A Wiki Based Model of Web Social Search

Antonio Gulli, Stefano Cataudella, and Luca Foschini
Ask.com, R & D
Corso Italia 58, Pisa, Italy
{agulli,stefano.cataudella,luca.foschini}@ask.com

ABSTRACT

Web search is extensively adopted for accessing information on the web, and recent development in personalized search and social bookmarking system has tremendously eased the user's path towards the desired information.

We propose JammingSearch, a novel model that unifies web search and social bookmarking by transparently leveraging a Wiki-based collaborative editing system. When an interesting search result is found, a user can share it with the Jamming Search community by simply clicking a button. This information is implicitly tagged with the query submitted to any commodity search engine. Currently, JammingSearch interacts with Ask.com, Google, Microsoft Live, and Yahoo!. This allows the JammingSearch's community to build an inexpensive and highly focused source of tagged information which can be exploited to increase the quality of web searches. To the best of our knowledge, JammingSearch is the first attempt to create a synergy between web search, social bookmarking and collaborative editing systems. In this paper, we argue the novelty JammingSearch, supporting our investigation by providing a prototype implementation and a preliminary evaluation. In addition, we introduce a novel ranking system for both the users and the information submitted by the community.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Online Information Services; H.5 [**Information Interfaces and Presentation**]: User Interfaces

Keywords

Collaborative editing, Wiki, Social Web Search, Social Bookmarks

1. INTRODUCTION

Nowadays, search is undoubtedly the primary way for users to access the information on the web. According to Nielsen [15], Google, Microsoft and Yahoo! were the three U.S. most visited properties as April 2008. Interactive Corp, Ask.com's parent company, occupies the 7th place; while the remaining top positions are taken by sites which actually provide content, such as Time Warner, News Corp, eBay, Wikipedia, Amazon and Walt Disney. The dominance of web search can be explained by the fact that search engines have become the main access to web contents, which, on the other hand, is hardly ever sought directly on the content provider's web site. As an example, in 2007, the 70% of Wikipedia's upstream visits came from external search engines, according to Hitwise [20].

However, search engines drive users to the desired information since they index distinct, albeit overlapping, portions of the web [3] and since they provide different results for the same query [13]. Users take advantages of this diversity: 48% of searchers regularly use two or three search engines, only 7% use more than three and 44% use just one, as shown in a recent study [16].

Although search plays a important role in user's navigation, the user activity on the web cannot be reduced to only search-related routines. Once the desired information is found, it has to be first, fruitful exploited and, secondly, possibly stored for user's future needs. This mechanism, called *bookmarking*, has been extended to become more and more social. With the advent of *social bookmarking*, bookmarks can be shared among different users and can be organized in different categories, annotated with metadata, and searched chronologically or by tag. The functionalities of search engines and social bookmarking sites are both aimed at providing the user with high quality information. Even if the search and social bookmarking are two worlds with similar intents, their integration has never been explored in-depth.

In this paper, we address this issue by proposing JammingSearch as a novel model that unifies Web search and social bookmarking. JammingSearch is a service made-up of two parts. On the user side we have a simple browser extension, while on the server side we have a centralized wiki, named JammingWiki. When the JammingSearch browser extension is activated, a user can perform a query on their preferred search engine and decide to store a selected part of the search results with a single click. This selected part can be one or more search results, or even a portion of them. The content is automatically tagged with the query issued by the user and transparently stored on the JammingWiki.

The users of the JammingSearch community are then allowed to access the JammingWiki to customize and edit their saved results and also to share and tag them, enabling a new

CIKM '08 Napa Valley, California USA

form of social activity online.

We stress that in our model, content tagging is transparent and automatic since tags are simply the queries submitted to the Web search engines. We also note that JammingSearch can seamlessly leverage search results given by different search engines. A user can start a search session on Microsoft Live, store some of the results on JammingSearch, and then move to Ask.com, Yahoo! or Google and continue to populate our social system. All the information stored in JammingSearch is immediately made available online to the social community and can be searched by keywords or tags.

The contributions of this paper are the following:

- We propose JammingSearch, a system that allows a synergic interaction between Web search and Social bookmarking by leveraging a collaborative editing system. This interaction is not intrusive with normal search activities and preserve the user privacy;
- We point out how JammingSearch implements a simple model of social tagging, where queries submitted to search engines are employed to tag portions of search results selected by the user;
- We show that this social bookmarking activity can be used to provide an improved form of Web search, where the ranked results returned by search engines are combined with the users' selections, tagging and Wiki editing.
- We introduce a new ranking model which orders both the users of our community, and the information they submit. This model can reduce the malicious content and the spam submitted to our system and can help to identify premium users in our community.

2. RELATED WORK

As we have seen in Section 1, JammingSearch is based on different models for discovering and sharing information on the Web. In this Section, we review those models and analyze the relationship between social bookmarking and search engines. We also discuss the importance of collaborative editing tools, on which the implementation of JammingSearch is based.

Social Bookmarking: A Social bookmarking system is a Web site that allows users to store, organize, and manage bookmarks of Web sites. These bookmarks are typically public, but can also be made private, or can be shared within a selected group of users.

Most of the Social Bookmarking sites advice the user to enrich the bookmarks with *tags* made up of a few distinctive words and to give *votes* on other users' contributions. The concept of shared online bookmark was pioneered by iList, Backflip, Clip2, ClickMarks, HotLinks , while the term *Social Bookmarks* was coined by Del.icio.us in 2004. Since then, many other social bookmarking services have been proposed, such as Furl, StumbleUpon, Simpy, and Diigo. Only recently the model of social bookmarking has been exported to more specific fields like online newspapers and academic publications. Digg, Reddit, and Newsvine allow users to bookmark and vote news items found on the web. With Lotus Connections IBM has taken a commercial outlook to social bookmarking, while CiteULike and BibSonomy [4] represents an application of the social bookmarking concept to the field of academic research. The interested reader can find in [17] a complete list of social software and online services.

The main advantage of social bookmarking is that discovering, tagging, and organizing the information are carried out by human beings. This means that users can potentially organize the content by leveraging their past experiences and tastes. Therefore, they can obtain a more accurate classification, when compared to automatic techniques yielded by machine learning, data mining or information retrieval. The fact that social systems are built by people is their greatest feature but, it also brings up several issues. A first problem is that different users can denote the same semantic concept by using different terms. A second problem is that users can deliberately spam a social repository with the sole goal of poisoning the common knowledge. In Section 3 we discuss how JammingSearch address those limitations. [1]

Social Bookmarking and Web Search: In the last years, major search engines have already incorporated customized solutions for social bookmarking. Google allows users to bookmark Web pages contained in Google's Web History, a service that tracks queries and search results for users who are logged on Google Account. Ask.com's MyStuff and Yahoo! Bookmarks offer a similar service also allowing users to specify public folders of bookmarks which can be searched and shared by others. These service are useful but cannot be extended to work with different search engines. A user searching on Google cannot share its bookmarks with others using Yahoo!, Ask.com, Microsoft Live, and vice versa. In Section 3, we describe how our model of Web social search allows users to seamlessly leverage the search results returned by different search engines.

Wikis: Wikis are the epitome of web based collaborative systems, a paradigm for creating and maintaining content managed by a community of users. In Wikis, users can modify the content using a simplified markup language or a WYSIWYG editing tool. A page history mechanism is usually provided to allow editor to restore a prior version of some content. Several software solutions [2] implement the Wiki paradigm and extend it adding features such as different authentication systems, advanced editing capabilities, and the possibility to link or embed non-textual information.

Wikipedia is undoubtedly the best known deployment of the Wiki model. Wikipedia contains 10 million articles in 253 languages, and has more than 680 million visitors per year. Even if there is no formal peer-review process, millions of volunteers constantly monitor any change in order to remove nepotistic links, spam and biased content [22]. The software that runs Wikipedia is MediaWiki [14] which proved to be very effective in terms of performance trough the extensive use of different levels of caching, load balancing and database replication. These features made MediaWiki the choice to run the server side part of JammingSearch, as we

[1] AG:indirizzare e definire il termine il problema della folksonomia. LUCA.

[2] WikiMatrix [18] counts more than one hundred Wiki software solutions.

discuss in Section 3.

Wiki and Web Search: Wikia Search [19] is the first attempt to integrate the traditional Web search with the Wiki collaborative model. Wikia Search, a part of Wikia (originally, Wikicities) has received a lot of attention, mainly because of its creator, Jimmy Wales who serves as Chairman Emeritus of the Wikimedia Foundation. Wikia Search allows users to create search content or "*mini articles*", which are short editorial contents related to the search terms. If no matching article exists, the user has the opportunity to write a new one. Wikia is an open source effort to create editorial content on top of the search engines' ten blue links. Albeit Wiki Search's high-profile goal, Techcrunch defined it as *"a Complete Letdown...one of the biggest disappointments I have had the displeasure of reviewing"*. The main reason behind this very harsh judgment can perhaps be found in the welcome page of Wikia, which reads *"Search Wikia is still very much under development, and we are aware that the quality of the search results is low."*. The point is, Wikia Search requires users to change their habits and leave their favorite search engine in favor of their lesser known and still under development service. On the other hand, JammingSearch does not require any switching costs, since users are encouraged to continue searching their favorite search engine.

We conclude our discussion of the present work by reviewing the ideas recently arisen in the academic research and following the same direction of JammingSearch. In fact, the intuition that Web Search and Social Bookmarking can mutually gain from each other, has already been investigated by scholars. Krause et al. [7] compares search in social bookmarking systems with traditional Web search. They show that a graph-based ranking approach on folksonomies yields results that are close to the rankings of the commercial search engines. Yeung et al. [12] extract semantic information from the collaborative tagging contained in Del.icio.us. This information is then used to disambiguate search results returned by Del.icio.us and Google. In [11], Yanbe et al. proposes to use data from social bookmarking systems to enhance Web searches. They suggest combining link-based ranking metric with the one derived using social bookmarking data. Our system extends these proposals since it is based on Wiki, works with multiple engines and is not intrusive for the user.[3]

3. JAMMINGSEARCH

In this Section we will present our novel contribution to the world of social bookmarking systems: JammingSearch, an automatic system which is able to seamlessly integrate Social Bookmarking, Wiki, Web Search and User's Personalization in a unified view. In Section 2, we already pointed out the major problems of social bookmark systems. Namely, user laziness, ambiguous tagging model, and malicious or spam content. In this section and in the next one, we will address those limits.

3.1 A Query Based Tagging Model

The key idea beyond JammingSearch is the following. A search query can be seen as a summary of each result retrieved by a search engine. This description becomes a very effective tag if one of those results is clicked by the user. For instance, if a user searches the query *"madonna songs"* and selects the `http://www.likeavirgin.com`, that web page can effectively be tagged with the keywords contained in the query. This intuition is very simple but defines a clear and uniform tagging model.

Figure 1: Search Engines supported by JammingSearch

JammingSearch has been tested to work with Google, Yahoo!, Microsoft Live Search, Ask.com and AOL (see Figure 1), on the contrary of other social bookmark systems which can interact only with proprietary search engines. As a consequence our social engine can be populated very quickly, since those engines have a large audience.

3.1.1 Client side operations

In Figure 2 we describe the typical usage of JammingSearch. A user submits the query *"beyonce songs"* by using the preferred search engine, in this case Ask.com.

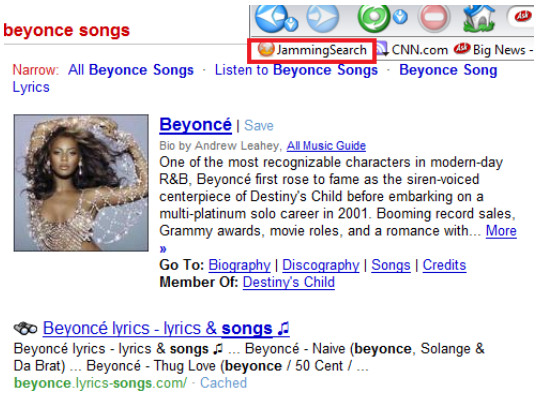

Figure 2: JammingSearch **for "Beyonce songs".**

Then, clicks on a search result and enters the page given in Figure 3. If the user likes the web content, he can bookmark it by clicking on a button installed on the browser. This button is a bookmarklet, a little snippet of javascript code that is executed when it is clicked. It captures:

1. The query submitted to the search engine, regardless of its commercial brand;

[3]AG: Serve espandere questa parte, perche' noi siamo meglio? LUCA STEFANO

2. The URL and the Title of the selected Web page;
3. A snippet of text. (Optionally, the user can highlights a part of text simply by using the mouse over the Web page);
4. A UserID, if the user is logged in our system.

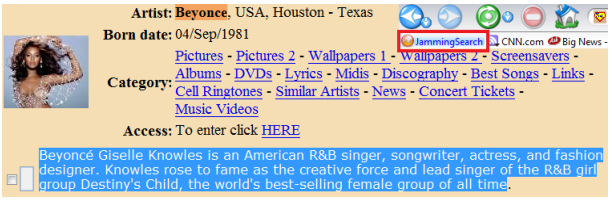

Figure 3: Picking one search result with JammingSearch for the query "Beyonce songs".

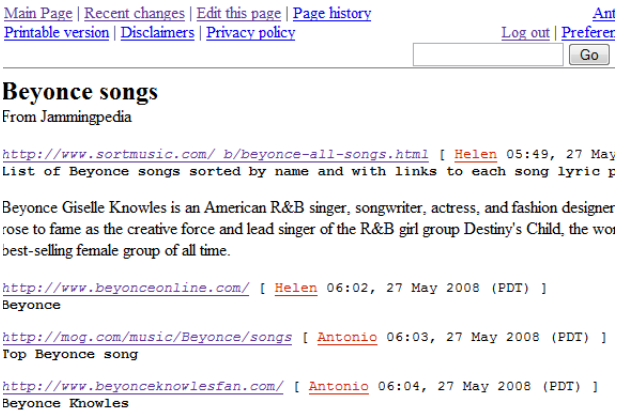

Figure 4: Wiki Social Bookmarking for "Beyonce songs".

Our bookmarklet works with all the major browsers such as Mozilla, Internet Explorer, Opera, Safari and many others. The bookmarklet can be sent via email or embedded in a web page as an hyperlink. The user who wants to install JammingSearch simply needs to *bookmark* this hiperlink on his browser and to show it on the bookmark toolbar.[4]

We like to point out that users do not need to interrupt their normal activity of searching and browsing, in order to interact with our system. The benefit for the user is to have a network place where to store those search results considered as useful. The benefit for the community is to leverage the contribution of each different user. Our JammingSearch is as a social bookmark system that works with all major Web search engines. We are *not* creating yet another search engine aiming at attracting users, but instead we encourage the users to continue using their preferred Web search engine. If they want to start a social activity on web search results, they can transparently use our system as an *auxiliary* tool.

Besides, we remark that our client extension is non intrusive and preserve the user privacy. No information is sent to our servers, if the user is not willing to explicitly submit it.

3.1.2 *Server side operations*

JammingSearch's bookmarklet sends the information selected by the user by means of an HTTP connection. Our central server stores this content in a structured format. We decided to adopt the MediaWiki markup language to store this data, since it is simple and naturally fits with our model. In addition, this choice allows leveraging all the infrastructure optimization currently adopted for Wikipedia. In Figure 4 there is an example of Wiki Social bookmarks of search results selected for "Beyonce songs".

3.2 A Wiki Based Model of Web Search

The structured content stored on JammingSearch can be edited when needed. An example is given in Figure 5 for the tag "restaurants in new york". The Wiki approach makes easy to track any modification on our index. This mechanism is similar to the one adopted by Wikipedia and allows the social community to control editorially any abuse of our system such as spam, obscene or malicious web content. In Figure 6, the list of recent changes applied to the pages is shown.

Figure 5: Optional Editing Wiki Social Bookmark for the query "Restaurants in New york ".

Besides, a mechanism of user access control is available. A user can join an already defined group of users or create a new one. After joining a group, he can decide to share his own personal bookmark with other members. We are currently extending this mechanism to allow a finer grain control of the access rights given to any information submitted to JammingSearch.

Figure 6: Recent updates on JammingSearch.

[4]Each browser gives a slightly different name to this toolbar. Here we are adopting the Mozilla terminology. Internet Explorer denotes the same toolbar with the *Favorites* name.

In addition, all the content submitted by the user is indexed on-the-fly and is available for search. Figure 7 is an example of results for the query *"winehouse"*. Currently, we adopt MYSQL as indexing engine, similar to the one adopted by Wikipedia. In section 4.1, we propose a different ranking algorithm which leverage users' clicks and `TF X IDF` text score. Communities adopting JammingSearch contribute to populate our index with fresh content. Any bookmark sent to our engine is annotated with a temporal tag that represents the instant when that information has been submitted. This temporal tag may be used to boost fresher search results, as we describe in next section.

Figure 7: An example of search on JammingSearch.

4. RANKING

In this Section we introduce the ranking scheme adopted by JammingSearch. It orders both the users in the community and content they submitted. This model is an extension of the one we introduced in [1] for ranking a stream of news.

4.1 Ranking Bookmarks and Users

In JammingSearch there are two different types of subjects that can be ranked: bookmark items and users. In particular:

- A bookmark item receives a positive vote in two situations: (a) each time it is submitted by an user via a bookmarklet or (b) every time it is clicked by an user during browsing or searching activities;
- A user receives a positive vote every time that a different users clicks on any of his bookmark items;

The key intuition is that user clicks and submissions reflect the social activities on our community. If a page is selected by many different users, it can gain a higher position in our bookmark ranking. If a user produces content that other users like, he can gain a higher position on our community ranking. The idea is similar to the one used by the HITS algorithm [6], in identifying hubs and authorities.

In JammingSearch users are encouraged to identify themselves with a login process. Bookmarks created under a logged-in session are accessible later on, and univocally associated to the creators. This straight association allows us to rank users and bookmarks items with no ambiguities.

4.1.1 Adding Temporal Information

In our system, user clicks and bookmarks are annotated with temporal information. Our ranking scheme can be formalized by adopting a model similar to the one we proposed in [1] for ranking a stream of news items. Let $G_w = (V, E)$ where $V = U \cup K$ and U are the nodes representing the users and K are the nodes representing the bookmark items seen in a time window ω. Analogously, the set of edges E is partitioned in two disjoint sets E_1 and E_2. E_1 is the set of undirected edges between U and K. It represents either the bookmark item creation process or the click on bookmark items. Here, we assume that anonymous clicks or bookmark submission are generated by a special node in U. E_2 is the set of undirected edges with both the endpoints in B and represents the bookmarks saved under the same tag, according to the model described in Section 3.1. Analogously to what we have proposed in [1], the adjacency matrix of the user-bookmarks graph is given by

$$A = \begin{bmatrix} 0 & B \\ B^T & \Sigma \end{bmatrix}.$$

The matrix is obtained by assigning an identifier to the nodes G so that all the identifiers assigned to nodes in U are less than identifiers assigned to nodes in K. The submatrix B refers to edges from users to bookmarks, and $b_{ij} = 1$ iff the user i created a bookmark j. The submatrix Σ refers to edges from bookmarks to bookmarks, and $\sigma_{i,j} = 1$ iff the bookmarks i and j are tagged under the same query. In [1] we discussed two non-time aware and three time aware algorithms for ranking the nodes in G. For space constrain reasons we describe one time aware ranking scheme and invite the interested reader to refer [1] for a detailed description of other schemes.

Let $R(b_i, t)$ be the rank of bookmark item b_i at time t, and analogously, $R(u_k, t)$ be the rank of user u_k at time t, where $k \in \{1 \dots n\}$ and $i \in \{1 \dots m\}$. Moreover, by $S(b_i) = u_k$ we mean that b_i has been posted by user u_k and by $C(b_i)$ the set of users that clicked the bookmark item b_i.

Decay rule: We adopt the following exponential decay rule for the rank of b_i which has been submitted at time t_i:

$$R(b_i, t) = e^{-\alpha(t-t_i)} R(b_i, t_i), \quad t > t_i.$$

The value α is obtained from the half-life decay time ρ. ρ is an input parameter denoting the time required by the rank to halve its value, with the relation $e^{-\alpha\rho} = \frac{1}{2}$ and $\alpha, \rho > 0$. In addition, we consider another parameter β, $0 < \beta < 1$, which gives us the amount of user' s rank we want to transfer to each bookmark item. Similar to [1], our ranking scheme is defined by the following equations:

$$R(b_i,t_i) = \left[\lim_{\tau\to 0^+} R\left(S(b_i), t_i-\tau\right)\right]^{\beta} + \quad (1)$$
$$+ \sum_{u_j\in C(b_i)} \left[\lim_{\tau\to 0^+} R\left(u_j, t_i-\tau\right)\right]^{\beta} +$$
$$+ \sum_{t_j<t_i} e^{-\alpha(t_i-t_j)}\sigma_{ij}R(b_j,t_j)^{\beta}$$

$$R(u_k,t) = \sum_{S(b_i)=u_k} e^{-\alpha(t-t_i)}R(b_i,t) + \quad (2)$$
$$+ \sum_{S(b_i)=u_k} e^{-\alpha(t-t_i)} \sum_{\substack{t_j>t_i \\ S(b_i)\neq u_k}} \sigma_{ij}R(b_j,t_j)^{\beta}$$

Equation 1 says that the rank of a bookmark item b_i at time t_i depends on three factors: (a) the rank of the user that posted the bookmark item, and (b) the rank of the users who clicked on it, and (c) the rank of other bookmark items previously posted under the same query tag. The importance of those previous bookmark items is decayed by a negative exponential factor. Note that the limit used in the equation captures the idea of a user rank computed "a little before" the time t_i. We invite the interested reader to refer [1] for the mathematical justification of this intuition.

Equation 2 says that the rank of a user u_k at time t_i depends on two factors: (d) the rank of the bookmark items that the user posted and (e) the rank of bookmark items under the same query tag of the bookmark posted by the user. This last factor is a "bonus" to those users who starts wiki pages which, later on, becomes popular among other wiki contributors. Note that the parameter β is similar to the magic ε accounting for the random jump in Google's Pagerank [8] and guarantee that the fixed point equation involving the users, has a non zero solution. We remind that in both equations, the time t_j is the posting time of bookmark b_j.

Note that our ranking scheme boosts those bookmark items posted or clicked by important users. This is a desired property which can help to demote spam or malicious content posted in our system.

4.1.2 Ranking at query time

When a user submits a query to JammingNet, the list of the bookmark items matching the query terms is ranked according to the following equation:

$$R(b_i,t_i,q) = \gamma\ \texttt{textRank}(\texttt{b}_\texttt{i},\texttt{q}) \ +\ (1-\gamma)\ R(b_i,t_i) \quad (3)$$

where $0 < \gamma < 1$ and `textRank(`b_i`, q)` is a `TF x IDF` textual ranking function. In future we plan to explore using SVM [2] (with linear kernel) to learn the optimal weight γ.

4.2 User Personalization

JammingSearch, allows three types of personalization:

1. **Snippet personalization**: Each user can highlight a snippet of the page and bookmark it in our system;
2. **Wiki bookmarks personalization**: Each user can edit indexed content, adding comments, removing wrong links, malicious information or spam;
3. **Group access personalization**: Each user can decide to mark his bookmarks as private or to share them with other groups of users;

In future works we plan to extend the ranking model given in Section 4.1.1 in order to take into account group access personalization. This will produce a personalized form of social ranking, but may generate scalability problems that still need to be addressed. In [9] Scott et al. propose a personalized variant of HITS. Anyway, the problem of computing personalized time-aware variants of HITS is still open.

5. EXPERIMENTAL RESULTS

5.1 Performance

On the client sides, browsers communicate with JammingSearch using a light bookmarklet extensions and standard HTTP connections. This is not adding any overhead to normal user activities. On the server side, JammingSearch uses the same infrastructure as Wikipedia. The interested reader can refer [21] for a discussion of strategies adopted by MediaWiki to address scalability issues (load balancing, distributed MySQL, memcache, and so on).

5.2 User Survey

We evaluated the features offered by JammingSearch running a survey on a group of twenty users with different Internet experience, ranging from very basic to advanced usage. The following questions have been asked:

1. *If you like a search result, would you mind to click on a button on your browser, so that you can bookmark your search result? This bookmark will be saved on a wiki page, tagged with the query you submitted.*

   This question aims to evaluate the user impression on the whole system. Moreover, we wanted to evaluate the ease of use of our bookmarklet and its intrusiveness in the users' activities. 75% of the interviewed users said that they would like to have their bookmarks in a centralized location, and be able to search them. Moreover, 90% of the users said that the system is very user-friendly, since it does not force the user to leave the current page, in order to create a bookmark.

2. *Would you like to personalize your bookmarks for search results, by adding comments, and voting them?*

   This questions aims to evaluate the personalization's options provided by the system. 100% of the interviewed users liked the chances to add comments to the bookmarks, while 75% thought that voting could be useful to clean the system and improve its results.

3. *Would you like to share your bookmarks with other users, and see what they have bookmarked for your same web search queries?*

   This question looks at the social aspects of our system, and in particular to the possibility of sharing the bookmarks offered by JammingSearch. 90% of the interviewed users would like to know what other users have picked for the same query they submitted.

4. *Having a sufficiently large base of queries, would you use* JammingSearch *as an alternative search engine?*

   This last question aims to evaluate what the users think about merging the results coming from different Web Search engines, as well as combining the results with the users' preferences. 90% of the interviewed users liked the idea of finding all the results picked by the users, in particular if they come from different search engines. In practice, *"it is like doing several searches for the same queries at the same time"*, a user said.

5.2.1 User Satisfaction

We tested the user satisfaction using a set of 20 queries. Twenty users with different Internet experience were partitioned in two distinct groups of 10 users each. The first group was asked to search for the queries on their preferred search engine(s), and bookmark the results they considered relevant. Moreover, they were also asked to add comments to their bookmarks, if necessary. The second group was asked to perform the same queries both on their preferred search engine(s) and on JammingSearch, and to compare the time required to find the desired result using the two solutions.

We noticed that the time required by the second group of users, when searching on JammingSearch, was always less than the time required to search on traditional search engines. This was mainly due to the fact that the users comments added by the first group helped the users in the second group to find quickly their results.

Here below follows a couple of queries where we had the most significant time difference for finding the desired information between the traditional search engines and JammingSearch:

1. **Vmware Workstation configuration on Ubuntu**
   This query is about the configuration of a software on a Linux operating system. Main engines give as search results many forums, where is possible to find tutorials on how to configure the software. In this particular case, the user is asked to download a patch (not supported by the Vmware house), and the link to this patch is often buried under a lot of posts in the forums. In this case, the main advantage given by JammingSearch is that the first group of users had already linked the patch into our Wiki. Therefore, the second group of users is not forced to read or scroll an entire forum for finding the desired information. We found similar results for other queries which required the users to access some forums.

2. **Latex tutorial**
   This is a very generic query, where all major search engines return many similar results which often confuse the users. Our system helped the users belonging to the second group, because they could read the comments posted by authoritative users in the first group. Moreover, the users of the first group had already filtered some of the unsatisfactory results given by commodity search engines.

5.3 Ranking users

Our ranking scheme assigns an order to both bookmark items and users. These two types of subjects play a similar role to the one played by news items and news source in our preceding work [1]. The experimental results observed for the news dataset in [1] have been confirmed on our social bookmark dataset. This dataset consist of 7845 bookmark items submitted by 13 users during a period of 3 months.

A first group of experiments addressed the sensitivity at changes of the parameters ρ and β. As a measure of concordance between the ranks produced with different values of the parameters, we adopted the well known Spearman [10] and Kendall-Tau [5] correlations.

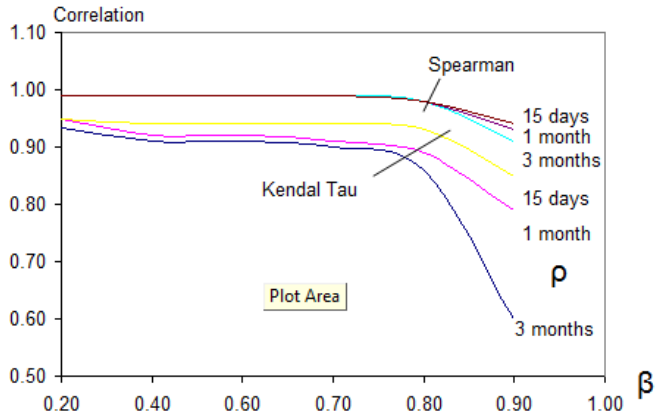

Figure 8: Correlations between ranks of users obtained with two successive values of β differing for 0.1. The solid lines are the Kendall-Tau measures, the dashed lines are the Spearman correlation coefficients.

We report the ranks computed with ranking schema TA3 [5], for values of $\beta_i = \frac{i}{10}$, where $i = 1, 2, \ldots, 9$ and for $\rho = 15$ days, 1 month and 3 months. In Figure 8, for a fixed ρ the abscissa β_i represents the correlation between the ranks obtained with values β_i and β_{i-1}. As in [1], we observe that the ranking scheme is not much sensitive to changing in the parameters involved.

5.4 Ranking bookmark items

In this section we evaluate how the scheme discussed in Section 4.1 improves the ranking of bookmark items over traditional `TF x IDF` scheme. In particular, we computed the ranking obtained with the Equation 4.1.2 for different values of γ, and on a set of 21 queries.

For each query, we evaluated the precision at the first N items generated. Precision at top N is defined as: $P@N = \frac{M@N}{N}$ where $M@N$ is the number of search results which have been manually tagged relevant among the N top-level labels computed by JammingNet. We believe that $P@N$ reflects the natural user behavior of considering the top-level results. We use $P@3$, $P@5$, $P@7$ and $P@10$ since lazy users do not like to browse many pages of search results. In Figure 9 we report the average P@N for different values of γ. The best results are obtained for $\gamma = 0.6$.

6. CONCLUSION AND FUTURE WORK

[5]We invite the interested reader to refer [1] for its definition

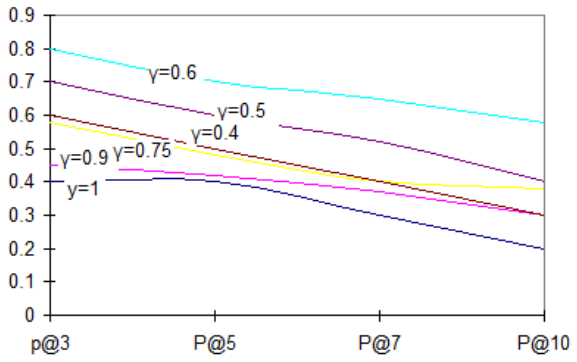

Figure 9: Average P@N for different values of γ.

6 7

In this paper, we propose JammingSearch, a novel model that unifies Web search, together with Wiki based content production and social bookmarking.

8

9 10

7. ACKNOWLEDGMENTS

We would like thank Sara Folegnani, Tony Savona, and Alessio Signorini for useful discussions.

8. REFERENCES

[1] G. M. D. Corso, A. Gullí, and F. Romani. Ranking a stream of news. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 97–106, New York, NY, USA, 2005. ACM.

[2] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning*. Methods Cambridge University Press, 2000.

[3] A. Gulli and A. Signorini. The indexable web is more than 11.5 billion pages. In *Proceedings of 14th International World Wide Web Conference*, pages 902–903, Chiba, Japan, 2005.

[4] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. BibSonomy: A social bookmark and publication sharing system. In *Proceedings of the First Conceptual Structures Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures*, pages 87–102, 2006.

[5] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 430:81Ű93, 1938.

[6] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[7] B. Krause, A. Hotho, and G. Stumme. A comparison of social bookmarking with traditional search. *Lecture Notes in Computer Science, Advances in Information Retrieval*, 4956:101–113, 2008.

[8] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.

[9] W. Scott and S. Padhraic. Algorithms for estimating relative importance in networks. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 266–275. ACM Press, 2003.

[10] C. Spearman. The proof and measurement of association between two things. *American Journal of Psychology,*, 15(72Ű101), 1904.

[11] Y. Yanbe, A. Jatowt, S. Nakamura, and K. Tanaka. Can social bookmarking enhance search in the web? In *JCDL '07: Proceedings of the 7th ACM/IEEE joint conference on Digital libraries*, pages 107–116, New York, NY, USA, 2007. ACM.

[12] C. M. A. Yeung, N. Gibbins, and N. Shadbolt. Web search disambiguation by collaborative tagging. *Workshop on Exploring Semantic Annotations in Information Retrieval at ECIR'08*, 2008.

[13] `http://www.infospaceinc.com/onlineprod/Overlap-DifferentEnginesDifferentResults.pdf`.

[14] `http://www.mediawiki.org/wiki/MediaWiki`.

[15] `http://www.nielsen-netratings.com/pr/pr_080515.pdf`.

[16] `http://www.pewinternet.org/pdfs/PIP_Searchengine_users.pdf`.

[17] `http://en.wikipedia.org/wiki/List_of_social_software`.

[18] `http://www.wikimatrix.org/`.

[19] `http://search.wikia.com/wiki/Search_Wikia`.

[20] `http://weblogs.hitwise.com/leeann-prescott/2007/02/wikipedia_traffic_sources.html`.

[21] `http://meta.wikimedia.org/wiki/Wikimedia_servers`.

[22] `http://en.wikipedia.org/wiki/Wikipedia:Wikipedians`.

[6] AG: abbiamo indirizzato tutti i problemi dati in section 3.1 di social bookmark (spam, lazyness, clear definition of tagging model)?

[7] AG: abbiamo ripreso tutti i punti aperti dai related?

[8] Luca: It could be also interesting to extend JammingSearch the other way around. You can have JammingSearch working in "Daemon mode" and listening for your queries to major search engines (of course you can start/stop the daemon mode). Once the daemon finds in the JammingPedia a recently posted/highly ranked post tagged with the same query you have just performed, it notifies you "dude, maybe there's something noteworthy for you in the JammingPedia". This ways is easier to test the JammingSearch effective improvement to the user experience.

[9] personalized time-aware variant of hits

[10] extend the ranking model to include user-user clustering, leveraging user groups