The EigenRumor Algorithm for Ranking Blogs

Ko Fujimura
NTT Cyber Solutions Laboratories
NTT Corporation

Takafumi Inoue
NTT Cyber Solutions Laboratories
NTT Corporation

Masayuki Sugisaki
NTT Resonant Inc.

ABSTRACT
The advent of easy to use blogging tools is increasing the number of bloggers leading to more diversity in the quality blogspace. The blog search technologies that help users to find "good" blogs are thus more and more important. This paper proposes a new algorithm called "EigenRumor" that scores each blog entry by weighting the hub and authority scores of the bloggers based on eigenvector calculations. This algorithm enables a higher score to be assigned to the blog entries submitted by a good blogger but not yet linked to by any other blogs based on acceptance of the blogger's prior work.

General Terms
Algorithms, Management, Experimentation

Keywords
Weblog, link-analysis, ranking, search engine.

1. INTRODUCTION
Many approaches on ranking Web pages have been proposed and studied[3]. PageRank[2] and HITS[7] are most successful of these and their effectiveness has been shown in both industry and the academic world. Of course these techniques are also effective for ranking blogs. The simple adoption of these algorithms to blogs, however, induces some issues as follows:

1) The number of links to a blog entry is generally very small. As the result, the scores of blog entries calculated by PageRank, for example, are generally too small to permit blog entries to be ranked by importance.

2) Generally, some time is needed to develop a number of in-links and thus have a higher PageRank score. Since blogs are considered to be a communication tool for discussing new topics, it is desirable to assign a higher score to an entry submitted by a blogger who has been received a lot of attention in the past, even if the entry itself has no in-links at first.

Considering these issues, this paper proposes a new link-analysis algorithm called "EigenRumor." The algorithm is designed for ranking information resources provided as blogs or other cyberspace communities, in which the identities of information providers are observable. Unlike generic web pages, a blog site is constructed from a set of blog entries written by a single blogger and the quality of blog entries and topics are dominated by the ability or interests of the blogger. Using this structural characteristic of blogs, the EigenRumor algorithm rates a new blog entry or other blog entries that have no in-links according to the past behavior of the blogger.

WWW 2005, May 10--14, 2005, Chiba, Japan.

In this paper, we define a *blog* (or *blog site*) from just the structure point of view, i.e., we do not concern ourselves with the contents of the blog. We assume that a blog has the following structure:

(a) A blog consists of a top page and a set of blog entries. A blog is generally updated and maintained by a single blogger.

(b) There are links from the top page of the blog to each blog entry and each blog entry has a permanent URI.

(c) Blog entries are frequently added and the notification of updates is, as an option, sent to a ping server [11].

(d) A mechanism to construct a trackback [10] is provided.

The EigenRumor algorithm has similarities to PageRank [2] and HITS [7] in that all are based on eigenvector calculation of the adjacency matrix of the links. In the EigenRumor model, however, the adjacency matrix is constructed from agent-to-object links, not page-to-page (or object-to-object) links. Note that an *agent* is used to represent an aspect of human being such as a blogger, and an *object* is used to represent any object such as a blog entity in this paper. Using the EigenRumor algorithm, the hub and authority scores are calculated as attributes of agents (bloggers) and by weighting these scores to the blog entries submitted by the blogger, the attractiveness of a blog entity that does not yet have any in-link submitted by the blogger can be estimated.

This paper also reports the implementation experiments of a blog search engine that returns the search result sorted by the scores calculated by this algorithm and evaluated the effectiveness of the ranking by submitting several queries. Our experience shows that links between blog entries are very sparse. Only 1.2% of blog entries have links to the blog entries of others. The aggregation on the agent (blogger) provided the EigenRumor algorithm enables us to assign non-zero scores to about 9.3% of blog entries. This greatly improves the usability of blog searches.

In Section 2, we discuss the classification of blog rankings and clarify the target of this paper. In Section 3, we present the EigenRumor algorithm that calculates the hub and authority scores for agents and the reputation score of objects. In Section 4, we describe how to apply the EigenRumor algorithm to blog ranking. In particular, we describe the normalization strategy of links to reduce the effect of search engine optimization (SEO) and so get better ranking. In Section 5, we briefly present an implementation for blog search engines and experiments learned from applying the system. Finally, we present related works and the conclusions in Sections 6 and 7, respectively.

2. BLOG RANKING
There are various types of ranks in the so-called "blog ranking" technology. In this section, we classify them and clarify the target of this paper. Although this is not

exhaustive, blog rankings are classified using the following axis:

(1) Subject of ranking
  (a) Blog entries
  (b) Bloggers
  (c) Articles referred to by blogs
  (d) Goods or services referred to by blogs
(2) Space of ranking
  (a) All blogs
  (b) Blogs that send notification of update to a specific ping server
  (c) Blogs in a specific provider
(3) Temporal space of ranking
  (a) All blogs
  (b) Specific period
  (c) Damping model
(4) Semantics of ranking
  (a) Strength of support from the community
  (b) Trustworthiness
  (c) Recency / freshness
  (d) Specific attribute, e.g., funniness or usefulness
(5) Source of evaluations collected
  (a) Hyperlink, e.g. trackbacks
  (b) Access, e.g., number of clicks
  (c) Collection of explicit votes
  (d) Natural language analysis

Regarding the subject of ranking (1), the target of this paper is both (a) and (b). We think that the ranking of goods or services referred to by blogs is important for marketing purposes. However, ranking blogger and blog entries is more important because if we have a reliable ranking of blogger or blog entries, we can then easily and reliably rank goods or services by weighting the reliability of the blogger or blog entries. This paper thus focuses on (a) and (b) as the first step.

Regarding the space of ranking (2), it is important from the viewpoints of business or implementation, but it has no, theoretically, impact, and we make no assumption regarding ranking space in this paper.

Regarding the temporal space of ranking (3), it is important to weight newer topics since blogs are usually used to find or discuss new topics. This paper thus presents a mechanism to support it.

Regarding the semantic of ranking (4), it depends on how the evaluations of blogs are collected, which is axis (5) above. At this moment, there is no mechanism to express the semantics and strength of support of resources that a blog refers to explicitly. Technorati [12] introduced a new attribute tag called "rel" to specify the category of link but this is not widely used yet. This paper thus collects evaluations of each blog entry by assuming that a link is an indication of *interest* in some aspect of the blog. Thus the semantics of ranking in this paper might be "attractiveness" rather than "strength of support" from the blog community.

3. THE ALGORITHM

The EigenRumor algorithm proposed here is a highly generic algorithm and applicable to not only blog communities but also any other cyberspace community in which the identities of information providers (agents) are observable, in other words, communities in which membership registration is required.

In this section, therefore, we describe the algorithm in an abstract manner and we use "agent" and "object" for "blogger" and "blog entry", respectively.

3.1 Community model

We assume a universe of m *agents* and n information *objects*. When agent i provides (posts) object j, a *provisioning link* is established from i to j. We will use the *provisioning matrix* $P=[p_{ij}]$ (i=1…m, j=1…n) to represent all provisioning links in the universe. In this notation, p_{ij}=1 if agent i provides object j and zero otherwise. When agent i evaluates the usefulness of an existing object j with the scoring value e_{ij}, an *evaluation link* is established from i to j. We will use the *evaluation matrix* $E=[e_{ij}]$ (i=1…m, j=1…n) to represent all evaluation links in the universe (Figure 1). The evaluation link is assigned weight e_{ij} based on the strength of the support given to object j. We assume e_{ij} has the range of [0,1] and higher values indicate stronger support. For simplicity, we do not consider negative values for e_{ij}.

Note that scoring value e_{ij} is not always given explicitly. It can be generated by a translation rule, e.g., e_{ij}=1 when an article (object j) receives a comment from an agent i, e_{ij}=0 otherwise. An example of a translation rule applied to blogspace is given in Section 5.

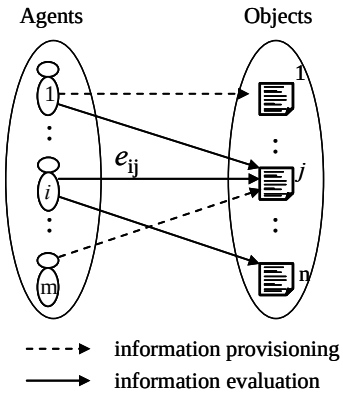

Figure 1. EigenRumor community model

3.2 Scores

The EigenRumor algorithm scores agents in two aspects: information evaluation (hub score) and information provisioning (authority score). These scores enable us to calculate the weighted score of an object.

To implement this idea, two scores for each agent and one score for each object are introduced in the algorithm:

Authority score (agent property)

This indicates to what level agent i provided objects in the past that followed the community direction. It is considered that the higher the score, the better the ability of the agent to provide objects to the community. We define $\vec{a}$ as a vector that contains the authority scores a_i for agent i (i=1…m).

Hub score (agent property)

This indicates to what level agent i submitted comments (evaluation) that followed the community direction on other past objects. It is considered that the higher the score, the better the

ability of the agent to contribute evaluations to the community. We define $\vec{h}$ as a vector that contains the hub scores h_i for agent i (i=1…m).

Reputation score (object property)

This indicates the level of support object j received from the agents, i.e., the degree to which j follows the community direction. It is considered that the higher the score, the better the object conforms to the community direction. We define $\vec{r}$ as a vector that contains the reputation score r_j (j=1…n) for object j.

3.3 The EigenRumor Alogrithm

The EigenRumor algorithm calculates three vectors, i.e., authority vector $\vec{a}$, hub vector $\vec{h}$, and reputation vector $\vec{r}$, defined in Section 3.2, from information provisioning matrix P and information evaluation matrix E, defined in Section 3.1.

Based on the following assumptions, these score vectors are mutually influenced:

Assumption 1: *The objects that are provided by a "good" authority will follow the direction of the community.*

Assumption 2: *The objects that are supported by a "good" hub will follow the direction of the community*.

Assumption 3: *The agents that provide objects that follow the community direction are "good" authorities of the community.*

Assumption 4: *The agents that evaluate objects that follow the community direction are "good" hubs of the community.*

Corresponding to the above assumptions, the algorithm introduces four equations as follows:

$$\vec{r} = P^T \vec{a} \quad \ldots(1)$$
$$\vec{r} = E^T \vec{h} \quad \ldots(2)$$
$$\vec{a} = P\vec{r} \quad \ldots(3)$$
$$\vec{h} = E\vec{r} \quad \ldots(4)$$

In order to merge equation (1) and (2) above, we use the following convex combination:

$$\vec{r} = \alpha P^T \vec{a} + (1-\alpha) E^T \vec{h} \quad \ldots(5)$$

where α is a constant with range of [0,1] that controls the weight of authority score and hub score. It is adjusted depending on the target community or application. Note that α can be assigned to each object separately and can be designed to decrease with time from the submission or the number of evaluations submitted to object j.

We now have three equations, (3), (4), and (5), that recursively define three score vectors, $\vec{a}$, $\vec{h}$, and $\vec{r}$. To find the "equilibrium" values for the score vectors, we integrate equation (3) and equation (4) with equation (5), and get:

$$\begin{aligned} \vec{r} &= \alpha P^T P \vec{r} + (1-\alpha) E^T E \vec{r} \\ &= (\alpha P^T P + (1-\alpha) E^T E)\vec{r} \quad \ldots(6) \\ &= S\vec{r} \end{aligned}$$

where

$$S = (\alpha P^T P + (1-\alpha) E^T E)$$

If S is a stochastic matrix, $\vec{r}$ will converge to the principal eigenvector of S simply by iterating procedure (6). Fortunately, the principal eigenvector of any non-negative matrix can be calculated by just adding a normalization procedure in each iteration procedure. In other words, we can get the equilibrium value for $\vec{r}$ such that that following equality is satisfied.

$$\lambda \vec{r} = S\vec{r} \quad \ldots(7)$$

where λ is the largest eigenvalue of matrix S. After getting $\vec{r}$, we can also get $\vec{a}$, $\vec{h}$ by equations (3) and (4). We can also get all of these scores simultaneously by the procedure shown in Figure 2.

$\vec{a}^{(0)} = (1,\ldots,1)^T$
$\vec{h}^{(0)} = (1,\ldots,1)^T$
while $\vec{r}$ *changes significantly do*
  $\vec{r}^{(k)} = \alpha P^T \vec{a}^{(k)} + (1-\alpha) E^T \vec{h}^{(k)}$
  $\vec{r}^{(k+1)} = \vec{r}^{(k)} / \| \vec{r}^{(k)} \|_2$
  $\vec{a}^{(k+1)} = P\vec{r}^{(k+1)}$
  $\vec{h}^{(k+1)} = E\vec{r}^{(k+1)}$
end while

Figure 2. The EigenRumor Algorithm

$\| . \|_2$ is function that computes the L_2 vector norm.

4. MAPPING TO BLOG COMMUNITY

There are several ways in which the EigenRumor community model described in Section 3.1 can be applied to the blog community. We applied the simplest mapping, shown in Figure 3, to the blog search system described in Section 5. As shown this figure, the links from the top page of the blog site to the blog entries are considered to be information provisioning links and links to blog entries in other blogs are considered to be information evaluation links.

In this mapping, the scoring value e_{ij} of each information evaluation link is 1 if there is a link and 0 otherwise, since no explicit scoring value is given. Note that this information evaluation link is actually an entry-to-entry hyperlink and no blogger-to-entry link exists. We use the translation rule to interpret actual entry-to-entry links as blogger-to-entry links. Note also that entry-to-entry hyperlinks are sometimes created by the trackback mechanism [10]. Our system deals with both normal hyperlinks and (forward) trackback links equally since both links are considered to be an indication of the interest of the blogger who cites the entry. On the contrary, the (backward) trackback links, i.e., automatically generated by the trackback protocol, are not considered to be an indication of interest of the blogger whose entries are referred to and often generated by spamming. We accordingly ignore these links.

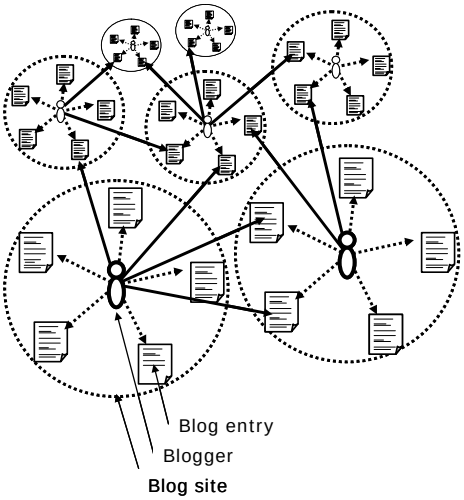

Figure 3. Mapping to blog community

Since the basic algorithm described in the previous section does not normalize information provisioning matrix P or information evaluation matrix E, it is susceptible to spamming. If some user creates many blog accounts and interlinks them, he/she can inflate the scores. To reduce the effect of this attack, normalization of the matrixes is important. PageRank [2] uses out-link normalization such that the total sum of out-links from one page is normalized to one. We have applied this method to the EigenRumor algorithm. It was found, however, that this approach does not work well for normalizing the links from agents (bloggers). Unlike web pages, the levels of activities of agents are quite diverse. Therefore, it is not fair to normalize total sum of out-links from one agent to one equally. Our experiments show that some blogs with only a few blog entries can earn the same level of authority scores as the blogs with a hundred of entries when we apply this normalization.

We also studied the behavior of scores in the case where no normalization is applied. In this case, it was also found that scores are seriously impacted by spamming as we expected.

The best normalization function we have found so far is to use the square root of the number of the objects submitted or evaluated by the agent, i.e.:

$$P' = [p'_{ij}]\,(i = 1...m, j = 1...n) \qquad p'_{ij} = \frac{1}{\sqrt{|P_i|}} \quad ...(7)$$

$$E' = [e'_{ij}]\,(i = 1...m, j = 1...n) \qquad e'_{ij} = \frac{1}{\sqrt{|E_i|}} \quad ...(8)$$

where $|P_i|$ and $|E_i|$ is the total number of objects provided and evaluated by agent i, respectively.

Generally, blogger interest in a specific blog entry submitted or cited decrease day by day. To implement this effect, we introduce an optional longevity factor to information provisioning links and information evaluation links, and we use the following $P^{(t)}$ and $E^{(t)}$ instead of P and E.

$$P^{(t)} = [p^{(t)}_{ij}] \qquad p^{(t)}_{ij} = \frac{\rho^{\,t-time(p_{ij})}}{\sqrt{\sum_{j=1...n} \rho^{\,t-time(p_{ij})}}} \quad ...(9)$$

$$E^{(t)} = [e^{(t)}_{ij}] \qquad e^{(t)}_{ij} = \frac{\gamma^{\,t-time(e_{ij})}}{\sqrt{\sum_{j=1...n} \gamma^{\,t-time(e_{ij})}}} \quad ...(10)$$

where t is the current time and $time(x)$ is the time when link x was created. ρ, γ are damping factors with range [0,1].

5. EXPERIMENTS

We implemented a blog search system that receives one or more keywords from the user terminal and returns a list of blog entries with the blog name as the search result.

In the database of the system, we stored about 9,280,000 entries from 305,000 blog sites collected by our crawler from October 16, 2004 to February 3, 2005. The collected data are mainly from 10 major blog providers in Japan and all of the entries are written in Japanese.

Of the 9,280,000 entries, 1,520,000 (16.3%) have one or more hyperlinks. Only 116,000 entries (1.25%) are linked to other blogs. Note that we distinguished whether the link is to a blog or not by checking whether the URI of the entry is also stored in the database. Therefore, the actual ratio of blog entries that are linked to other blogs is somewhat higher. Very few blog entries are referred to by other blogs, only 107,000 (1.15%). This means that only 1.15% of blog entries can be scored by PageRank if we use only this dataset. (The actual set is higher in number since there are some links from non-blog pages to the blogs in the database.) This ratio, 1.15%, seems too small to yield useful rank search results.

The EigenRumor algorithm solves the above problem since it assigns hub and authority scores to bloggers and then propagates these scores to all entries submitted by the blogger. As a result, 36,200 bloggers (blog sites) have at least one blog entry linked to (or from) other blogs and 28,300 bloggers have nonzero authority scores. This is 9.28 % of the 305,000 bloggers. These authority scores are propagated to their entries so 862,000 (9.28%) of blog entries have nonzero reputation scores. This ratio is still small but it is sufficient for ranking search results since the ranking is important the number of search results is large. Moreover, our observation shows that search engine users check only the top 20 search results.

We also investigated the effectiveness of the ranking by conducting a face-to-face user survey. We asked 40 guests who visited our exhibition held on February 2005, to use our blog search system. They were asked to compare the ranking quality with that of traditional blog ranking schemes, i.e., sorting by the number of in-links and TFIDF sorting [9]. The number of in-links directly counts toward the total number of links to all articles submitted by the agent. In this survey, all guests were asked to submit only one query that could be freely selected. We only interrupted when the guest submitted a query that had already been submitted. The blog search system showed the three rankings and the subjects were asked to indicate the best ranking. According to their replies, about 48% of queries showed no significant difference from the simple count of in-links. For 45% of the queries, the proposed scheme was superior while for about 7.5% of queries it was inferior (Table 1).

Table 1. The summary of user survey

Best result	EigenRumor	In-link	TFIDF	Not determined
Queries	18 (45%)	2 (5%)	1 (2.5%)	19 (48%)

In this experiment, we also found that if the query was generic such as "baseball," i.e. many search results are returned, there was no prominent difference between EigenRumor and In-link. However, in case of more specific queries such as "baseball ichiro" EigenRumor generally provided the better ranking. This is considered to indicate the effect of score aggregation on agents provided by the algorithm. It is also observed that simple in-link rankings are more susceptible to spamming in which blogs attempt to create several accounts and link them to each other to inflate the ratings. Actually, we often found such attacks in the rankings generated by the number of in-links. This type of attack is more prominent when we submit specific queries.

6. Related Works

Blog ranking is an important topic in web mining but it still has not been widely studied. Adar el al. [1] proposed the concept of ranking called iRank, which assigns high ratings to the sites that contain original (source) information whereas PageRank and EigenRumor assign high ratings to popular sites. In this sense, iRank and EigenRumor have different purposes. However, both approaches have similarities in addressing the issue of the sparseness of the blogspace and the importance of the dynamic structure of links. (We introduced a link longevity factor in Section 4).

Technorati [12] provided a commercial blog search and some similarities with our system appears to exist. However, details of the ranking algorithm were not published.

Access ranking is widely used in the blogspace, but it requires the bloggers or blog providers to participate in the ranking process and thus has a fundamental disadvantage in terms of limited coverage.

Apart from the area of the blogspace, the EigenRumor algorithm has a unique characteristic as a new link-analysis tool. Most link-analysis schemes proposed so far consider page-to-page links or agent-to-agent links [6]. On the contrary, the EigenRumor algorithm analyzes agent-to-object links directly and it dispenses with the need to collect agent-to-agent links. This widens the application field of link analysis.

The EigenRumor algorithm is based on eigenvector analysis similar to PageRank [2] and HITS [7] but it manages scores for agents and objects separately and reputation scores are introduced as well as hub and authority scores as illustrated in Figure 4. As a result, an object provided by an agent with high authority score can be ranked highly from the time submitted. This is impossible with PageRank or HITS which require many reviews before useful scores can be assigned. The normalization of link described in Section 4 is also a unique feature of the EigenRumor algorithm since this it allows the analysis of agent-to-object links and the levels of activities of agents are quite different from those of static web pages.

Authors have presented some related ranking algorithms [4][5], but none of them are based on eigenvector calculation or address blogspace-specific issues.

7. CONCLUSION

In this paper, we presented a new algorithm for ranking blogs and showed its effectiveness by calculating the score of 9,280,000 blog entries. The important feature of the algorithm is to widen the coverage of blog entries that are assigned a score by only from static link analysis. This feature is especially important for blog ranking since the link structure of blogspace is sparser than that of Web.

	PageRank	HITS	EigenRumor
Entities	Web page	Web page	Agent/Object
Link types	Evaluation (E)	Evaluation (E)	Evaluation (E) Provisioning (P)
Scores	Authority ($\vec{a}$)	Authority($\vec{a}$) Hub($\vec{h}$)	Authority($\vec{a}$) Hub($\vec{h}$) } Agent Reputation($\vec{r}$) Object
Algorithm	$\vec{a} = (\frac{d}{N}\mathbf{1}_N + (1-d)E^T)\vec{a}$	$\vec{h} = E\vec{a}$ $\vec{a} = E^T\vec{h}$	$\vec{r} = \alpha P^T\vec{a} + (1-\alpha)E^T\vec{h}$ $\vec{a} = P\vec{r}$ $\vec{h} = E\vec{r}$

Figure 4. Comparison with PageRank and HITS Algorithms

This approach also enables to assign a higher score when the blog entry is submitted by a blogger who has been accepted a lot of attention in the past, even if the entry itself has no in-links at first. This is a desirable feature of blog rankings since blogspace are considered to be a community in which discussing new topics.

Future work can be a new user interface or visualization of search results in which take advantage of the algorithm that calculates three scores, i.e., authority, hub, and reputation scores. More detail analysis on the durability of spamming is also an important future work.

8. ACKNOWLEDGEMENTS

We would like to thank Naoto Tanimoto, Yoshinobu Tonomura, and Masahiro Oku for helpful discussions and comments.

9. REFERENCES

[1] E. Adar, L. Zhang, L. Adamic, and R. Lukose, "Implicit Structure and the Dynamics of Blogspace," In *Proceedings of the Workshop on the Weblogging and Ecosystem at the 13th International World Wide Web Conference*, 2004.

[2] S. Brin and L. Page, "The Anatomy of a Large-scale Hypertextual Web Search Engine," In *Proceedings of 7th International World Wide Web Conference*, 1998.

[3] S. Chakrabarti, *mining the web*, Morgan Kaufmann Publishers, 2003.

[4] K. Fujimura and T. Nishihara, "Reputation Rating System based on Past Behavior of Evaluators," In *Proceedings of the 4th ACM Conference on Electronic Commerce*, 2003.

[5] K. Fujimura, N. Tanimoto, and M. Iguchi, "Calculating Contribution in Cyberspace Community Using Reputation System "RuMoR"", In *Proceedings of the AAMAS Workshop on Trust in Cyber-societies*, July 2004.

[6] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," In *Proceedings of 12th International World Wide Web Conference*, 2003.

[7] J. M. Kleinberg, "Authoritative sources in hyperlinked environment," *Journal of the ACM*, Vol. 46, No. 5, 1999.

[8] D. Libby, "RDF Site Summary (RSS) 0.9 official DTD," http://my.netscape.com/ publish/formats/ rss-0.9.dtd, 1999.

[9] C. D. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, MA 1999.

[10] B. and M. Trott, "Trackback Technical Specification, " http://www.sixapart.com/movabletype/docs/mttrackback, 2002.

[11] D. Winer, "Blog.Com XML-RPC interface," http://www.xmlrpc.com/weblogsCom, 2001.

[12] Technorati, Inc. www.technorati.com.