Reputation Systems for Open Collaboration*

Bo Adler†
Fujitsu Labs of America
thumper@alumni.caltech.edu

Luca de Alfaro‡
Google, Inc.
luca@dealfaro.org

Ashutosh Kulshreshtha
Google, Inc.
ashu@google.com

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

ABSTRACT

Open, on-line collaboration is becoming an important way in which content is created, and knowledge is organized. Open collaboration, however, carries challenges both for content creation, and for content use. The process of content creation is open to abuse. On the other end, content consumers are faced with the outcome of a complex collaboration process, and can have difficulty discriminating high from low-quality content.

Reputation systems can help stem abuse, and can offer indications of content quality. We discuss some basic design principles and choices in the design of *content-driven reputation systems,* which rely on an analysis of the content and the collaboration process, rather than on explicit user feedback. Content-driven reputation systems can be easier to employ and harder to subvert. We illustrate the choices in their design by examining two systems we have built: the WikiTrust reputation system for Wikipedia authors and content, and the Crowdsensus reputation system for Google Maps editors. We conclude with some thoughts on the relationship between reputation systems for on-line collaboration and the usual bodies of law that regulate people's off-line interaction.

1. INTRODUCTION

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 later expanded to include, for instance, video editing (e.g., MetaVid [8]), documents (e.g., Google Docs[1], ZOHO[2]), architectural sketching (e.g., Sketchup[3]), and geographical maps (e.g., OpenStreetMaps [14], Map Maker[4]).

Open collaboration carries immense promise, as shown for instance 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"

*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.

‡On leave from the University of California, Santa Cruz. Part of his work was performed while the author was at the University of California, Santa Cruz.

Communications of the ACM 2010

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

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 particularly 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 [10]. Content-driven reputation systems derive user feedback from user actions, and can be more resistant to manipulation [6].

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 [9, 10].

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 [21, 22] and other approaches to Wikipedia content reputation [23]. The algorithms PageRank [17] and HITS [15] 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.

2. WIKITRUST: A REPUTATION SYSTEM FOR WIKI AUTHORS AND CONTENT

We present here the main ideas in WikiTrust[5], a reputation system for wiki authors and content we developed with the following goals:

- Incentivize users to give lasting contributions.
- Help users, and editors, increase the quality of the content and spot vandalism.
- Offer content consumers a guide to the quality of the 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.

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. Users who wish to investigate the origin of text can click on any word in the WikiTrust tab: they are redirected to a page showing the author of the clicked word, as well as the full edit in which the word was inserted. WikiTrust does not currently display user reputations, out of a desire not to alter the social experience of contributing to the Wikipedia.

2.1 The User Reputation System

WikiTrust relies solely on the following assumptions:

- The content evolves in a sequence of revisions; each revision being produced by a single author.
- It is possible to compare two revisions and measure their difference.
- It is possible to track content that is unchanged across revisions.

Thus, while WikiTrust was conceived for wikis, it relies on ideas that are applicable to many collaborative systems.

The quality of a contribution. The reputation of users is computed according to the quality and quantity of contributions they make. A contribution is considered of good

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

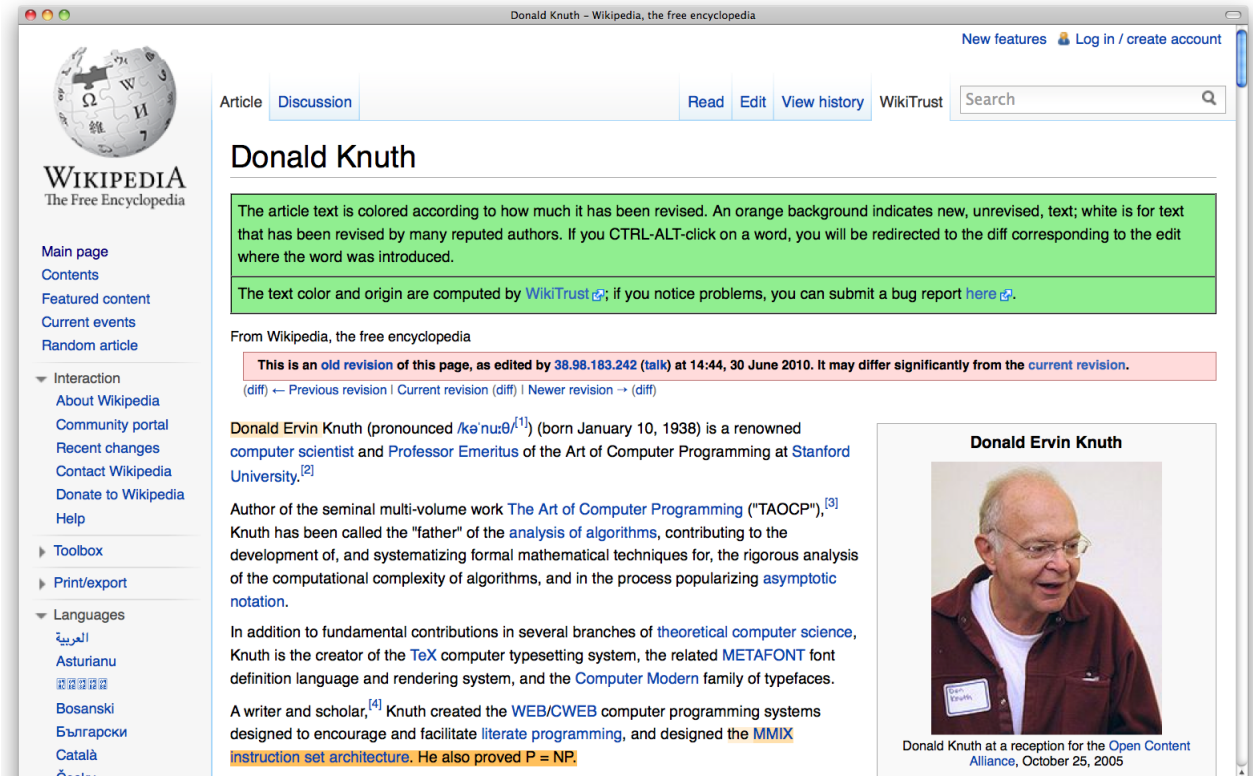
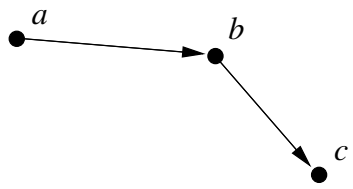
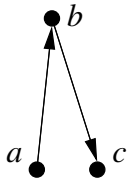


Figure 1: The Wikipedia page for Don Knuth, as rendered by WikiTrust. The text background is a shade of orange that is the darker, the lower the reputation of the text.



(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.

quality if the change it introduced is preserved in subsequent revisions [3, 12, 4]. To estimate how much of each contribution is preserved in subsequent revisions, WikiTrust relies on an edit distance function d. The distance $d(r, r')$ between two revisions r and r' measures the extent of the change in going from r to r'; we compute it on the basis of how many words have been deleted, inserted, replaced, and displaced in the edit that led from r to r' [19, 18]. Relying on an edit distance, rather than on lexical or language analysis, yields a reputation system that is language-independent: a benefit given the large number of languages in Wikipedias. Each revision b is judged with respect to a past revision a and a future revision c. We consider the situation from the point of view of the author C of c (see Figure 2). The author C clearly believes that the version c is more desirable than previous versions, since she inserted it. From the point of view of C, the revision b improved on a by an amount $d(a,c) - d(b,c)$. This improvement was produced by an amount of work equal to the size $d(a,b)$ of the change from a to b. The quality of b with respect to a and c is measured as the improvement per unit of work:

$$q(b \mid a,c) = \frac{d(a,c) - d(b,c)}{d(a,b)}.$$

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. We note that the above algorithm is unable to judge newly created revisions: it must wait until

other users express their judgement implicitly by creating a subsequent revision.

From contribution quality to user reputation. WikiTrust considers only non-negative reputation values, and it assigns to new users a reputation very close to 0. As it is very easy to register a new identity on the Wikipedia, new users have the same reputation as vandals: if vandals could receive any lower reputation, they would simply move to a new identity. Each revision b is judged with respect to 5 subsequent revisions, and with respect to a set of past revisions which includes the 5 revisions preceding b, as well as 2 previous revisions by high-reputation authors, and 2 previous revisions with high average text reputation. This advanced selection process is used to make the reputation system hard to subvert; for the details, see [6]. After comparing b with a previous revision a and a future revision c, WikiTrust increments the reputation $r(B)$ of the author B of b by an amount that is proportional to the size of the edit leading to b, to the quality $q(b \mid a, c)$, and to the logarithm of the reputation $r(C)$ of the author C of c. Thus, the influence of a user's judgements on other users is proportional to the logarithm of the judging user's reputation. 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.

The fact that user reputation is derived from an analysis of user edits 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.

To ensure fairness, WikiTrust employs an additional technique. When considering the triple (a, b, c), it increments the reputation of B only if C has higher reputation, or if the revision b has stood the test of time (on Wikipedia, for a few days) without anyone reverting it [6]. Provided that new edits are patrolled every few days, this scheme prevents a user from creating fake identities, and using these fake identities to gain reputation, performing a so-called *Sybil attack* [11, 7, 16]. While Sybil attacks are difficult to avoid in general, they can be limited in content-driven reputation systems where there is no "dark corner" in the content, that is, no place that is hidden from inspection by the general user community.

Evaluation: predicting revision quality. 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 [3]. 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.

For each revision b of a Wikipedia, let r_b be the reputation of the revision's author at the time the revision was made, and let q_b be the quality of the revision, taken to be the average of $q(b \mid a, c)$ over the past revisions a and the future revisions c considered by the reputation algorithm. Note that the reputation r_b depends on the *past* of b (the reputation accrued due to b is not part of r_b), while the quality q_b depends on how the change performed in b is preserved in the *future* of b. Thus, the only common point between r_b and q_b is the author of b, and any correlation between r_b and q_b is due to the predictive ability of reputation on quality. To measure this predictive ability, we consider the ability of low reputation (r_b in the lowest 10% of reputation values) to predict reversions ($q_b < -0.8$). The data, computed for b ranging over all revisions of various Wikipedias, is reported in Table 1. The *precision* is the percentage of contributions by low-reputation authors that were reverted; the *recall* is the percentage of reverted contributions that was made by low-reputation authors. 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].

2.2 From User to Content Reputation

One of the main goals of a reputation system for collaborative content is to provide indications to users about the quality of the content. Ideally, the reputation system should be informative, robust, and explainable:

- *Informative.* The reputation of content should be a good indicator of content quality.
- *Robust.* It should be difficult for malicious users to cause arbitrary content to gain high reputation, without wider support from the community.
- *Explainable.* It should be possible for users to understand how their (and other users') actions affect content reputation.

WikiTrust computes content reputation according to the extent to which the content has been revised, and according

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

to the reputation of the users who revised it [20, 2]. 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: predicting deletions. 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 [2]:

- *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.

2.3 A Few Lessons Learned

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

Much of the feedback is related to our unfortunate decision of calling *text trust* what in this paper is more appropriately called *text reputation.* As the text reputation is intended to help users decide how much they can trust the text, we thought that "trust" would be a concise and informative term for the quantity we computed. What a mistake! While people seem to accept that a user's "reputation" is a mathematical quantity computed by an algorithm, and it is not indicative of a user's reputation in real life, no such flexibility is generally accorded to the word "trust".

The original reputation system described in [3] 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 [6]. 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.[7] 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.

Developing a system capable of computing in real-time the user and content reputation for the largest Wikipedias turned out to be a major challenge. The code needs to be very robust, and very easy to debug: the Wikipedia is a veritable Babel's library [5], as everything you can imagine (and many things you cannot) appear somewhere in it, and can break your code, usually a few days into a long computation. Furthermore, the size of the Wikipedia is such that only linear-time algorithms worked for us. Indeed, our edit distance algorithms work in linear-time: we accomplish this apparently impossible feat by sacrifizing accuracy for very large revisions, trusting that Wikipedia revisions with more

[7] As an example, if Ada adds n words to a revision a, obtaining revision b, and Bob replaces these n words with other n words, leading to c, we have $d(a,b) = n$, $d(a,c) = n$, and $d(b,c) = \frac{n}{2}$, so that $q(b \mid a, c) = \frac{1}{2}$.

- **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.

than a million words are probably the work of spam robots, rather than revisions for which we need precise analysis.

3. 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 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 [17] or HITS [15]. 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 [17, 15]. 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 [6]. A global reputation system that does not consider action or contribution timestamps lacks this valuable defense. Indeed, a common attack against reputation systems based on global graph algorithms consists for the attackers in replicating a valid part of the graph, with fake identities taking place of valid users in the replica [7]. If the algorithm is sensitive only to the graph structure, it confers to the users involved in the replica the same high reputation it confers to the users whose interactions are copied. The use of timestamps, or of chronological algorithms, is successful in breaking the symmetry between the preexistent and the replicated portions of the graph, preventing attacks.

4. CROWDSENSUS: A REPUTATION SYSTEM FOR MAPS BUSINESS LISTINGS

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.[8] The

[8]Another Google system, called Map Maker, enables users to draw the geographical features of maps, such as roads, parks, rail tracks, and so forth. Map Maker is based on

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 problem* [13], 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[9] 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.

5. CONCLUSIONS

We conclude on a note of optimism for the role of reputation systems in mediating on-line collaboration. In many

a different reputation system, where user contributions are either auto-approved (if the user has sufficient reputation), or they require approval by other high-reputation users.

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

ways, reputation systems are the on-line equivalent of the body of laws that regulates the real-world interaction of people, and as such, we expect that they will receive a growing amount of interest from practitioners and researchers alike. Among the research directions that draw our interest, we mention the following two.

How can we develop a "population-dynamics science" approach to reputation systems? Reputation systems are commonly studied from the point of view of individual users, or fixed sets of users. But reputation systems live in a dynamical environment, where users can join, start contributing or interacting, develop connections, leave, and perhaps come back. How can reputation systems lead to happy, active, healthy communities? How should the influence and needs of novice and established users be balanced? Can reputation systems be studied as elements of dynamical systems, of which individual users are the "elementary particles"?

How can the contrasting goals of reputation systems be managed? Reputation systems are often called to serve different goals at once. For instance, reputation systems should provide an *incentive* to constructive behavior, and also, *inform* other users of the quality of particular users or content. These two goals can be in contrast. Suppose that we somehow learn from the first contribution of a user that the user is likely to produce excellent future contributions. Should we then award the user top reputation right away? Where would the incentive provided by trying to attain high reputation go? How can we build reputation systems that meet multiple goals?

As the online world becomes increasingly the place where people interact and collaborate, we believe that the study of user reputation systems will be an important chapter of computer science research.

6. 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.

7. REFERENCES

[1] C. Boyd A. Jøsang, R. Ismail. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43:618–644, 2007.

[2] B.T. 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.

[3] B.T. 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.

[4] B.T. 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.

[5] J.L. Borges. *La biblioteca de Babel*. 1941. Republished in Ficciones, 1944.

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

[7] A. Cheng and E. Friedman. Sybilproof reputation mechanisms. In *Proc. of the ACM SIGCOMM workshop on Economics of peer-to-peer systems*. ACM Press, 2005.

[8] 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 Intl. Conf. on Digital Government Research*, pages 309–310, 2009.

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

[10] 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.

[11] J.R. Douceur. The Sybil attack. In *Peer-to-Peer Systems: First Intl. Workshop*, volume 2429 of *Lect. Notes in Comp. Sci.*, pages 251–260, 2002.

[12] 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.

[13] 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.

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

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

[16] B.N. Levine, C. Shields, and N.B. Margolin. A survey of solutions to the Sybil attack. Technical Report 2006-052, Univ. of Massachussets Amherst, 2006.

[17] 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.

[18] W.F. Tichy. The string-to-string correction problem with block move. *ACM Trans. on Computer Systems*, 2(4), 1984.

[19] Robert A. Wagner and Michael J. Fischer. The string-to-string correction problem. *J. ACM*, 21(1):168–173, 1974.

[20] 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.

[21] Y. Liu, X. Huang, A. An, X. Yu. HelpMeter: A nonlinear model for predicting the helpfulness of online reviews. Wi-iat, 1:793–796, IEEE/WIC/ACM Intl. Conf. on Web Intelligence and Intelligent Agent Technology, 2008.

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

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