) and a previous bookmark's content (d_i) makes for a useful content weighting parameter. We call this profile *cosine*:

$$p(w|P_{cosine}) = \frac{\sum_{i \neq n} similarity(\tau_n, d_i) c(w, d_i)}{\sum_{i \neq n} similarity(\tau_n, d_i)|d_i|} \quad (16)$$

where $similarity(\tau_n, d_i) = \frac{\sum_{w \in \tau_n} c(w, \tau_n) c(w, d_i)}{\sqrt{\sum_{w \in \tau_n} c(w, \tau_n)^2} \sqrt{\sum_{w \in d_i} c(w, d_i)^2}}$

For completeness, we also test the decaying profile mentioned earlier, this time using the content of documents rather than the tags:

$$p(w|P_{decaying}) = \frac{\sum_{i=1}^{n-1} \delta^{n-i} c(w, d_i)}{\sum_{i=1}^{n-1} \delta^{n-i} |d_i|} \quad (17)$$

And we do the same for the decaying time profile:

$$p(w|P_{time}) = \frac{\sum_{i \neq n} c(w, d_i)/(T_n - T_i)}{\sum_{i \neq n} |d_i|/(T_n - T_i)} \quad (18)$$

6. EXPERIMENTS

We now discuss the experiments performed to investigate whether tag data is useful for evaluating personalized IR.

6.1 Retrieval Engine

For the experiments we use the Terrier Information Retrieval framework [9]. We note that the choice of particular information retrieval system is not important for our experiments as we are not interested in absolute performance, since we are comparing two versions of the same system: a personalized version versus a non personalized version. The most important aspect of our experiments was to fix all aspects

tags/bookmark	≥ 1	≥ 3	≥ 5
bookmarks	56858	28462	12975
success@100	20.8%	22.6%	20.7%
success@10	12.3%	15.5%	14.6%
success@1	5.1%	7.3%	7.3%

Table 2: Baseline Retrieval Performance

of both systems under comparison except for the personalization feature of the new algorithm.

6.2 Personalizing Retrieval

We are interested in applying user profile information to personalize a retrieved list of documents returned from a search engine. There are two approaches to personalize the results from an information retrieval system without changing the internal retrieval function of the system. The first approach is to expand the query with new (possibly weighted) terms. The second is to re-rank retrieved results based on similarity to the user profile. The former approach has the distinct advantage that documents that do not appear in the baseline ranking can still be boosted by the retrieval engine itself to appear in the personalized ranking. We choose this approach for our main experiment. Thus to incorporate the user profile for personalized information retrieval we simply expand the query with terms from the profile, weighting them appropriately. We limit the number of expansion terms to be added to the query so as to limit the amount of noise and total length of the expanded query. We select the the K most frequent terms from the profile and normalize the weights to account for the missing terms.

We use the set of bookmarked documents described in section 4.1 as a test collection. While the collection is relatively small (around 250,000 documents), we decided not to add extra documents to it as that would have only resulted in increased experiment execution times without affecting the results. The *known relevant document* would appear further down the personalized and not-personalized ranked lists (hence the increased execution time), but its relative ordering in the two lists would remain unchanged.

6.3 The Baseline

Before testing the personalized IR approaches, we ran a preliminary experiment to see whether it made sense to approximate user queries using tag data. The experiment involved 2000 del.icio.us users and documents were ranked using the standard BM25 retrieval function. We recorded the position of the bookmarked document in the list of search results given the bookmark tags as the input query. The *success*@$\{1, 10, 100\}$ is given as a percentage for these "tag queries" in the first column of Table 2. The percentages are relatively high considering the source of the queries is tag data. Almost 5% of time the bookmarked URL is the top result and appears in the top 100 results more than 1 in 5 times. Thus we conclude that the tags do a relatively good job of "discovering" the relevant content in the collection. It therefore would appear, that tags may provide a reasonable approximation for user re-finding queries.

It is interesting to note the percentages in the second and third column. If we select only longer queries (more tags per bookmark) the number of times the bookmarked document appears towards the top of the list increases slightly (before dropping again). Intuitively this makes sense - the longer the query the more specific (less ambiguous) it is and the more likely it is to "direct" the retrieval system to the relevant result. This observation supports the rationale in section 5.1 for investigating Dirichlet priors as a smoothing technique, which weights profile information less as query length grows.

Parameter	Value
number of del.icio.us users	2000
min # of bookmarks/user	30
max results list size, N	100
max profile length, K	25
profile decay parameter, δ	0.8
recent bookmarks parameter, k	5
smoothing model	*fixed* or *Dirichlet*
smoothing parameter, λ	0.1 or 1.0

Table 3: Parameter Settings for the Experiment

6.4 Main Experiment

We now discuss an experiment in which we test whether it is possible to improve search results through personalization based on the information present in a user's bookmark history. Before running the experiment we investigated several different parameter settings over multiple short runs involving hundreds of del.icio.us users. The chosen parameter values are shown in Table 3. We claim that these values are reasonable but not optimized, as optimization would have involved considerable computing resources and was outside the scope of this paper. Moreover the values for some parameters, such as the maximum profile length, are highly dependent on the particular retrieval system being used.

We make note of the fact that we only chose users for the experiments who had bookmarked at least 30 pages. We did this for practical reasons – that there be enough historical information about the user for personalization systems to have any hope of improving performance. There were 11,731 users in our sample of del.icio.us who had bookmarked at least 30 documents. Of them we randomly picked 2,000 users for our large scale evaluation.

Since each user had bookmarked at least 30 documents, our evaluation involves approximately 60,000 queries. This query set is orders of magnitude larger than related work on personalization such as that of Shen et al. [10]. The reason for not using all of the data available was due to resource limitations. We considered 2,000 to be a sufficient sample for drawing inferences on a wider population of users.

For each bookmark in the user's history, we compared the top 100 personalized and not-personalized rankings looking for the bookmarked document. If the document appeared in either ranking we recorded its position. The results for the first part of the experiment, using *fixed* interpolation as a smoothing model ($\lambda = 0.1$), are reported in Table 4. The table contains the metrics defined in section 3, namely the positive and negative counts: R^+ & R^-; the change in the Mean Reciprocal Rank (MRR) compared to the baseline system, and the change in the *success*@N values. Results in the first two columns (R^+ & R^-) are given in bold if they are statistically significantly greater than one another according to the Sign Test at the $p \leq 0.001$ level.[5]

[5]The Sign Test doesn't take ties into account when estimating significance, but the number of tied rankings was small enough to be ignored.

Profile	Type	R^+	R^-	ΔMRR	$\Delta success@\{1,10,100\}$		
tag	*simple tag*	**6087**	4104	0.0074	0.14%	1.87%	6.09%
	common tag	**4288**	3590	0.0007	-0.15%	0.51%	2.45%
	recent tag	**5642**	4231	0.0071	0.23%	1.71%	5.08%
	decaying tag	4035	**6709**	-0.0176	-1.38%	-2.47%	-2.90%
	time decaying tag	2957	**5738**	-0.0160	-1.20%	-2.46%	-3.16%
content	*simple content*	**5475**	4289	0.0002	-0.18%	0.46%	2.36%
	same tag content	**5409**	3853	0.0013	-0.13%	0.64%	2.36%
	similar tag content	**5446**	3849	0.0015	-0.09%	0.62%	2.25%
	similar content	**5402**	4786	-0.0021	-0.30%	-0.03%	1.05%
	decaying content	**6052**	3897	0.0051	0.10%	1.33%	4.35%
	time decaying content	**3180**	2304	0.0020	0.00%	0.61%	2.21%
	baseline			0.0746	5.12%	12.09%	20.66%

Table 4: Profile Comparison for 2000 users, with BM25, Fixed Interpolation Smoothing and $\lambda = 0.1$

Profile	Type	R^+	R^-	ΔMRR	$\Delta success@\{1,10,100\}$		
tag	*simple tag*	**6433**	5442	0.0055	0.03%	1.57%	5.59%
	common tag	4377	4152	-0.0015	-0.33%	0.22%	2.08%
	recent tag	5671	5582	0.0028	-0.13%	1.09%	3.85%
	decaying tag	5121	**6239**	-0.0058	-0.69%	-0.28%	1.19%
	time decaying tag	3544	**5732**	-0.0100	-0.83%	-1.40%	-1.17%
content	*simple content*	**5992**	5326	-0.0002	-0.22%	0.47%	1.99%
	same tag content	**5968**	4484	0.0019	-0.18%	0.90%	2.57%
	similar tag content	**6035**	4491	0.0021	-0.15%	0.92%	2.54%
	similar content	5827	5577	-0.0032	-0.49%	0.04%	0.79%
	decaying content	**6759**	4953	0.0053	0.05%	1.54%	4.34%
	time decaying content	**3443**	2906	0.0015	-0.04%	0.60%	2.02%
	baseline			0.0746	5.12%	12.09%	20.66%

Table 5: Profile Comparison for 2000 users, with BM25, Dirichlet Smoothing and $\lambda = 1.0$

The results in Table 4 indicated positive improvements for eight out of the eleven profiles when considering all performance metrics. For the tag based profiles, *same-tag* and *recent-tag* profiles reported strong positive results, improving performance well above the baseline. In general, however, the content based profiles performed better than the tag based ones. Amongst the content based profiles, the *same-tag*, *similar-tag* and *decaying-content* performed the best. This fact leads us to conjecture that a more complicated profiles combining tag similarity and discounting strategies might provide for even better performance.

In the second part of the experiment, smoothing was performed in a query-length dependent fashion using *Dirichlet* priors (with $\lambda = 1.0$). The results of this experiment are shown in Table 5. The results were very similar to those for *fixed* interpolation based smoothing. The content based profiles generally performed better than the tag based ones, with the *simple-tag* profile being the only tag based profile to perform positively. Again, the *same-tag*, *similar-tag* and *decaying content* profiles performing the best amongst the content based profiles. It is interesting to note that the *similar-content* profile did not prove useful indicating that the vocabulary of tags and documents may be inherently dissimilar.

Overall, it appears that there was little difference between the two different smoothing techniques, with the same profiles performing well in both cases. The three different types of performance measures (R^+/R^-, ΔMRR and $\Delta success@N$) did show variation across the different profiles, with only two profiles: *simple-tag* and *decaying-content*, consistently producing positive results across all metrics (including $\Delta success@1$). This observation leads us to conclude that is is prudent to calculate all three families of metrics when using tag data to evaluate a personalized information retrieval strategy.

6.5 Dependence on the Smoothing Parameter

We ran a second experiment (this time over 1000 users) using only the content-based *similar-tag* profile and changing the value of the smoothing parameter λ by $\pm 20\%$. As expected, slight changes in the smoothing parameter had negligible effects on the results, shown in Table 6.

7. CONCLUSIONS & FUTURE WORK

The aim of this paper was to show that tag-data can be viewed as a useful substitute to query logs for evaluating personalized information retrieval systems. In this study, we have demonstrated: (i) that tag data be used to approximate user queries to a search engine; (ii) how to evaluate personalized IR systems using information contained in social bookmarks; and (iii) that there is sufficient information in a user's bookmark history to successfully personalize search results.

In a study on implicit feedback mechanisms in Information Retrieval, Shen et al. [10] found that the content of clicked URLs was much more informative than the past queries used to search for the URL. We have found analogous results with tag data. The content of bookmarked documents is often su-

Smoothing	λ	R^+	R^-	ΔMRR	$\Delta success@\{1,10,100\}$		
fixed	0.08	**2598**	1886	0.0008	-0.09%	0.47%	1.81%
	0.1	**2683**	1909	0.0019	-0.01%	0.63%	2.03%
	0.12	**2742**	1959	0.0023	0.02%	0.69%	2.11%
Dirichlet	0.8	**2908**	2181	0.0019	-0.11%	0.88%	2.22%
	1.0	**2946**	2226	0.0020	-0.12%	0.94%	2.26%
	1.2	**2963**	2291	0.0021	-0.13%	0.95%	2.21%
baseline				0.0732	4.91%	12.13%	20.75%

Table 6: Comparing different Smoothing Parameter values over 1000 users (*similar tag content* profile)

perior to tags when it comes to building user profiles. The best results in our experiments, however, were achieved using a simple tag frequency based profile as well as using a content frequency based profile that was weighted by recency. Good results were also achieved using profiles that combined tag data with the content of bookmarked documents. For these profiles, document content was used to populate the profile while tag similarity was used to weight the contribution of each bookmark.

One possible explanation for the performance of bookmark content over tag data is the vocabulary mis-match problem [6]. The content of bookmarked URLs are likely to share vocabulary with other bookmarked documents, while the user tags for the different bookmarks are also likely to share vocabulary with one another, but a different (user-generated) vocabulary. Hence it makes sense to use the content of bookmarked documents to populate the user profile, and the tags to proportion the influence of each bookmark.

Future work will investigate collaborative filtering approaches to personalization by smoothing a user's profile using the tags and bookmark content of their nearest neighbors in the tag graph. The similarity between users could be estimated by the number of common bookmarks or the similarity of tags used to describe documents. Finally, we plan to investigate the use of a learning-to-rank framework for combining different aspects of the user profile into a single retrieval function.

8. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval*, 2006.

[2] E. Agichtein and Z. Zheng. Identifying "best bet" web search results by mining past user behavior. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 902–908. ACM, 2006.

[3] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing web search using social annotations. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 501–510, 2007.

[4] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM, 2005.

[5] Z. Dou, R. Song, and J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, 2007.

[6] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. Statistical semantics: analysis of the potential performance of keyword information systems. In *Human factors in computer systems*, pages 187–242. Ablex Publishing Corp., Norwood, NJ, USA, 1984.

[7] P. Heymann, G. Koutrika, and H. Garcia-Molina. Can social bookmarks improve web search? In *WSDM '08: Proceedings of the First ACM International Conference on Web Search and Data Mining*, 2008.

[8] N. Neubauer, C. Scheel, S. Albayrak, and K. Obermayer. Distance measures in query space: How strongly to use feedback from past queries. In *WI'07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, 2007.

[9] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR)*, 2006.

[10] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50, 2005.

[11] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.

[12] J. Teevan, E. Adar, R. Jones, and M. A. S. Potts. Information re-retrieval: repeat queries in yahoo's logs. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval*, 2007.

[13] E. M. Voorhees and D. K. Harman, editors. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, Cambridge, Massachusetts 02142, 2005.

[14] Y. Yanbe, A. Jatowt, S. Nakamura, and K. Tanaka. Can social bookmarking enhance search in the web? In *JCDL '07: Proceedings of the 2007 conference on Digital libraries*, pages 107–116. ACM, 2007.

[15] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214, 2004.