Challenges in Searching Online Communities

Sihem Amer Yahia
Yahoo! Research

Michael Benedikt
Oxford University

Philip Bohannon
Yahoo! Research

Abstract

An ever-growing number of users participate in online communities such as Flickr, del.icio.us , and YouTube , making friends and sharing content. Users come to these sites to find out about general trends – the most popular tags, or the most recently tagged item – as well as for more specific information, such as the recent posts of one of their friends. While these activities correspond to different user needs, they all can be seen as the filtering of resources in communities by various search criteria. We provide a survey of these search tasks and discuss the challenges in their efficient and effective evaluation.

1 Introduction

Online communities such as LinkedIn, Friendster, and Orkut attract millions of users who build networks of their contacts and utilize them for social and professional purposes. Recently, online *content* sites such as Flickr, del.icio.us, and YouTube have begun to draw large numbers of users who contribute content – photos, urls, text and videos. They also annotate the content: tagging it with appropriate keywords, rating it, and commenting on it. A key feature distinguishing these sites from previous content-management sites is the effective integration of the user's social network into the experience of exploring and tagging content. Similarly, some of the most popular online communities such as MySpace and Facebook encourage content-sharing as well as contact-making. As a result, a variety of popular online communities have a rich body of data comprised of user-contributed content, user relationships, and user ratings. We call a Web site supporting such a community a *social content* site.

The functionality of social content sites is based on data generated by users. Users spend their time browsing content and using keyword search to look for interesting content, people who share their tastes, and content posted by like-minded people. Hotlists of new/popular content, keywords, or recommendations may also be offered to users. In all these cases, the user is presented with lists of ranked content. It is critical that the ranking of results has the ability to leverage all the user-generated content and social connection information.

However, ranking of search results over the data on a social content site is far from trivial. First, search needs to take into account *social activity*. For example, if the user types "sunset" on a social photo-sharing site, a social-aware search should account in the ranking of results the rating of the photo by users, the tags of the photo, and potentially for each tag the status or reputation of the tagger – is this person's own photos tagged or endorsed? Second, search results need to be *personalized* based on *the user's context*. The user's contributions include many implicit and explicit indicators of interest – tagging, rating, and browsing activities; friendships, and the activities of friends. While in traditional Web search one *may* wish to utilize user information to enhance

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

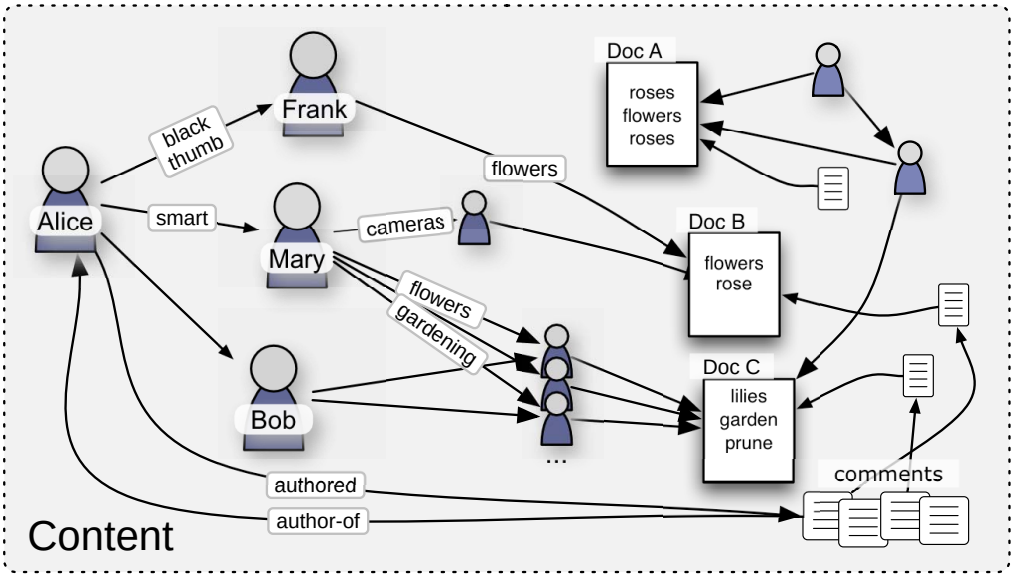

Figure 1: Gardening Social Content Example

search effectiveness, in social content search this information is readily available and essential for meeting the querier's expectation. For example, if the user searches for "birthday party" on the same photo-sharing site, it is reasonable to boost the rank of results from within the querier's social circle. Finally, when displaying hotlists of content or keywords, *recency* is often an important factor, requiring dynamic incorporation of new content in a manner similar to news search (e.g. [New]). Our first contribution is a classification of ranking factors in social search and illustrated with examples drawn from existing social content sites.

Given a particular ranking method, the next critical issue is the *efficiency* with which results can be computed. Obviously, techniques used for Web search are scalable, and have dealt successfully with the astronomic growth of the traditional Web. However, it is far from obvious that social intent, personalization, and recency can be incorporated into search without sacrificing efficiency. We discuss efficient and effective search in Section 3. We conclude and discuss some future challenges in Section 4.

2 Workload and Relevance Factors

In this section, we describe the *relevance factors* that tend to be operative in social content search. We then survey some of the functionality of existing social content Web sites in terms of these factors. In general, we consider three kinds of search targets: *content*, hot keywords, and *people* (usually called *expertise* search [MA00, KSS97]). We will use the term *resource* to refer to any of these search targets.

To illustrate the issues involved in relevance computation, we consider an example fragment of a hypothetical social-content Web site concerned with gardening, displayed in Figure 1. In this example, there are users and two kinds of content, blog posts and comments on those posts. Users can establish links with each other, and may assign a label to their relationship. For example, there is such a link between Alice and Frank. Note that this link is tagged with "black thumb", indicating that Alice has a low opinion of Frank's gardening skills. Users can tag content (dashed arrows), as Frank has tagged Document B with "flowers". Content can also refer to other content via hyperlinks (solid arrows). The right hand side of the figure shows examples of comments referring to documents, with these documents referring to the three named documents, A, B and C.

Whether the user is navigating to a hotlist of resources, browsing another user page or performing a keyword

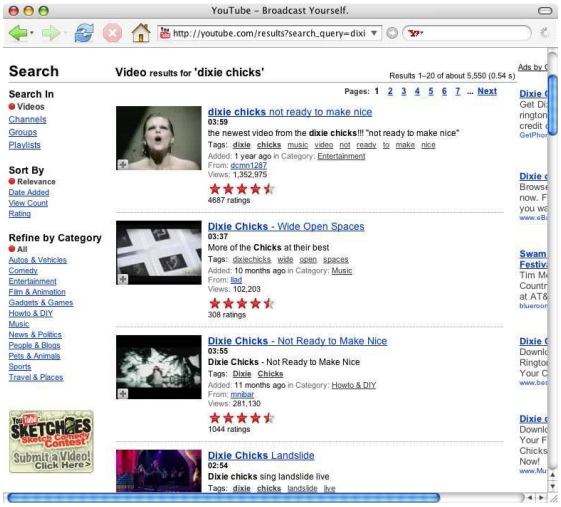

Figure 2: YouTube

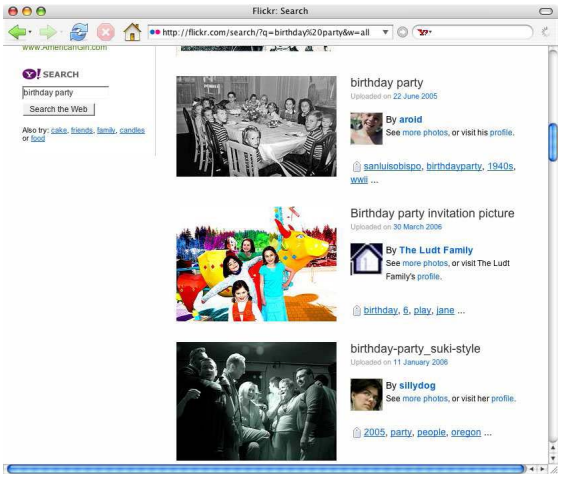

Figure 3: Flickr "Birthday Party" Search

search, the goal is the same: to return a list of resources ranked by score. The score of a resource can be informally defined using any subset of the following factors:

Text Features. When a keyword query is posed, the text associated with any content resource can be scored using standard metrics of relevance such as TF-IDF [SM83]. For multimedia content, there is often a title and description associated with it that can be matched against the keywords. For example, in Figure 1, Document A has content "roses" and "flowers", with "rose" repeated to show a high TF-IDF score.

Timeliness and Freshness. A resource may be more interesting if it is itself recently added. In the case of content , a simple interpretation of timeliness is as the inverse of the time elapsed since it was posted. One can also measure the timeliness via the popularity of the text associated with the resource – whether or not it is tagged with "hot" keywords.

Incoming Links and Tags. Tags associated with resources are usually a strong indicator of meaning. The anchor text on hyperlinks plays a role analogous to tags. We refer to either kind of link as an *endorsement*. Of course, PageRank-style metrics can be used to measure transitive endorsement - we discuss a variety of different ways such metrics can be computed and used below.

Popularity. A more subtle interpretation of timeliness may consider second-order recency or "buzz" - how much recent tagging and linking activity has targeted a resource? For example, if the references to document A from users on the right had been established in the last hour, A might be considered "hot". Further, measures of how many times an object is viewed may be incorporated in ranking.

Social Distance. A content resource can be considered "socially close" to the querier if it is tagged by individuals in the querier's social network. For example, a restaurant tagged by an acquaintance or by a reviewer that the querier has listed as a friend may be more interesting, since the querier may trust (or at least understand) the recommender's taste. Social distance can be computed by associating a score with each inter-personal link, and using the weight of paths from the querier to some target content. In Figure 1, there is a social connection from Alice, who might issue a query like "flowers", to document B of only two links through Frank, but the initial edge to Frank may have a low weight due to the tag. The paths from Alice to Document C are longer, but more paths exist, and the initial weights of the edges to Mary and Bob are stronger. Such factors must be balanced when estimating social distance.

Relevance for People. In the case a resource is a person, the documents authored by the person and the

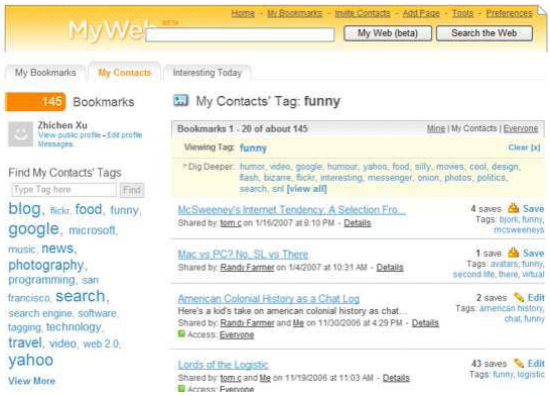

Figure 4: MyWeb Humor Search

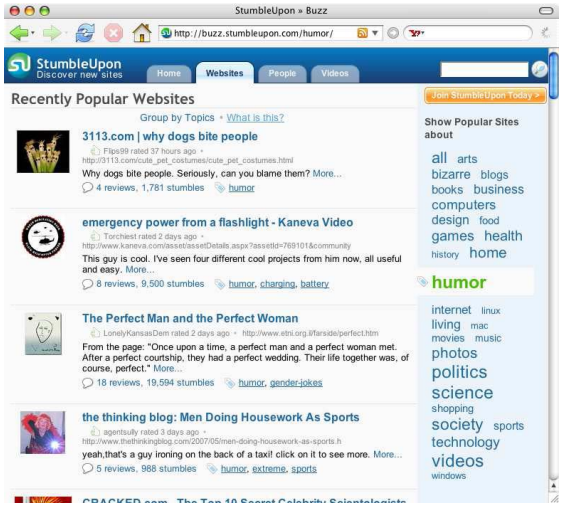

Figure 5: StumbleUpon Web List

outgoing tags can be considered as indicators of the person's *interests*, and thus can play a role analogous to text features of a content resource. For example, the comment authored by Alice may serve as a indicator for her. Inward links can also be important. For example, a person whose blog posts are well-reviewed or which are frequently tagged with words related to gardening might be considered an *expert* in the topic, and new posts by this individual about gardening might thus receive a higher rank.

We now give examples of social search in existing sites and relate them to the relevance factors discussed above.

Examples from Social Sites. One task is certainly keyword search. A straight keyword query can be used to search videos in YouTube – as in Figure 2, which shows the result of the query "dixie chicks". Clearly, ranking should take into account text features such as title and description, along with incoming tags applied to videos weighted by the popularity of the tags. One could envision adding a user's social activity, e.g., to rank videos tagged by friends higher. For blog posts, recency and popularity can be combined with text features [BK07].

Another typical task is browsing content resources – by tag, or by user. Even in browsing, ranking is important, since a given user or tag may have a large quantity of associated contributed content. For example, a user may browse photos related to the tag "birthday party" in Flickr, arriving at the result page shown on Figure 3. It may well be that the user will be more satisfied with photos of recent birthday parties by friends, in which case social distance needs to be accounted for. In some sites, that choice is left to the user - for example MyWeb gives the choice of searching "the Web" or "my contacts" (friendship network). Figure 4, shows the result of searching for "humor" over resources from a user's network. Note that the number of views and saves from the user's network are overlaid with each answer.

Finally, the search may not have keywords at all. That is, content may be *recommended* to the user (see, for example, [HKTR04]). StumbleUpon recommends "hot" resources of the moment (Figure 5), with a fresh list provided each time the page re-loads. The intention of a hotlist may be to show something popular, or to show something that is interesting based on past user clicks. It may even be simply trying to show the querier something in order to test the quality of an unrated content resource.

From these examples, we see clearly that the relevance factors defined in the previous section can be combined in a number of useful ways to power real search functionality in modern social-content sites. In the next section we discuss possible techniques and a number of open challenges in implementing this range of features into effective and efficient search functionality on social content sites.

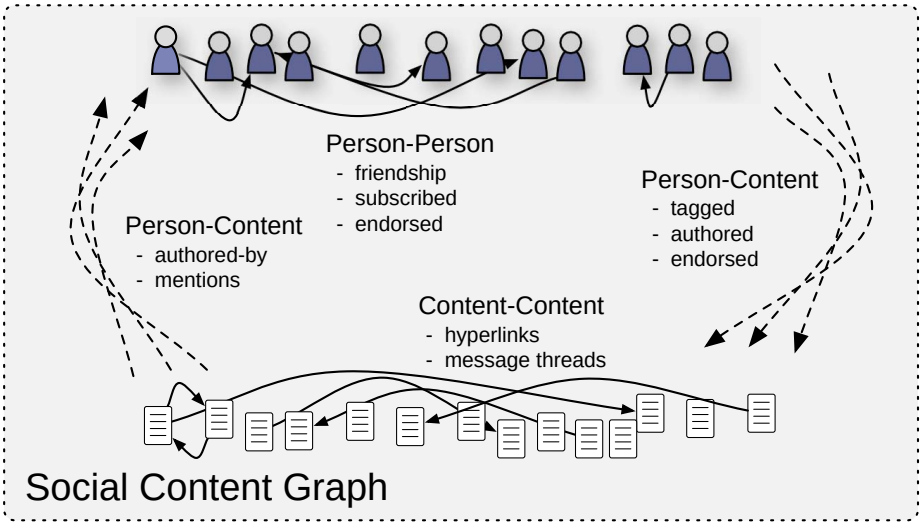

Figure 6: The Social Content Graph

3 Search Efficiency and Effectiveness

In this section, we outline techniques applicable to ranking search results in a social-content graph.

3.1 Integrated Approach

In attempting to capture the interactions between content linking and endorsement from friends, it is natural to treat resource endorsements, friendship links and people endorsements of content uniformly as edges in a "social-content graph". One such integrated approach is to model the relevance requirements of social content sites by parameterizing the behavior of a "random surfer" within a social-content graph, applying variants of PageRank [BP98] or HITS [Kle99] to compute the relevance of person or content nodes to a user's query. We now discuss the elements of this approach.

The Social-Content Graph An example of the directed graph that underlies this approach, called a "social-content graph," is shown in Figure 1, and the general form of such graphs is shown in Figure 6. A social-content graph has two types of nodes, corresponding to people and content (text, photos, videos, etc.). Edges may have associated text (e.g. tags). The semantics of edges in the graph depends on the type of the source and target node. Person-to-Person edges represent endorsement, friendship, or some other social relationship. The text of the edge may come from an explicit tag or the category of the relationship (e.g. "family", "coworker"). The Person-to-Content edges capture tagging, authoring, etc. Text associated with these links is derived naturally from tags. Content-to-Content edges may be hyperlinks, or may represent threading relationships between comments. Text associated with the link may be taken from the anchor text of the hyperlink. Finally, the Content-to-Person edges may show authorship or reference. For example, a search engine might use named-entity recognition (see, for example, [ZPZ04]) to identify references in picture titles to people's names, and establish Content-to-Person links from the picture to the person. Examples of each type of edge can be seen in the example of Figure 1.

Transition Probabilities We now consider how to model a "random surfer"[BP98, Kle99] traversing a social content graph. In this framework, a surfer begins at an arbitrary node in the graph. At each step, the probability of jumping rather than following an outgoing edge is called the *damping factor*. If the user decides to jump, she goes to any node according to its *node weight*. If the surfer follows a edge, then it picks any particular outgoing

edge with the probability proportional to an *edge weight*. Modifying the *node weight* based on user preference or on the keywords of the search leads to a *personalized* or *topical* PageRank computation [JW03, Hav02], while modifying the edge and node weights together to reflect the match of content with a query is performed in "intelligent surfing" [RD02]. Note that the *damping factor* controls the *locality* of each "surf".

From Probabilities to Ranking Given a social-content graph, a querier and a (possibly empty) set of keywords, an integrated approach to relevance computation proceeds as follows. First the system assigns node and edge-weights to be used for random surfing to nodes and edges in the social content graph. Second, the stationary probability that a surfer arrives at each node is computed (e.g. a PageRank computation). Third, the k nodes with the highest such probability are returned to the user.

From Relevance Factors to Probabilities We now briefly describe how each of the relevance factors discussed in the previous section might be handled by adjusting the parameters of the computation. First, *text features* can be handled by computing the query relevance of the text associated with a resource to the query terms and setting *node weights* proportional to this relevance. To handle *timeliness and freshness* as well as *popularity*, edges and node weights can be adjusted by their recency [BBVW06]. To handle *incoming links and tags*, the edge-weights can be adjusted to reflect relevance of the query to the tag or other text associated with the edge in the social-content graph (see [BWF+07]). Finally, *social distance* is incorporated naturally in this model by applying a significant fraction of the total node weight to the querier's node and adjusting Person-to-Person edge weights according to the strength of the connection. While node and edge weights can be set individually, coarser parameters will make these easier to manage. For example, it may be helpful to adjust the overall weight of person nodes vs. content nodes for random jumps or of "friends" vs. "family" edges when following a link.

Feasibility and Performance While the integrated approach is conceptually clean and general, there are substantial feasibility and performance issues. As with all variants of PageRank, the stationary probability can be calculated using a fix-point algorithm. But given that the probabilities are only known at query-time, one cannot compute this off-line. One response to this problem is to use random surfing for only a subset of the features, relying on existing Web techniques for the remainder. For example, if one excludes hyperlink structure, one arrives at a graph that is considerably smaller than Web graphs. This *modular* approach is discussed in the next section. Another route is to apply recent work on accelerating dynamic PageRank [FR04, Cha07]. A technique to address the social-distance component is to approximate [CGS04] PageRank values only for nodes in the neighborhood of the querier. However, this approach is complicated by the fact that social graphs obey a so-called "power-law" distribution [WS98, New00, WF94], meaning that individual neighborhoods may be relatively large on average.

3.2 Modular Approach

The integrated approach gives the possibility of accounting for social distance in a very fine-grained way. It can account for the propagation of authority from a socially-close user through resource hyperlinks. But it certainly lacks modularity, since it cannot exploit the components already developed for Web search, particularly in the area of content and page-quality scoring.

A more modular but coarser approach is to consider each factor (or a subset of the factors) in isolation, coming up with separate rankings that are averaged to get one final score. Consider a case where a resource is content, the query-relevance could be done by traditional TF-IDF scoring of the content, and the resource endorsements could be ranked via PageRank or some other link analysis method. This does not eliminate the issue of incorporating social distance, but it does reduce it to two main components. The first is calculating a social endorsement score, which should average the impact of query-matching tags from friends in a user's network; The second issue is the overall combination problem, which should result in a single score formed by combining component scores over the various dimensions e.g., by taking a weighted average. The second problem revolves around the choice of weights, which we discuss in Section 3.4.

The gain in modularity in this approach is counter-balanced by a possible loss in effectiveness, since each factor is now considered in isolation. Consider a page resource that is not itself tagged by many in the user's community, but which is linked to many pages that are tagged by many in the community: such a resource might score low both in link-quality and tagging quality, although it is likely to be quite relevant.

3.3 Computing Social Endorsement

To see the daunting efficiency issues that remain in the calculation of the social endorsement score, consider a simple example where the query is a keyword search and the social endorsement score of a resource is computed as the number of times friends of the querier have tagged the resource with query-matching tags. One must consider two related issues here: what sort of indexing structure is available for the tagging information, and the query evaluation algorithm. The most natural approach is to organize data in inverted lists by tag, in direct analogy with what is done for terms in a standard IR setting. Each entry in the list records a resource identifier and also its list of taggers.

Given a query, the score of a resource could be computed as the number of its taggers who are friends of the querier, or as the sum of all the people in its connected component, weighted by social distance. We can thus see the social endorsement score as another instance of combining the rankings of different lists, where the "combining" here requires revising the scores in each list based on personalization. The difference from standard rank combination problems (see e.g. [Fag02]) is that exactly which users in the list contribute to the score is dependent on the querier. Standard algorithms for combining scores rely on the sorting of the inverted lists by static upper-bound scores. This is the case, for example, in the family of Threshold Algorithms [Fag02]. In the case of personalized scoring, it is not clear what kind of upper-bound could be used – a global (e.g., querier-independent) upper-bound would be too coarse since it overlooks the difference in behavior between users. A possible solution is to devise an adaptive algorithm that discovers friendships and resource endorsements during query evaluation and uses them to refine upper-bounds.

3.4 Refining Scores by Clustering

The social endorsement described in the previous section takes as given the fact that the score of a resource for a given query should depend upon the tagging activity of the members of a querier's explicitly-recognized community. The integrated approach allows the score to depend transitively on the impact of tagging, friendship links, resource hyperlinks, and resource content matching, but is still based on the notion of community given by explicit friendship information. One may be interested in using derived notions of affinity between users, creating either links between users or clusters of users based on common behavior. Derived links can be used as a substitute for explicit links in either an integrated or modular approach. User clusters can also play a role in gaining effectiveness by getting more personalized versions of algorithm parameters. In the integrated approach, this would mean replacing the various global damping parameters with multiple per-cluster weights. User clusters would also naturally fit in a modular approach, replacing the social-endorsement score by a per-cluster or per-term/cluster endorsement score for each resource. The definition of user clusters would help refine score upper-bounds. Since a querier only belongs to one cluster, the refined upper-bound of the cluster can be used for more effective pruning than a single score upper-bound per resource. The question of how many user clusters should be defined remains open.

Due to the large number of features of users, a possible approach to deriving user clusters is via machine learning. In this setting, training data corresponds to labeled resources that can be either inferred from click logs [XZC+04] or requested explicitly from users. The idea would be to cluster users based on their click behavior or on their explicit feedback on ranked results.

A promising aspect of a machine-learning approach is that it can exploit the feedback mechanisms already present in these sites to generate a significant amount of high-quality training data. We can easily imagine

allowing end-users to evaluate rankings at query time. This would be particularly appealing to users in the context of online communities, due to their already-active participation in evaluating content. In particular, by making it "easy" for users to mark a resource as good or bad, the user is only one click a way from providing that information, per resource. This is already enabled, in a limited way, by some systems such as the thumbs up/thumbs down feature in StumbleUpon [stu].

4 Conclusion

Social-content websites depend fundamentally on search functionality for their core value proposition. In this paper, we have outlined the relevance factors that must be combined for effective social-content search. We have presented an *integrated approach* in which any subset of these relevance factors can be mapped into a "random surfer" model and relevance calculated with a PageRank computation. In the face of feasibility and performance challenges with this technique, we have discussed the difficulties faced in adapting more traditional relevance computation techniques to the requirements of social-content search. The development of efficient search techniques capable of effective search in this domain is an important problem, just starting to be addressed by recent work [SBBW07, BWF+07, Cha07].

One challenge to such research is evaluating quality. Accepted standards of search quality typically involve carefully annotated example sets [tre, ine]. Providing the assessments on which these metrics are based is a tedious task, but one which is nevertheless necessary. Unfortunately, this task is even more difficult to implement in the context of personalized search in social content sites. To our knowledge, there is still no principled way to evaluate the quality of proposed relevance algorithms for search involving personalization or social distance. Research papers such as [SBBW07, BWF+07] rely on manual assessments done by individuals (e.g., paper authors and students). An important issue is how to compare quality of social ranking techniques across applications and research groups.

In the context of actual systems, however, there is a great potential for users to provide feedback to the system, since expressing opinions on topics of interest may be, along with finding interesting content, a key motivation for users visiting social content sites. One avenue to explore is the incorporation of techniques and interfaces built for collaborative filtering (see, for example, [HKTR04, KSS97]) to collect feedback from users, and thus evolve and tune relevance functions over time. A key issue in this process is how to *cluster* users to efficiently predict preferences across a variety of topics and content types.

References

[BBVW06] Klaus Berberich, Srikanta Bedathur, Michalis Vazirgiannis, and Gerhard Weikum. BuzzRank ... and the Trend is Your Friend. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, 2006.

[BK07] Nilesh Bansal and Nick Koudas. Searching the Blogosphere. In *10th International Workshop on the Web and Databases (WebDB 2007)*, 2007.

[BP98] Sergey Brin and Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks*, 30:107–117, 1998.

[BWF+07] Shenghua Bao, Xiaoyuan Wu, Ben Fei, Guirong Xue, Zhong Su, and Yong Yu. Optimizing Web Search using Social Annotations. In *WWW '07: Proceedings of the 12th International World Wide Web Conference*, New York, NY, USA, 2007. ACM Press.

[CGS04] Yen-Yu Chen, Qingqing Gan, and Torsten Suel. Local Methods for Estimating PageRank Values. In *CIKM '04: Proceedings of the 13th ACM International Conference on Information and Knowledge Management*, 2004.

[Cha07] Soumen Chakrabarti. Dynamic Personalized Pagerank in Entity-relation Graphs. In *WWW '07: Proceedings of the 16th International World Wide Web Conference*, pages 571–580, New York, NY, USA, 2007. ACM Press.

[Fag02] Ronald Fagin. Combining Fuzzy Information: an Overview. *SIGMOD Record*, 32(2):109–118, 2002.

[FR04] Dániel Fogaras and Balázs Rácz. Towards Scaling Fully Personalized PageRank. In *WAW '04*, volume 3243 of *LNCS*, pages 105–117, 2004.

[Hav02] Taher H. Haveliwala. Topic-sensitive PageRank. In *WWW '02: Proceedings of the 11th International World Wide Web Conference*, pages 517–526, New York, NY, USA, 2002. ACM Press.

[HKTR04] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[ine] Initiative for the Evaluation of XML Retrieval. http://inex.is.informatik.uni-duisburg.de/.

[JW03] Glen Jeh and Jennifer Widom. Scaling Personalized Web Search. In *WWW '03: Proceedings of the 12th International World Wide Web Conference*, pages 271–279, New York, NY, USA, 2003. ACM Press.

[Kle99] J. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46:604–632, 1999.

[KSS97] Henry Kautz, Bart Selman, and Mehul Shah. Referral Web: combining Social Networks and Collaborative Filtering. *Communications of the ACM*, 40(3):63–65, 1997.

[MA00] David W. McDonald and Mark S. Ackerman. Expertise Recommender: A Flexible Recommendation System and Architecture. In *CSCW '00: Proceedings of the 2000 ACM conference on Computer Supported Cooperative Work*, pages 231–240, New York, NY, USA, 2000. ACM Press.

[New] http://news.search.yahoo.com/.

[New00] M. E. J. Newman. Models of the Small World. *Journal of Statistical Physics*, 101(3-4):819–841, November 2000.

[RD02] Matthew Richardson and Pedro Domingos. The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. In *Advances in Neural Information Processing Systems 14*, 2002.

[SBBW07] J. Stoyanovich, S. Bedathur, K. Berberich, and G. Weikum. Entityauthority: Semantically enriched graph-based authority propagation. In *10th International Workshop on the Web and Databases (WebDB 2007)*, 2007.

[SM83] Gerald Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[stu] `http://www.stumbleupon.com`.

[tre] Text REtrieval Conference. http://trec.nist.gov.

[WF94] Stanley Wasserman and Katherine Faust. *Social Network Analysis*. Cambridge University Press, Cambridge, 1994.

[WS98] Duncan Watts and S. H. Strogatz. Collective Dynamics of "small-world" Networks. *Nature*, 393:440–442, 1998.

[XZC+04] Gui-Rong Xue, Hua-Jun Zeng, Zheng Chen, Yong Yu, Wei-Ying Ma, Wensi Xi, and Weiguo Fan. Optimizing Web Search Using Web Click-through Data. In *CIKM'04*, pages 118–126, New York, NY, USA, 2004. ACM Press.

[ZPZ04] Li Zhang, Yue Pan, and Tong Zhang. Focused Named Entity Recognition using Machine Learning. In *SIGIR '04: Proceedings of the 27th Annual International ACM SIGIR Conference*, 2004.