Reputation Systems for Open Collaboration*

B. Thomas Adler†
Facebook, Inc.
thumper@alumni.caltech.edu

Luca de Alfaro‡
UC Santa Cruz
luca@dealfaro.org

Ashutosh Kulshreshtha
Google, Inc.
ashu@google.com

Ian Pye†
CloudFlare, Inc.
ian@cloudflare.com

Content creation used to be an activity pursued either individually, or in closed circles of collaborators. Books, encyclopedias, map collections, had either a single author, or a group of authors who knew each other, and worked together; it was simply too difficult to coordinate the work of large, geographically dispersed groups of people when the main communication means were letters or telephone. The advent of the internet has changed all this: it is now possible for millions of people, from all around the world, to collaborate. The first open-collaboration systems, wikis, focused on text content; the range of content that can be created collaboratively has since expanded to include, for instance, video editing (e.g., MetaVid [5]), documents (e.g., Google Docs[1], ZOHO[2]), architectural sketching (e.g., Sketchup[3]), and geographical maps (e.g., OpenStreetMaps [10], Map Maker[4]).

Open collaboration carries immense promise, as shown by the success of Wikipedia, but also carries challenges both to content creators and to content consumers. At the content-creation end, contributors may be of varying ability and knowledge. Collaborative systems open to all will inevitably be subjected to spam, vandalism, and attempts to influence the information. How can systems be built so that constructive interaction is encouraged and the consequences of vandalism and spam are minimized? How can the construction of high-quality information be facilitated? At the content-consumption end, visitors are presented with the outcome of a complex collaboration process. The content may result from the weaving together of many contributions, whose authors are usually not known to the visitor, and may even be anonymous. The corollary of "anybody can contribute" is "anybody could have contributed it". How can users judge how much trust to put into the information they are presented?

Reputation systems can help with the above challenges, facilitating both content creation and content consumption. To support this claim, we describe the reputation systems we have built for two major collaborative applications: the writing of articles for the Wikipedia, and the editing of business locations on Google Maps.

We chose to describe these two systems because they have been designed for well-known cooperative systems, and because they represent in several ways opposite ends of a design spectrum. The Wikipedia reputation system *WikiTrust* relies on a chronological analysis of user contributions to articles, and meters positive or negative increments of reputation whenever a new contribution is performed. Users can obtain new identities at will, and there is no "ground truth" against which their contributions can be compared. The reputation mechanism can be explained in simple terms to the users, and it could be used to provide an incentive to provide good-quality contributions. The Maps system *Crowdsensus* compares the information provided by users on map business listings and computes both a likely reconstruction of the correct listing and a reputation value for each user. In contrast to WikiTrust, users have a stable identity in the system, and their contributions can be compared with the "ground truth" of the real world, if desired. The reputation system operates largely in the background, and works not chronologically, but by iteratively refining joint estimates of user reputations, and listing values.

Content-driven vs. user-driven reputation. The Wikipedia and Maps systems we describe are both *content-driven:* they rely on automated content analysis to derive the reputation of the users and content. In contrast, reputation systems such as the Ebay system for sellers and buyers, and the Amazon and NewEgg systems of product reviews and ratings, are *user-driven:* they are based on explicit user feedback and ratings.

Content-driven systems derive their feedback from an analysis of all interactions, and consequently, they get feedback from all users uniformly. In contrast, user-driven systems often suffer from selection bias, as users who are par-

*The authors like to sign their papers in alphabetical order; thus, the author order does not necessariy reflect the size of the contributions.

†Part of his work was performed while the author was at the University of California, Santa Cruz.

‡Part of his work was performed while the author was at Google, Inc.

[1]http://docs.google.com
[2]http://www.zoho.com
[3]http://sketchup.google.com
[4]http://www.google.com/mapmaker

Communications of the ACM, Volume 54, Issue 8, August 2011

ticularly happy or unhappy are more likely to provide feedback or ratings. Moreover, in user-driven sytems, users can do one thing and say another. Sellers and buyers may give each other high ratings simply to obtain high ratings in return, regardless of how satisfied they are with the transaction [7]. Content-driven reputation systems derive user feedback from user actions, and can be more resistant to manipulation [4].

The deployment of user-driven and content-driven reputation systems presents different challenges. The success of a user-driven system depends crucially on the availability of user feedback. Even for successful sites, establishing a community of dedicated users and accumulating sufficient high-quality feedback can take years. When useful feedback can be extracted automatically from user interactions and data, on the other hand, content-driven reputation systems can deliver results immediately.

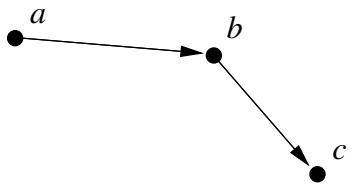
On the other hand, the algorithmic nature of content-driven reputation systems can play against their success, preventing users from understanding, and consequently trusting, the reputation values they generate. When a user reads: "Product A received 25 positive, 12 neutral, and 2 negative votes", the user understands the meaning of it, and often trusts to some extent the result — in spite of possible selection bias of voting users, and possible manipulation schemes by malicious users. In contrast, when an algorithm produces the answer for a Wikipedia page "this sentence has reputation 4 out of a maximum of 10", users typically wonder how the reputation is computed and question the appropriateness of the algorithms. In reputation systems that make reputation values available to users, simpler can be better even when the performance, in numerical terms, is worse: users need to understand the origin of reputation to be able to trust it [6, 7].

WikiTrust and Crowdsensus are just two examples of content-driven reputation systems. Other examples include systems that analyze the wording of consumer reviews to extract reviewer and product reputation [12, 15] and other approaches to Wikipedia content reputation [14]. The algorithms PageRank [13] and HITS [11] constitute content-driven reputation systems for ranking Web pages. Beyond the Web, consumer credit rating agencies are an example of content-driven reputation systems in the financial world.

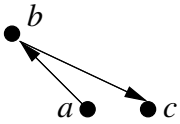
1. WIKITRUST

We present here the main ideas in WikiTrust[5], a reputation system for wiki authors and content. We developed WikiTrust with the goals of providing an incentive to give quality contributions to the Wikipedia, and offer Wikipedia visitors indications on the quality of content. To achieve these goals, WikiTrust employs two reputation systems: one for users, and one for content. Users gain reputation when they make edits that are preserved by subsequent authors, and lose reputation when their work is partially or wholly undone. Text starts with no reputation, and it gains reputation when it is revised by high-reputation authors; text can lose reputation when disturbed by edits. While WikiTrust was designed for wikis, its principles can be applied to any content management system in which the content evolves in a sequence of revisions, provided the difference between revisions can be somehow measured.

[5]http://www.wikitrust.net



(a) $d(a,c) > d(b,c)$, so that $q(b \mid a,c) > 0$.



(b) $d(a,c) < d(b,c)$, so that $q(b \mid a,c) < 0$.

Figure 2: A revision as in Figure 2(a), which bring b closer to c, is judged of positive quality; a revision as in Figure 2(b), which is largely reverted, is judged of negative quality.

WikiTrust is currently available via a Firefox browser extension. When a user visits a page of one of several Wikipedias, the browser extension displays an additional *WikiTrust* tab, alongside the standard wiki tabs such as *edit* and *history*. When users click on the WikiTrust tab, the extension contacts the back-end servers to obtain the text reputation information, which is visualized via the text background color: perfect-reputation text appears on a white background, and the background turns a darker shade of orange, as the reputation of the text lowers. The text coloring thus alerts viewers to content that might have been tampered, as illustrated in Figure 1. WikiTrust does not currently display user reputations, out of a desire not to alter the social experience of contributing to the Wikipedia.

User reputation. The reputation of users is computed according to the quality and quantity of contributions they make. A contribution is considered of good quality if the change it introduced is preserved in subsequent revisions [2, 8, 3]. To evaluate the quality of a contribution that produced a revision b, WikiTrust compares b with two reference points: a past revision a and a future revision c. From the point of view of c, if b is closer than a, then the author of b did good work, since she made changes that made the page more similar to how it will be in the future revision c (see Figure 2(a)). On the other hand, if b is farther away from c than a was, this means that the change from a to b was not preserved in c (see Figure 2(b)). To capture this intuition, we define the quality $q(b \mid a,c)$ of b with respect to a and c as the amount of improvement $d(a,c) - d(b,c)$ divided by the amount of work $d(a,b)$ involved in creating b. If the distance d satisfies the triangular inequality, we have that $q(b \mid a,c)$ is comprised between -1 and $+1$: it is equal to -1 if $a = c$ (so that the change $a \to r$ was entirely reverted), and it is equal to $+1$ if the change $a \to b$ was entirely preserved.

Authors start with a very small amount of reputation. When a new revision c is produced, it is used to judge the quality of several preceding revisions b, using as reference point revisions a that are either not too far in time from b and c, or are by high-reputation authors [4]. For each such triple considered, the reputation of the author of b is

Figure 1: The Wikipedia page for Don Knuth, as rendered by WikiTrust. The text background is a shade of orange that is the darker, the lower the reputation of the text.

increased by the amount $q(b|a,c) \cdot \log(1 + r_c)$, where r_c is the reputation of the author of c. The dependence of the increment on the reputation of c's author ensures that the judgement of higher-reputation authors carries more weight. A linear dependence would lead to an oligarchy in which long-time good users have an overwhelming influence over new users, while new users can give no significant feedback in return. Assigning all users the same influence would lead to a completely democratic system; this would not be ideal in wikis, as good users who entered in reversion wars with vandals would put their reputation too much at risk. The logarithmic factor balances oligarchy and democracy.

Users judge other users via their actions (their edits), and are thus liable to be judged in turn; this makes the system resistant to manipulation. For instance, the only way in which user A can damage the reputation of user B is by reverting user B's edits. However, if subsequent users reinstate B's edits, it will be A's reputation who will suffer the most, as B's contribution will prove to be longer-lived than A's.

When developing a reputation system, it is essential to be able to evaluate its performance quantitatively: otherwise, it is impossible to tune the system or compare different algorithms. A powerful evaluation criterion is the ability of user reputation to predict the quality of *future* user contributions [2]. On the one hand, this is a tough test to pass: it means that reputation is not only a badge gained via past work, but an indicator of future behavior. On the other hand, if low-reputation users were as likely as high-reputation users to do good work, why pay attention to user reputation?

Wikipedia	Precision	Recall
Dutch	58.1	95.6
English	58.0	77.1
French	43.7	89.1
German	50.4	93.4
Polish	43.1	91.7
Portuguese	48.3	94.1

Table 1: Predictive ability of the WikiTrust user reputation system on various Wikipedias. The table reports the precision and recall of low author reputation as a predictor for reversions.

To evaluate the user-reputation system of WikiTrust, we measured the precision and recall with which low-reputation can predict reversions [2]. For each revision b, we say that the author of b has low reputation of his reputation is in the bottom 20%, and that b has been reverted if its average quality is below -0.8. We note that this is a proper evaluation, since the reputation of b's author depends only on the past of b, whereas the quality of b depends on how b will be judged by future revisions. The results are reported in Table 1. The recall is high, indicating that high-reputation authors are unlikely to be reverted; the precision is lower because many novice authors make good-quality contributions. In measuring precision and recall, each contribution is weighed according to the number of words added and deleted. The data is based on Wikipedia dumps ending in late 2009, except for the English Wikipedia, where the dump is from

January 2008, and it has been augmented with updates until January 2010 for the 30,000 pages of the Wikipedia 0.7 project[6].

Content reputation. WikiTrust aims at providing an indication of the quality of Wikipedia content, in particular alerting visitors to possible vandalism and content tampering. To this end, WikiTrust computes and displays the *reputation* of Wikipedia content, at the granularity of individual words. An example is given in Figure 1, where the assertion on NP-completeness, having low reputation, has been highlighted in orange.

WikiTrust computes content reputation according to the extent to which the content has been revised, and according to the reputation of the users who revised it [14, 1]. When a new revision is created, the text that has been directly affected by the edit is assigned a small fraction of the revision author's reputation. Instead, the text that is left unchanged gains reputation: the idea is that the author, by leaving it unchanged, has implicitly expressed approval for it. The same idea can be applied to many types of content: all we need to do is to identify, when an edit occurs, which content is new or directly affected by the edit (this content will receive a fraction of the author's reputation), and which content has been left unaffected, and thus has been implicitly validated (this content may gain reputation).

WikiTrust adds to this idea some tweaks that make the content reputation system difficult to subvert. Since it is possible to alter the content of sentences not only by inserting new text, but also by re-arranging or deleting text, WikiTrust ensures that each of these actions leaves a low-reputation mark. Furthermore, the algorithm allows users to raise text reputation only up to their own reputation. Thus, low-reputation users cannot erase the low-reputation marks they leave behind with more activity. To ensure that a single high-reputation user gone rogue cannot raise arbitrarily the reputation of text via repeated edits, we associate with each individual word the identities of the last few users who raised the word's reputation, and we prevent users whose identity is associated with a word from again raising the word's reputation. The resulting content reputation system has the following properties:

- Content reputation is an indication of the extent to which the content has been revised, and of the reputation of the users who revised it.
- High content reputation requires consensus: it can only be achieved as a result of the approval of multiple distinct high-reputation users.

Evaluation. We use the predictive ability of the content reputation system as a measure of its performance. The idea is that higher-quality content should be less likely to be deleted in future revisions. This evaluation is imperfect, as it disregards the fact that our content reputation aims to have not only predictive value, but also warning value with respect to unrevised, possibly malicious edits. An analysis of 1000 articles selected at random among English Wikipedia articles with at least 200 revisions gave the following results [1]:

- *Recall of deletions.* Only 3.4% of the content is in the lower-half of the reputation range, yet this 3.4% corresponds to 66% of the text that is deleted from one revision to the next.
- *Precision of deletions.* Text in the lower half of the reputation range has a probability of 33% of being deleted in the very next revision, in contrast with the 1.9% probability for general text. The deletion probability raises to 62% for text in the bottom 20% of the reputation range.
- *Reputation as a predictor of content longevity.* Top-reputation words have an expected lifespan that is 4.5 times longer than words with bottom reputation.

A few lessons learned. WikiTrust has been available to the public for some time, and we have received much feedback from users.

The original reputation system described in [2] was open to many attacks that allowed users to gain reputation while doing no useful work (or worse, while damaging the system). For instance, under the original proposal a user could gain reputation by first vandalizing a revision using an alternate "sacrifical" identity and then undoing the vandalism using their main identity. As we believed that these attacks could have crippled the reputation system, we took pains to prevent them before making the system available [4]. Yet, neither the users, nor the researchers that provided us with feedback, showed any concern for the robustness of the original design, or appreciated our work to fix the weaknesses. We suspect that we would have been more successful by making WikiTrust available earlier, and dealing with the security issues only later, adopting the common (if hardly principled) approach of "security as an afterthought".

There was much interest, instead, in how we measure contribution quality. Early on in the development of the system, we realized that if we relied on a standard edit distance between revisions, users whose contributions were later reworded sometimes lost reputation, in spite of their good work. This was solved by adopting an edit distance that accounts for block moves, and that differentiates between word insertions and deletions, which are both given a weight of 1, and word *replacements,* which are given a weight of only $\frac{1}{2}$; under this edit distance, authors of reworded contributions still receive partial credit for their work. We were sure that our choice of edit distance would remain an obscure detail buried in the codebase. Instead, we found ourselves explaining it many times to Wikipedia contributors: users care deeply how their reputation is computed — even when the reputation is not displayed to anyone. Perceived fairness is a very important quality of a reputation system.

2. THE DESIGN SPACE

Table 2 summarizes the design space for reputation-systems for collaborative content. The first distinction has to do with the signals used for computing the reputation: are the signals derived from explicit user feedback, or are the signals inferred algorithmically from system events? Of course, the two types of systems can work side-by-side: for instance, sale and product return information could be used to compute NewEgg product ratings, and WikiTrust users have been recently given the possibility to vote explicitly for the correctness of Wikipedia revisions.

The second distinction concerns the visibility of the reputation system to the users. Many systems can be useful

[6]http://en.wikipedia.org/wiki/Wikipedia:Version_1.0_Editorial_Team

- **User-driven vs. content-driven.** User-driven reputation systems rely on ratings provided by users; content-driven systems rely on the algorithmic analysis of content and user interactions.
- **Visible to users?** Are users aware of the existence of the reputation system?
- **Weak vs. strong identity.** How easily can users acquire a new identity in the system?
- **Existence of ground truth.** Is there a "ground truth" to which we expect the content converges, if users were truthful?
- **Chronological vs. global reputation updates.** Chronological algorithms consider system activity in the order it occurs; global algorithms consider the whole system, and typically operate in batch mode.

Table 2: The design space for reputation systems for collaborative content.

even if they work "behind the curtains": such systems can be used to rank content, prevent abuse, fight spam, and more. Examples of such systems are web content ranking algorithms such as PageRank [13] or HITS [11]. Reputation systems that work behind the curtains can make use of any signals avaliable on users and content, and can use advanced algorithms and techniques such as machine learning. On the other hand, if the goal of the reputation system is to influence user behavior, its existence and the reputation values it computes need to be revealed to the users. In this case, it is important that the users can form some idea of how the reputation values are computed: people want to know the metrics used to judge them, and systems that cannot be understood are typically considered arbitrary, capricious, unfair, or downright evil.

The strength of the identity system is a relevant factor in the design of reputation systems. In systems with weak identity, new users must be assigned the same amount of reputation as bad users. There can be no "benefit of the doubt": if new users could enjoy a reputation above the minimum, bad users could simply start to use a new identity whenever their reputation fell below that of new users.

The next distinction concerns the existence of a "ground truth" to which content should correspond in order to have perfect quality. No such ground truth exists for Wikipedia articles: they do not converge to a canonical form as they are edited, but rather, they continually evolve as content is added and refined. In contrast, for Maps business listings such a ground truth exists for many information fields: for example, there is one (or a few) correct values for the telephone number of each business. As another example, in the Ebay seller rating system, it can be usefully assumed that each seller has an intrinsic "honesty"; buyer feedback is processed to estimate such honesty. This last example highlights how the existence of a ground truth matters not so much because we can check what the ground truth is (this is often expensive or impossible), but rather, because the *assumption* that a ground truth exists affects the type of algorithms that can be used.

Finally, reputation algorithms span a spectrum from *chronological* to *global.* At one extreme, purely chronological algorithms consider the stream of actions on the systems (contributions, comments, and so forth), and for each action they update the reputations of the participating users. The Ebay reputation system is chronological, and so is WikiTrust. At the other end of the spectrum are reputation systems based on global algorithms that operate at once on the whole network of recommendations, generally in batch mode. Each type of algorithm has advantages. Global algorithms can make use of the information in the graph topology: an example is the way in which PageRank or HITS propagate reputation along edges [13, 11]. Global algorithms, however, may require more computational resources, as they need to consider the whole system at once. Chronological algorithms can leverage the asymmetry between past and future to prevent attacks. In a chronological reputation system, new identities (including fake identities used for attacks) are assigned an initial reputation lower than that of established users. By making it difficult for users to gain reputation from users who are themselves of low reputation, WikiTrust is able to prevent many types of Sybil attacks [4].

3. CROWDSENSUS

To illustrate how the characteristics of the design space can influence the structure of a reputation system, we briefly overview *Crowdsensus,* a reputation system we built to analyze user edits to Google Maps. Users can edit business listings on Google Maps, providing values for the title, phone, website, address, location, and categories of business. The goal of Crowdsensus is to measure the accuracy of the users who contribute information, and to reconstruct insofar as possible correct listing information for the businesses.

The design space of a reputation system for editing Google Maps business listings differs in several respects from the design space of a Wikipedia reputation system.

First, for each business listing there is at least in first approximation a ground truth: ideally, each business has exactly one appropriate phone number, website, and so forth. Of course, the reality is more complex: there are businesses with multiple equivalent phone numbers, alternative websites, and so forth. Nevertheless, for the purposes of this article, we consider the simpler setting in which every listing attribute has exactly one correct value. We note also that it might be quite expensive to check the ground truth for each business listing: in the worst case, it might require sending someone on site! Crowdsensus does not require actually checking the ground truth: it simply relies on the *existence* of such a ground truth. Second, the user reputation is not visible to the users. Consequently, users need not understand the details of how reputation is computed, making it possible to use advanced algorithms and techniques. Third, the identity notion is stronger in Google Maps than on the Wikipedia. In particular, it is a practical nuisance for established users of Google products to open and use separate accounts for Maps editing. Fourth, the ample computational resources available at Google enable us to consider global reputation systems, in addition to chronological ones.

These considerations led to a design for Crowdsensus that is very different from the one of WikiTrust. The input to Crowdsensus consists in a sequence of *statements,* which are triples of the form (u, a, v), meaning: user u asserts that attribute a of some business has value v. Thus, Crowdsensus is set to solve what is called a *collective revelation prob-*

lem [9], even though some of the instruments by which such problems are solved, such as monetary payoffs, or elaborate ways of revealing a user's information, are not available in Crowdsensus. Crowdsensus is structured as a fixpoint graph algorithm; the vertices of the graph are the users and the business attributes. For each statement (u, a, v), we insert an edge from u to a labeled by v, and an edge from a back to u. Crowdsensus associates to each user vertex u a *truthfulness value* q_u, representing the probability that u is telling the truth about the values of attributes; this value is initially set to an a-priori default, and it is then estimated iteratively.

The computation of Crowdsensus is structured in a series of iterations. At the beginning of each iteration, user vertices send to the attributes their truthfulness value. Each attribute vertex thus receives the list $(q_1, v_1), \ldots, (q_n, v_n)$ consisting of the values $v_1, \ldots, v_n$ that have been proposed for the attribute, along with the (estimated) truthfulness $q_1, \ldots, q_n$ of the user who proposed them. An *attribute inference algorithm* is then used to derive a probability distribution[7] over the proposed values $v_1, \ldots, v_n$. Crowdsensus then sends to each user vertex u_i the estimated probability that v_i is correct; on this basis, a *truthfulness inference algorithm* estimates the truthfulness of the user, concluding the iteration. The algorithm employs multiple iterations, so that the information about a user's truthfulness gained from some statements can propagate to other statements.

The attribute inference algorithm is the heart of Crowdsensus. Originally, we used standard algorithms, such as Bayesian inference, but we quickly noticed that they were suboptimal for the real case of maps. First, users do not have independent information on the correct value of attributes. There is typically only a few ways in which users can learn, for instance, the phone number of a restaurant: they can go there and ask, or they can read it on a coupon, for instance, but 100 users providing us data will not correspond to 100 independent ways of learning the phone number. Thus, we had to develop algorithms that can take into account this lack of independence. Second, business attributes have different characteristics, and we found it very important to develop attribute inference algorithms tailored to every type of attribute. For example, geographical positions (expressed as a latitute-longitude pairs) have a natural notion of proximity (a distance), and it is essential to make use of it in the inference algorithms; websites also have some notion of distance (at least insofar as two websites may belong to the same domain). Thus, our implementation of Crowdsensus employs different inference algorithms for different types of attributes. The complete system is more complex in several respects: it contains algorithms for attributes with multiple correct values, for dealing with spam, and for protecting the system from abuse. Furthermore, we remark that the Google Maps data pipeline comprises several inter-dependent algorithms and subsystems; we designed Crowdsensus as one of the many components of the overall pipeline.

We illustrate the working of the Crowdsensus algorithm via a simple example. We consider the case of N users and M attributes; the true value of each attribute is chosen uniformly at random among a set of K possible values. For each user u, we choose a probability p_u uniformly at random in the $[0, 1]$ interval: user u will provide with probability p_u the correct attribute value, and will provide with probability $1-p_u$ a value selected uniformly at random among the K possible values. We note that Crowdsensus is not informed of the probability p_u of a user u: rather, Crowdsensus will compute the truthfulness q_u for u from the statements by u. For simplicity, we assume that for each attribute, we have J estimates provided by J users selected at random. We experimented using a standard Bayesian inference for attribute values. For $M = 1000$, $N = 100$, $K = 10$, and $J = 10$, Crowdsensus has an error rate in the reconstruction of the correct value of each feature of 2.8%. In contrast, a (non-iterative) algorithm that performs Bayesian inference without using information on user reputation has an error rate of 7.9%. The roughly three-fold reduction in error rate, from 7.9% to 2.8%, is due to the power of user reputation in steering the inference process. The statistical correlation between the true truthfulness p_u and the reconstructed truthfulness q_u over all users was 0.988, indicating that Crowdsensus was able to precisely reconstruct the user truthfulness. If we take $J = 5$, the error rate of Crowdsensus is 12.6%, compared with an error rate of 22% for standard Bayesian inference; the correlation between true and inferred truthfulness is 0.972.

[7]In fact, the algorithm computes a *sub-probability distribution,* as the probabilities may sum to less than 1.

4. CONCLUSIONS

We conclude on a note of optimism for the role of reputation systems in mediating on-line collaboration. Reputation systems are the on-line equivalent of the body of laws that regulates the real-world interaction of people. As a larger fraction of people's productive lives will be carried on via on-line, computer-mediated interaction, we expect that the development of such an on-line body of algorithmic legislation will be a rich field of work and research, with wide implications for society overall.

5. ACKNOWLEDGMENTS

This work has been supported in part by CITRIS: Center for Information Technology Research in the Interest of Society, and by ISSDM: Institute for Scalable Scientific Data Management. We thank Shelly Spearing and Scott Brandt for their enthusiastic support.

6. REFERENCES

[1] B. Adler, K. Chatterjee, L. de Alfaro, M. Faella, I. Pye, and V. Raman. Assigning trust to Wikipedia content. In *Proc. of WikiSym 08: International Symposium on Wikis*. ACM Press, 2008.

[2] B. Adler and L. de Alfaro. A content-driven reputation system for the Wikipedia. In *Proc. of the 16th Intl. World Wide Web Conf. (WWW 2007)*. ACM Press, 2007.

[3] B. Adler, L. de Alfaro, I. Pye, and V. Raman. Measuring author contributions to the Wikipedia. In *Proc. of WikiSym 08: International Symposium on Wikis*. ACM Press, 2008.

[4] K. Chatterjee, L. de Alfaro, and I Pye. Robust content-driven reputation. In *First ACM Workshop on AISec*. ACM Press, 2008.

[5] M. Dale, A. Stern, M. Deckert, and W. Sack. System demonstration: Metavid.org: A social website and open archive of congressional video. In *Proc. of the 10th International Conference on Digital Government*

Research: Social Networks: Making Connections between Citizens, Data, and Government, pages 309–310. Digital Government Society of North America, 2009.

[6] C. Dellarocas. The digitization of word-of-mouth: Promises and challenges of online reputation systems. *Management Science*, October 2003.

[7] C. Dellarocas, M. Fan, and C.A. Wood. Self-interest, reciprocity, and participation in online reputation systems. Technical Report Paper 205, Center for eBusiness, Sloan School of Management, MIT, 2005.

[8] G. Druck, G. Miklau, and A. McCallum. Learning to predict the quality of contributions to Wikipedia. In *Proceedings of AAAI: 23rd Conference on Artificial Intelligence*, 2008.

[9] S. Goel, D.M. Reeves, and D.M. Pennock. Collective revelation: A mechanism for self-verified, weighted, and truthful predictions. In *EC 09: Proc. of the 10th ACM Conference on Electronic Commerce*, pages 265–274. ACM Press, 2009.

[10] M. Haklay and P. Weber. OpenStreetMap: User-generated street maps. *Pervasive Computing*, pages 12–18, 2008.

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

[12] Y. Liu, X. Huang, A. An, and X. Yu. Helpmeter: A nonlinear model for predicting the helpfulness of online reviews. In *WI-IAT: IEEE/WIC/ACM Intl. Conf. on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 793–796, 2008.

[13] 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.

[14] H. Zeng, M.A. Alhoussaini, L. Ding, R. Fikes, and D.L. McGuinness. Computing trust from revision history. In *Intl. Conf. on Privacy, Security and Trust*, 2006.

[15] Z. Zhang. Weighing stars: aggregating online product reviews for intelligence E-commerce applications. *IEEE Intelligent Systems*, 23(5):42–49, 2008.