Robust Content-Driven Reputation*

Krishnendu Chatterjee[1] Luca de Alfaro[1] Ian Pye[2]

[1]Computer Engineering Dept.
UC Santa Cruz, CA, USA
{c_krish,luca}@soe.ucsc.edu

[2]Computer Science Dept.
UC Santa Cruz, CA, USA
ipye@ucsc.edu

Technical Report UCSC-SOE-08-06
School of Engineering, University of California, Santa Cruz
May 2008

Abstract

In content-driven reputation systems for collaborative content, users gain or lose reputation according to how their contributions fare: authors of long-lived contributions gain reputation, while authors of reverted contributions lose reputation. Existing content-driven systems are prone to Sybil attacks, in which multiple identities, controlled by the same person, perform coordinated actions to increase their reputation. We show that content-driven reputation systems can be made resistent to such attacks by taking advantage of the fact that the reputation increments and decrements depend on content modifications, which are visible to all.

We present an algorithm for content-driven reputation that prevents a set of identities from increasing their maximum reputation without doing any useful work. A variation of the algorithm ensures that the reputation of each identity which performs only non-useful work decreases. Here, work is considered useful if it causes content to evolve in a direction that is consistent with the actions of high-reputation users. We argue that the content modifications that require no effort, such as the insertion or deletion of arbitrary text, are invariably non-useful. We prove a truthfullness result for the resulting system, stating that users who wish to perform a contribution do not gain by employing complex contribution schemes, compared to simply performing the contribution at once. In particular, splitting the contribution in multiple portions, or employing the coordinated actions of multiple identities, do not yield additional reputation. Taken together, these results indicate that content-driven systems can be made robust with respect to Sybil attacks.

1 Introduction

On-line collaboration is fast becoming one of the primary ways in which information is being created, aggregated, and shared. The success of sites such as the Wikipedia, YouTube, MySpace, and of the many wikis and discussion groups disseminated over the web owes to their ability to harness the contributions of millions of people all over the world. As the volume of such collaborative information grows, so does the problem of assessing its quality, preventing vandalism and spam, and providing incentives to constructive collaboration. Reputation systems have been proposed as a help in this direction.

Some of the largest bodies of collaborative information are *versioned:* users build on each other's contributions, modifying and improving them. The prime example of such bodies of information are wikis,

*This work has been partially supported by CITRIS: Center for Information Technology Research in the Interest of Society.

among which the Wikipedia, currently the largest on-line encyclopedia and the 8th most frequently visited site on the Web.[1] As on-line collaboration expands, versioned information will become increasingly common; indeed, editable and shareable maps, such as layers on Google Earth, and edit-shared documents, represent additional examples. Versioned bodies of information can employ *content-driven* reputation systems, which compute user reputation on the basis of content evolution: authors of long-lived contributions gain reputation, while authors of contributions which are short-lived or reverted lose reputation [2]. Content-driven reputation systems thus provide an incentive to contribute lasting content; they are also intrinsically objective, as the reputation changes are tied to content evolution. For instance, the only way a user A can denigrate a user B is by reverting B's contribution; if subsequent users reinstate B's contribution, it is A's reputation, rather than B's, which will suffer the most. The content-driven reputation of Wikipedia authors has been shown to be a good statistical predictor of the longevity (and thus, presumably, of the quality) of their future contributions [2]; author reputation has also been used as the basis for computing text trust [1].

Reputation confers status, and it can be used to manage edit rights to high-visibility information, or as the basis for the computation of content quality [1, 3]. Consequently, reputation systems are subject to attack by users who wish to increase their reputation without performing useful (and thus, time-consuming) work. Thus far, the use of content-driven reputation has not led to resistance to attacks. Indeed, the reputation system proposed in [2] can be subject to a wide number of *Sybil attacks*, in which a single person uses multiple identities (or *sock-puppets*) to increase her reputation without providing valuable contributions [7, 4, 14, 11, 9]. In the simplest of these attacks, a user controls two identities: a primary identity A, and a "sacrifical" identity $\hat{A}$. In the attack, $\hat{A}$ first performs vandalism, for instance by deleting the entire content of a wiki article, or by inserting spurious text; A then promptly reinstates the original content of the article. As subsequent users build on the unvandalized content of the page, A reputation will rise, since A's intervention is preserved. Many similar attacks are possible, and some of them are described in this paper.

Attacks to reputation sytems are a pervasive problem, and [4, 9, 11] provide comprehensive surveys of the general problem and of solution approaches. In this paper, we show that content-driven reputation systems can be made resistant to many forms of Sybil attacks. The key idea consists in exploiting the connection between content evolution, and reputation computation. In particular, reputation changes are due to content modification that can be inspected by all users. Thus, under the assumption that content is visited by a wide variety of users, as it happens in real systems, we will be able to provide strong guarantees of immunity to attacks. The algorithms we present do not depend on the specific nature of the content, and can be applied to wikis, as well as to other versioned content: all we need to assume is that we have some way to measure the *distance* between contributions. In wikis and other text-based systems, *edit distance* can be used [18, 15, 6]. Nevertheless, we chose to present the algorithms in the context of wikis, both to provide readers with a familiar context, and because the evaluation of the algorithms will be performed on the French Wikipedia.

Our starting point is the content-driven reputation algorithm proposed in [2]. The algorithm assesses the value of each contribution by comparing it with past and future versions of the content, due to different authors. If the contribution went in the general direction of content evolution, as estimated from the change from past to future versions, the contribution is judged positively, and its author gains reputation; otherwise, it is judged negatively, and the author loses reputation.

As a first step, we describe the REPUTATION-CAP algorithm, where the reputation that can be gained by the contributing author's is capped by the reputation of the authors of the past and future versions to which it is compared. The REPUTATION-CAP algorithm prevents groups of sock-puppets from increasing their maximum reputation unless they perform *useful work*, or work that is considered positively by higher-reputation users. Unfortunately, the REPUTATION-CAP algorithm also prevents the global reputation growth of sys-

[1] In May 2008, according to the rankings at `www.alexa.com`.

tem users. In particular, if everybody starts with low reputation, nobody can ever gain high reputation. To remedy this, we relax the assumptions on the REPUTATION-CAP algorithm, allowing users to gain uncapped reputation, provided their contributions have first withstood the test of time without being judged negatively; this yields the REPUTATION-CAP-NIX algorithm. We show that under weak assumptions on content visitation rate, assumptions that hold for most real systems including the Wikipedia, the REPUTATION-CAP-NIX is able to prevent Sybil attacks while allowing global reputation growth.

Next, we turn to the *truthfulness* property, stating that if a user wishes to perform an edit e, the user cannot gain by splitting e in multiple sub-edits, or by employing complex editing schemes involving sock-puppets, compared to doing e directly. This property is inspired by mechanism design in game theory: there, a mechanism (such as an auction procedure) is truthful if it is a weakly dominant strategy for the players to reveal their utility [5, 8, 16, 12]. We show that while the REPUTATION-CAP-NIX algorithm does not enjoy the truthfulness property, a simple modification does. The modification allows some *reputation denial* attacks, but this can be once more remedied under weak assumptions, met in the real world, on the relative infrequency of disputes (reversion wars) among high-reputation authors. This leads to our final algorithm, the LOCAL-GLOBAL algorithm, which is our candidate for implementation in on-line content-driven reputation systems. The algorithm is resistant to Sybil attacks and truthful, under weak assumption about visitation and editing dynamics of a site.

We evaluate the algorithms with respect to their ability to produce informative, high-quality reputation information, which has good predictive value with respect to the longevity of future contributions by the authors. Using a 100,000-article, 56-million revision subset of the French Wikipedia as our dataset, we show that the modifications required to make the algorithms robust do not decrease the quality of the reputation they compute.

2 Content-Driven Reputation

Before presenting the robust reputation algorithms, it is useful to summarize the content-driven algorithm of [2], on which the robust algorithms are based, and examine attacks to which this original algorithm can be subject.

2.1 Notation

We consider *content-driven* reputation algorithms which compute author reputation on the basis of the sequence of versions of each wiki article. The algorithms are on-line, and examine each version as it is introduced in the system. We denote the versions of an article p by $v_1^p, v_2^p, v_3^p, \ldots$; the letter p stands for *page*. We indicate with $a_1^p, a_2^p, a_3^p, \ldots$ the authors of these versions, and we indicate by $e_i^p = v_{i-1}^p \rightsquigarrow v_i^p$ the edit (the text modification) that produced v_i^p, for $1 < i$. We indicate the times at which the versions v_1^p, v_2^p, ... have been created by t_1^p, t_2^p, ...; for simplicity, we assume that all versions have distinct timestamps. We denote by $r_i^p(a)$ the reputation of author a just before version v_i^p was entered. We assume that the reputation is bounded to the range $[0, T_{\max}]$, for some $T_{\max} > 0$, so that if a reputation increment or decrement causes the reputation to go below 0 (resp., above $T_{\max}$), the reputation is set to 0 (resp., to $T_{\max}$). In the following, we will occasionally omit the superscript p and focus on one article at a time; however, we stress that the algorithms operate strictly chronologically, according to the order in which versions are entered into the wiki.

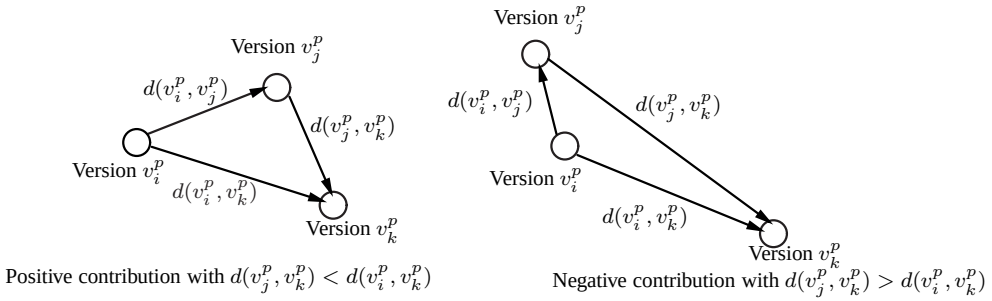


Figure 1: The $\mathsf{qual}(v_i^p, v_j^p, v_k^p)$ computation.

2.2 The BASIC algorithm

The *content-driven* reputation of [2] is based on the idea of assigning reputation to authors according to how long their contributions last: authors of long-lived contributions gain reputation, and authors of short-lived or reverted contributions lose reputation. In [2], it was proposed to measure the contribution given by an author in two ways: according to the text that was inserted (the *text* contribution), and according to the overall modification performed (the *edit* contribution). The edit contribution largely subsumes the text one; for this reason, we discuss here only the algorithm based on edit contributions. Our evaluation, reported in Section 4, will show that considering edit contributions only does not yield inferior quality for the computed reputation, compared to the algorithm of [2].

The BASIC algorithm takes, as a basic building block, an algorithm to compute the *edit distance* between two text documents. The edit distance *edit distance* $d(v, v') \geq 0$ between documents v and v' is a measure of the amount of text insertions, deletions, and replacements that is required to transform v into v'. The problem of computing edit distances has been well-studied in the literature [18, 15, 6]; the particular approach we chose is discussed in [2]. All authors initially have reputation zero. When a version v_k^p is entered into the wiki, the algorithm considers triples of versions (v_i^p, v_j^p, v_k^p), with $0 < i < j < k$. In each triple (v_i^p, v_j^p, v_k^p), the author a_k judges the quality of v_j^p on the basis of the version v_k^p she just produced, and on the basis of a previous version v_i^p taken as reference. The idea is as follows. The author a_k, having just produced version v_k^p, will naturally believe that v_k^p is better (in her own personal opinion) than any previous version. Thus, a_k judges a_j on the basis of whether a_j's edits brought the article closer to the version v_k^p. The total change from v_i^p to v_j^p is $d(v_i^p, v_j^p)$. This change caused the distance to v_k^p to decrease by $d(v_i^p, v_k^p) - d(v_j^p, v_k^p)$. Thus, the algorithm computes the ratio

$$\mathsf{qual}(v_i^p, v_j^p, v_k^p) = \frac{d(v_i^p, v_k^p) - d(v_j^p, v_k^p)}{d(v_i^p, v_j^p)} \tag{1}$$

between the total change, and the change towards v_k^p. The situation is illustrated in Figure 1. We say that the version v_j^p *receives negative feedback* if $\mathsf{qual}(v_i^p, v_j^p, v_k^p) < 0$. As the edit distance satisfies the triangular inequality $d(v, v') \leq d(v, v'') + d(v'', v')$ for all versions v, v', v'', we have that, for all $0 < i < j < k$, that $-1 \leq \mathsf{qual}(v_i^p, v_j^p, v_k^p) \leq 1$. Given a positive integer m as a parameter, the BASIC algorithm considers triples (v_i^p, v_j^p, v_k^p), where $i+1 = j$ and $k - i \leq m$ (so that the judged version v_j^p is compared to the preceding one, and a window of size m revisions is analized). The algorithm increases the reputation of a_j by the amount

$$\mathit{Inc}^p(i, j, k) = c_s \cdot d(v_{j-1}^p, v_j^p) \cdot \mathsf{qual}(v_i^p, v_j^p, v_k^p) \cdot w(r_k^p(a_k)) \ , \tag{2}$$

where $c_s > 0$ is a scaling constant, $d(v_{j-1}^p, v_j^p)$ is the amount of change performed in the edit $e_j = v_{j-1}^p \rightsquigarrow v_j^p$, and $r_k^p(a_k) \geq 0$ is the reputation of the judge a_k at the time v_k^p is created; w is a monotonic increasing

function. As in [2], we take $w(x) = \log(1.1 + r(x))$, thus reducing the influence of high-reputation authors: if this were not done, our experiments indicated that high-reputation authors would wield disproportionate power. The results of this paper are independent on the particular choice of w, provided $w(0) > 0$, and $x \geq y$ implies $w(x) \geq w(y)$.

2.3 Attacks against the BASIC algorithm

The BASIC algorithm is prone to attacks, in which users can increase their reputation without performing any amount of productive work. These attacks rely on *sock-puppets*, or multiple user identities that are controlled by the same person.

A simple attack of this kind is the *delete-restore* attack. The attack can be carried out by a person having two identities: a main identity A, whose reputation the person wants to increase, and a sock-puppet identity A'. In the attack, A' removes all the text of the article, producing an empty version v_j^p; immediately afterwards, identity A restores the text in version v_{j+1}^p. Since stable Wikipedia pages usually evolve via small edits, subsequent authors will build on version v_{j+1}^p, and $\mathsf{qual}(v_j^p, v_{j+1}^p, v_k^p)$ will be positive and close to 1, leading to an increase in reputation for A. Identity A' of course loses reputation, but this does not matter: this identity is simply a "sacrificial" one, and all it matters is that it is permitted to carry out edits; if A' is banned, the person controlling A and A' can simply create a new sock-puppet A''.

The delete-restore attack is somewhat easy to spot: wiki administrators may become suspicious if they notice that A is always restoring the text of deleted pages, while doing little else. A variation that is harder to spot is the *add-restore* attack, in which the sock-puppet identity A' introduces spurious text in an article (for instance, a nonsensical paragraph, spam, or other clearly inappropriate material), which A proceeds to remove in the immediately subsequent edit.

Another attack is the *fake-followers attack*. In this attack, a person controls a main identity A, and some sock-puppet identities A_0, A_1, $A_2, \cdots$. In this attack, A performs an edit $v_{j-1}^p \rightsquigarrow v_j^p$ which introduces any material, plausible or not; immediately afterwards, A_0, A_1, A_2, ..., proceed to develop on version v_j^p, thus increasing A's reputation. When the edit of A is finally undone at version $v_k^p = v_{j-1}^p$, if $k - j > m$, the reputation of A is not harmed, so that A can retain the gains accrued in the course of the attack.

These attacks have many variations, and are only a representative sample of the set of possible successful attacks to the BASIC algorithm. The focus of this paper is not to provide a classification of attacks, but to present modified content-driven algorithms that are robust with respect to *any* sock-puppet attack.

3 Robust Content-Driven Reputation

In this section, we develop from algorithm BASIC new algorithms that are resistant to Sybil attacks and that enjoy the truthfulness property.

3.1 The REPUTATION-CAP algorithm

The first algorithm, REPUTATION-CAP, bounds the reputation increase, so that the maximum reputation of a set of identities can increase only if useful work is performed. In order to update the reputation of an author, we see from (1) that algorithm BASIC compares a version v_j^p produced by the author with two two versions, that are taken as reference: an older version v_i^p (for $i = j - 1$), and a newer version v_k^p. The attacks described in Section 2.3 rely on the fact that at least one of the two reference versions is due to a sock-puppet, rather than to a legitimate author. This suggests that, when updating the reputation of a_j^p, we do not increase it beyond that of the reputations of a_i^p or a_k^p: this prevents the use of low-reputation sock-puppets

for increasing the reputation of the main identity. In the following, for simplicity we drop the superscript p, since the algorithm only compares versions belonging to the same article.

The REPUTATION-CAP algorithm is obtained by modifying the reputation increase of the basic algorithm. The REPUTATION-CAP algorithm first computes $\mathit{Inc}(i,j,k)$ as in (2), and then proceeds as follows:

- If $\mathit{Inc}(i,j,k) < 0$, then the reputation of a_j is incremented by $\mathit{Inc}(i,j,k)$ (leading to a reputation decrease); this coincides with the basic algorithm.
- If $\mathit{Inc}(i,j,k) \geq 0$, the algorithm first retrieves the current reputations $r_k(a_i)$, $r_k(a_j)$, $r_k(a_k)$ of a_i, a_j, a_k; it then updates the reputation of a_j to

$$\max(r_k(a_j), \min(r_k(a_i), r_k(a_k), r_k(a_j) + \mathit{Inc}(i,j,k))). \tag{3}$$

The formula (3) has two consequences. If the reputation of a_j is greater than that of a_i or a_k, the reputation of a_j cannot increase, and it can decrease if $\mathsf{qual}(v_i, v_j, v_k) < 0$. On the other hand, if the reputation of a_j is lower than both the reputations of a_i and a_k, then the reputation of a_j can increase, but only up to the minimum of the reputations of the "referees" a_i and a_k. Thus, an author can gain high reputation only when her versions are compared with versions produced by high-reputation authors. In particular, if an author a starts with low reputation, and if her versions are only compared with the versions of authors of reputation below r, the author a will be unable to gain reputation above r. This is the mechanism that prevents the sock-puppet attacks outlined in Section 2.3. To ensure that enough triples with high-reputation reference points are considered, when a version v_k is entered, the algorithm considers all triples (i,j,k) with $0 < i < j < k$ and $k - i \leq m$, thus lifting the restriction $i + 1 = j$ of algorithm BASIC.

The key property we wish to show of the REPUTATION-CAP algorithm can be informally summarized as follows: *If a person controls a set of sock-puppets whose maximum reputation is r, then unless useful editing work is done, no sock-puppet can increase the reputation beyond r.* To formalize this statement, we need to provide a definition of "useful". We formalize this notion as follows.

Definition 1 (useful work) Given $r \in [0, T_{\max}]$ and a triple (i,j,k) with $0 < i < j < k$, we say that the triple (i,j,k) is *r-good* iff both a_i and a_k have reputation at least r when the triple is created or evaluated; precisely, (i,j,k) is r-good if $\big(r_i(a_i) \geq r \text{ or } r_k(a_i) \geq r\big)$ and $r_k(a_k) \geq r$. We say that the version v_j is *r-useful* iff $\mathsf{qual}(v_i, v_j, v_k) > 0$ for some r-good triple (i,j,k). A version that is not r-useful work is called *r-useless.*

Intuitively, this definition states that the version v_j is useful iff there is at least a pair of reference versions v_i and v_k, one in the past, and the other in the future, both by authors of reputation at least r, that judge in positive fashion the contribution of v_j. High-reputation authors do not always fully agree agree on what is the best direction of change for an article; the definition gives the benefit of the doubt to version v_j, and calls it useful if it agrees with the direction of change undertaken by at least some of these authors.

Nevertheless, we argue that producing a useful version does not come for free, but in the great majority of cases, requires some effort on the part of the author. A useful version, after all, is a version that comes closer to some *future* contribution by high-reputation authors: it is unlikely that such a version can be produced by acts that do not require effort, such as removing or inserting text at random. The following theorem provides the main property of the REPUTATION-CAP algorithm, which shows that a set of authors cannot increase their maximal reputation without doing useful work. We formalize this property as the *no-free-increase* property. This theorem rules out Sybil attacks such as the ones outlined in Section 2.3.

> **No-free-increase property.** Consider a set U of authors, which at time t all have reputation below $r \in [0, T_{\max}]$. If, after time t, the authors in U only contribute r-useless versions, then reputation of no authors in U can grow above r.

Theorem 1 *The* REPUTATION-CAP *algorithm ensures the no-free-increase property.*

Proof. Consider any sequence of edits such that, after time t, authors in U only contribute r-useless versions. Notice that the reputation of an author $u \in U$ can only grow when u contributes a version v_j, and the triple (i, j, k) is considered for feedback, where $0 < i < j < k$. There are two cases. If $r_k(a_i) \leq r$ or $r_k(a_k) \leq r$, then by (3) and $r_k(u) < r$ we have that the reputation of u cannot increase above r. If $r_k(a_i) > r$ and $r_k(a_k) > r$, then since the version a_j is r-useless, we have that $\mathsf{qual}(v_i, v_j, v_k) < 0$, leading to $\mathit{Inc}(i, j, k) < 0$, so that the reputation of u again cannot increase above r. ∎

3.2 Allowing global reputation growth

While the REPUTATION-CAP algorithm is effective against Sybil attacks, it has one major drawback: if applied it throughout the lifetime of a wiki, it would prevent the maximal reputation of wiki authors from growing. In particular, our basic content-driven reputation system starts by assigning reputation 0 to all authors. If we applied the REPUTATION-CAP algorithm from the beginning, authors reputations would not be allowed to grow.

We note, first of all, that this drawback is pertinent to growing wikis. The REPUTATION-CAP algorithm is well suited to mature wikis, such as the Wikipedia in the major languages (English, German, and French being the largest), which have a large pool of authors who have reputation very close to the top value $T_{\max}$. In mature wikis, high-reputation authors can increase the reputation of other authors, and the pool of high-reputation authors would most likely be self-renovating.

Nevertheless, we wish to obtain a reputation algorithm that is not only resistant to Sybil attacks, but that can also be used from the inception. To this end, we modify the REPUTATION-CAP ALGORITHM, obtaining the REPUTATION-CAP-NIX algorithm. The modified algorithm is based on the following idea. On the Wikipedia, it is very unlikely that low-quality or vandalistic edits survive for long time; indeed, according to some studies, vandalism has a very high probability of being removed from a page in a few minutes [17, 10, 13]. Therefore, we assume that any useless version will cause its author a negative reputation increment within a short interval of time. If a version survives for long enough without having ever accumulated negative feedback, then the version is unlikely to be part of a Sybil attack, and we revert to the basic algorithm, which enables the reputation of an author to grow, even though the author's contributions are only compared with the contributions of lower reputation authors.

The REPUTATION-CAP-NIX algorithm takes as input a delay value $T > 0$, called the *validation interval*. When a version v_j is created, we set its *nix* bit to 0, indicating that v_j has not received any negative feedback. When a version v_k is entered, the REPUTATION-CAP-NIX algorithm considers again all triples (i, j, k) with $0 < i < j < k$ and $k - i \leq m$; when considering (i, j, k), it proceeds as follows:

1. If one of these two conditions holds, set the nix bit to 1; otherwise, leave it unchanged:

   (Nix1): $t_k - t_j \leq T$ and $\mathsf{qual}(v_i, v_j, v_k) < 0$ (Nix2): $k - i \geq m$ and $t_k - t_i \leq T$.

2. If the nix bit of t_j is 1 or $t_k - t_j \leq T$, we update the reputation of a_j using the REPUTATION-CAP ALGORITHM, that is, by the amount given in (3).

3. If the nix bit of t_j is 0 and $t_k - t_j > T$, we update the reputation of a_j using the basic algorithm, that is, by the amount given in (2).

Condition (Nix1) states that, if a revision received negative feedback within time T, we set its nix bit: thus, the version will not benefit from the more liberal basic algorithm after time T has elapsed. As we will

assume that visits from high-reputation authors are spaced less than T, this helps prevent reputation increase when no useful work is performed. The condition (Nix2) has to do with the fact that we consider only triples (i, j, k) with $k - i \leq m$, so that our evaluation algorithm has a finite horizon. If we omitted clause (Nix2), then an author could perform a *stuffing attack*, immediately preceding each of her contributions by m contributions of a sock-puppet, and avoiding in this way the nixing bit to be set via (Nix1).

The following lemma states that, if high-reputation users regularly edit the article the REPUTATION-CAP-NIX algorithm provides the same guarantees against Sybil attacks as the REPUTATION-CAP algorithm.

Lemma 1 *Assume that an article p is edited in such a way that each time interval of length T contains at least one edit by a user of reputation at least r. Consider a set U of authors, which at time t all have reputation below r. If, after time t, the authors in U only contribute r-useless versions, the reputation of no authors in U can grow above r.*

Proof. An author $a \in U$ can increase her reputation when a triple (i, j, k) is considered, with $a_j = a$. We distinguish two cases.

1. The triple (i, j, k) is such that $r_k(a_k) > r$ and $r_k(a_i) > r$. By hypothesis, we have *Inc*$(i, j, k) < 0$, so this cannot increase the reputation of $a = a_j$.

2. The triple (i, j, k) is such that $\min(r_k(a_i), r_k(a_k)) \leq r$. If *Inc*$(i, j, k) \leq 0$, the result follows. If *Inc*$(i, j, k) > 0$, we distinguish two sub-cases:

   (a) If $t_k - t_j \leq T$, then the reputation update (3) is used, preventing a_j's reputation from growing above $\min(r_k(a_i), r_k(a_k)) \leq r$.

   (b) If $t_k - t_j > T$, then due to the hypothesis on the edit frequency by authors of reputation at least r, there must have been two versions v_h and v_l, with $t_l - t_h < T$, $h < j < l < k$, and with $r_l(a_l) > r$ and $r_h(a_h) > r$. We consider two cases:

      i. If $l - h \leq m$, then since v_j is r-useless, we have *Inc*$(h, j, l) < 0$, and the nix bit of v_j has been set due to (Nix1).

      ii. If $l - h > m$, then the nix bit of v_j has been set due to (Nix2).

      In either case, the nix bit of v_j is set, so that the reputation increment to $a = a_j$ is given by (3), ensuring once more that the reputation of a does not increase beyond r.

This analysis leads to the result. ∎

The results of the lemma can be extended to the case in which high-reputation authors *check* the article regularly, editing it only if desired. Precisely, we say that an author a *checks* the article at time t if a reads the version v of the article, decides what would be the best version v', and inserts v' in the system whenever $v \neq v'$. The following theorem summarizes the main properties of the REPUTATION-CAP-NIX algorithm. The first part of the theorem extends the previous lemma, replacing the assumption that high-reputation users regularly *edit* the article with the weaker one that they regularly *check* the article. The second part of the theorem ensures the *global-reputation-growth* property (i.e., maximum reputation of all wiki users can increase), making this algorithm suited to wikis in which there is no established group of high-reputation users yet. We now formally state the global reputation growth property.

> **Global-reputation-growth property.** Consider the set U of all authors editing an article p from times t to t', and assume that at time t, all users in U have reputation below $r \in [0, T_{\max}]$. If $t' - t > T$, then the reputations of authors in U can increase above r by time t'.

Theorem 2 (properties of the REPUTATION-CAP-NIX **algorithm)** *The following assertions hold.*

1. *Assume that an article p evolves in such a way that each time interval of length T contains at least one edit or check by a user of reputation at least r. Then the* REPUTATION-CAP-NIX *algorithm ensures the no-free-increase property.*

2. *The* REPUTATION-CAP-NIX *algorithm has the global-reputation-growth property.*

Proof. An author $a \in U$ can increase her reputation when a triple (i, j, k) is considered, with $a_j = a$, and we distinguish two cases.

1. The triple (i, j, k) is such that $r_k(a_k) > r$ and $r_k(a_i) > r$. As before, by hypothesis we have *Inc*$(i, j, k) < 0$, so that the reputation of $a = a_j$ cannot increase.

2. The triple (i, j, k) is such that $\min(r_k(a_i), r_k(a_k)) \leq r$. If *Inc*$(i, j, k) \leq 0$, the result follows. If *Inc*$(i, j, k) > 0$, we distinguish two sub-cases:

   (a) If $t_k - t_j \leq T$, then the reputation update (3) is used, preventing a_j's reputation from growing above $\min(r_k(a_i), r_k(a_k)) \leq r$.

   (b) If $t_k - t_j > T$, then due to the hypothesis on the check frequency by authors of reputation at least r, there must have been two times t' and t'', with $t'' - t' < T$, and with $t' < t_j < t'' < t_k$, where authors of reputation at least r checked the article. This means that there were two versions v_h and v_l, with $t_h \leq t' < t_j \leq t_l \leq t'' < t_k$, such that users of reputation above r agreed with t_h and t_l. We consider two cases:

      i. If $l - h \leq m$, then since by hypothesis a_j did r-useless work, we have *Inc*$(h, j, l) < 0$. Furthermore, from $t'' - t' < T$ and $t' < t_j \leq t_l \leq t''$ we derive $t'' - t_j < T$, so that the nix bit of v_j has been set due to (Nix1).

      ii. If $l - h > m$, notice that v_h is the version immediately preceding time t', and $t_l - t' < T$. This means that there must be at least m versions between times t_l and t', and the nix bit of v_j has been set due to (Nix2).

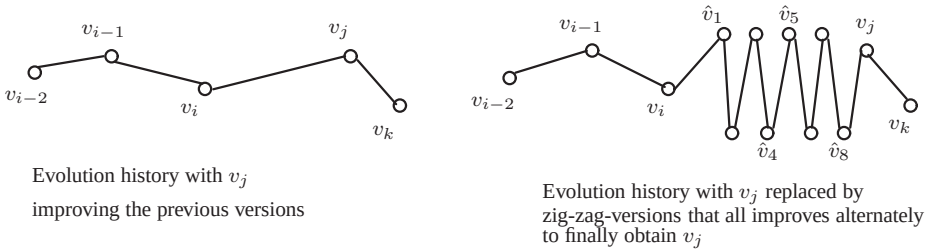
   In either case, the nix bit of v_j is set, so that the reputation increment to $a = a_j$ is given by (3), ensuring once more that the reputation of a does not increase beyond r.

For the second part of the theorem, note that if the authors in U do useful work, leading to positive ratios (1), then the nix bit of their contributions will not be set, so that the BASIC algorithm may be used, allowing their reputations to eventually grow above r. ∎

3.3 Truthfulness

We say an algorithm for reputation computation enjoys the *truthfulness* property if an author who wishes to perform an edit cannot gain by splitting the edit into multiple edits, or by employing complex editing schemes, as compared to truthfully performing the edit in a single step. We first show that the REPUTATION-CAP and the REPUTATION-CAP-NIX algorithm can be subject to a *zig-zag-attack* that violates the truthfulness property.

The Zig-Zag-Attack. Consider an author a with reputation r at time t such that the author can perform an r-useful edit e_j to produce a version v_j. When the version v_j is judged by a later high reputed author of version v_k, and compared against v_{j-1} (i.e., the triple (v_{j-1}, v_j, v_k) is considered for reputation increment), then the author a gains in reputation. However, the author can split the edit e_j to produce versions



Evolution history with v_j improving the previous versions

Evolution history with v_j replaced by zig-zag-versions that all improves alternately to finally obtain v_j

Figure 2: The zig-zag-attack.

$\hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{v}_4, \ldots, \hat{v}_f = v_j$ in a zig-zag-fashion (as shown in Figure 2) such that for all $1 \leq i \leq f$ we have $d(\hat{v}_i, v_k) \leq d(v_{j-1}, v_k)$. Hence the author gains reputation from v_k for each split sub-edit, and the sum of the reputation increment of the split sub-edits can exceed the reputation increment for the single edit e_j.

The LOCAL **algorithm.** The zig-zag-attack against truthfulness for the REPUTATION-CAP (and the REPUTATION-CAP-NIX) algorithm relied on the fact that the algorithms considers triple (v_i, v_j, v_k) for reputation increment that takes care of the *global* improvement of the article. These algorithms ignores the *local* effect: that is how an article is improved by an edit as compared to the immediate previous version. We remedy this in the LOCAL algorithm by considering only the local feedback. The LOCAL algorithm follows the REPUTATION-CAP-NIX algorithm: when a version v_k is entered, it considers all triples (i, j, k) with $0 < i < j < k$ and $k - i \leq m$, as in that algorithm. However, (2) is modified as follows:

$$\mathit{IncLocal}^p(i,j,k) = \begin{cases} 0 & i \neq j-1 \\ \mathit{Inc}^p(i,j,k) & \text{otherwise} \end{cases} \tag{4}$$

Thus, while the LOCAL algorithm follows REPUTATION-CAP-NIX for the use of the nix bits, it only increases reputation when the revision being evaluated is compared with the immediately preceding one. Since the LOCAL algorithm considers only a different set of triples as compared to the REPUTATION-CAP-NIX ALGORITHM for reputation increment, but follows the same procedure as the REPUTATION-CAP-NIX algorithm, a theorem corresponding to Theorem 2 holds.

Theorem 3 (robustness of the LOCAL **algorithm)** *The following assertions hold.*

1. *Assume that an article p evolves in such a way that each time interval of length T contains at least one edit or check by a user of reputation at least r. Then the* LOCAL *algorithm ensures the no-free-increase property.*
2. *The* LOCAL *algorithm has the global-reputation-growth property.*

We now show the truthfulness property of the LOCAL algorithm.

Theorem 4 (truthfulness property of the LOCAL **algorithm).** *Consider an author A in control of a set U of authors with maximal reputation r at time t_j. Let σ be the evolution history of an article such that A performs an r-useful edit e_j producing a version v_j. Consider an alternative evolution history σ' of the article in which the edit e_j is split into multiple edits performed by identities in U, and otherwise the evolution history σ' coincide with σ. If the* LOCAL *algorithm is followed, then A does not gain more maximal reputation in σ' as compared to σ.*

Proof. We consider the two evolution histories σ and σ'. In σ' the edit e_j is replaced by multiple edits by identities in U, otherwise σ and σ' coincide. Let v_{j-1} be the version before the edit e_j in σ, and we denote

the versions produced by edits of identities in U in σ' as $\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_f = v_j$ (i.e., the final version $\hat{v}_f$ of the edits by U is the version v_j of σ). The following analysis shows that the maximal reputation gain in σ' is no more than the maximal reputation gain in σ. Consider a version $\hat{v}_i$ produced by an identity in $u \in U$ with maximal reputation r. Let the version $\hat{v}_i$ be judged by a later version v. If v is produced by an edit of an identity in U, then the reputation of the author of the version judging $\hat{v}_i$ is at most r (since the maximal reputation of identities in U is r). Thus the reputation of u does not increase. Hence we consider the case when the judging version is a version v_k after v_j, and show the total reputation increment in σ' is bounded by the reputation increment in σ. We consider a triple (v_{j-1}, v_j, v_k) for reputation increment in σ. The sum γ of the reputation increments in σ' for edits by U producing v_j as judged by v_k is given as follows:

$$\begin{aligned}\gamma &= c_s \cdot d(v_{j-1}, \hat{v}_1) \cdot \mathsf{qual}(v_{j-1}, \hat{v}_1, v_k) \cdot w(r_k(a_k)) + c_s \cdot \textstyle\sum_{l=1}^{f-1} d(\hat{v}_l, \hat{v}_{l+1}) \cdot \mathsf{qual}(\hat{v}_l, \hat{v}_{l+1}, v_k) \cdot w(r_k(a_k)) \\ &= c_s \cdot w(r_k(a_k)) \cdot \big(d(v_{j-1}, v_k) - d(\hat{v}_1, v_k)\big) + c_s \cdot w(r_k(a_k)) \cdot \textstyle\sum_{l=1}^{f-1} \big(d(\hat{v}_l, v_k) - d(\hat{v}_{l+1}, v_k)\big) \\ &= c_s \cdot w(r_k(a_k)) \cdot \big(d(v_{j-1}, v_k) - d(\hat{v}_f, v_k)\big).\end{aligned}$$

We obtain the first equality by applying (4) for reputation increment, and the second equality follows since

$$\mathsf{qual}(v_{j-1}, \hat{v}_1, v_k) = \frac{d(v_{j-1}, v_k) - d(\hat{v}_1, v_k)}{d(v_{j-1}, \hat{v}_1)}; \qquad \mathsf{qual}(\hat{v}_l, \hat{v}_{l+1}, v_k) = \frac{d(\hat{v}_l, v_k) - d(\hat{v}_{l+1}, v_k)}{d(\hat{v}_l, \hat{v}_{l+1})}.$$

Since $\hat{v}_f = v_j$, it follows that the above sum is equal to $c_s \cdot w(r_k(a_k)) \cdot \big(d(v_{j-1}, v_k) - d(v_j, v_k)\big)$. In the evolution history σ, the reputation increment for the triple $(j-1, j, k)$ is given by

$$\begin{aligned}\mathit{IncLocal}^p(j-1, j, k) &= c_s \cdot d(v_{j-1}, v_j) \cdot \mathsf{qual}(v_{j-1}, v_j, v_k) \cdot w(r_k(a_k)) \\ &= c_s \cdot \big(d(v_{j-1}, v_k) - d(v_j, v_k)\big) \cdot w(r_k(a_k)).\end{aligned}$$

It follows that $\gamma = \mathit{IncLocal}^p(j-1, j, k)$. It follows that the maximal reputation in σ' for indentities in U is no more than the maximal reputation in σ. We remark that it is possible that a triple $(j-1, j, k)$ is considered for reputation increment for the evolution history σ, but all sub-edits by identities in U that produce v_j in σ' do not get reputation increment being judged by v_k. This is because since in σ' multiple edits produce v_j, the number of edits between an edit by U and v_k may exceed m in σ', whereas the number of edits between v_{j-1} and v_k may be smaller than m in σ. Hence the maximal reputation in σ can exceed the maximal reputation in σ'. ∎

Theorem 4 shows that if an author wishes to do a η-units of useful work, then the most rational policy to gain reputation is to truthfully do the η-units of useful work at once. Thus by Theorem 3 and Theorem 4 we obtain that the LOCAL algorithm has two highly desired properties: robustness against sock-puppet attacks, and truthfullness for useful work. However, the algorithm can be subject to *denial-of-reputation* attack.

Denial-of-reputation attack. In the REPUTATION-CAP-NIX ALGORITHM, for edits that are nixed or have not crossed the validation interval, the reputation cannot increase beyond the minimum of the two judging versions. In the LOCAL algorithm, one judging point of a version is fixed as the immediate previous version. Hence low reputed users can perform many edits, and ensure that the following useful edits are not credited with reputation increment. We remedy this partially in the following algorithm.

The LOCAL-GLOBAL **algorithm.** The REPUTATION-CAP-NIX algorithm was subject to the zig-zag-attack because it only considered the global feedback, whereas the LOCAL algorithm is subject to denial-of-reputation attack since it only considered the local feedback with respect to the immediate previous version. The LOCAL-GLOBAL algorithm considers both the local and global feedback as follows: the algorithm like

the REPUTATION-CAP-NIX ALGORITHM considers triples of the form (i,j,k) with $0 < i < j < k$, and $k - i \leq m$, but instead of the global feedback the reputation increment *Inc* is modified to be the minimum of the feedback of the global and local effect. Formally, for a triple (i,j,k) and an article p we modify (2) to the following equation:

$$\textit{IncLocalGlobal}^p(i,j,k) = c_s \cdot d(v^p_{j-1}, v^p_j) \cdot \min\left(\mathsf{qual}(v^p_{j-1}, v^p_j, v^p_k), \mathsf{qual}(v^p_i, v^p_j, v^p_k)\right) \cdot w(r^p_k(a_k)) \ . \quad (5)$$

In (5), instead of the global feedback $\mathsf{qual}(v^p_i, v^p_j, v^p_k)$ of (2), the minimum of the global feedback $\mathsf{qual}(v^p_i, v^p_j, v^p_k)$ and the local feedback $\mathsf{qual}(v^p_{j-1}, v^p_j, v^p_k)$ is used. The LOCAL-GLOBAL algorithm follows the REPUTATION-CAP-NIX algorithm replacing *Inc* by *IncLocalGlobal* for reputation increment. Observe that for all triples (i,j,k) we have $\textit{IncLocalGlobal}^p(i,j,k) \leq \textit{Inc}^p(i,j,k)$. Thus the LOCAL-GLOBAL algorithm always assigns a reputation lower as compared to the REPUTATION-CAP-NIX algorithm and hence the robustness property of the REPUTATION-CAP-NIX algorithm against sock-puppet attacks also holds for the LOCAL-GLOBAL algorithm (i.e., Theorem 3 holds for LOCAL-GLOBAL algorithm).

Almost-truthfulness of the LOCAL-GLOBAL **algorithm.** We now argue that the LOCAL algorithm ensures truthfulness in all practical cases. We can show that if an edit e_j is split as in the analysis of Theorem 4, then the reputation increment in the LOCAL-GLOBAL algorithm for the split edits is bounded by the local feedback of the single edit.

Lemma 2 (almost-truthfulness property of the LOCAL-GLOBAL **algorithm).** *Consider an author A in control of a set U of authors with maximal reputation r at time t_j. Let σ be the evolution history of an article such that A performs an r-useful edit e_j producing a version v_j. Consider an alternative evolution history σ' of the article in which the edit e_j is split into multiple edits performed by identities in U, and otherwise the evolution history σ' coincide with σ. If the* LOCAL-GLOBAL *algorithm is followed, then the reputation increment for the multiple edits is bounded by the reputation increment of v_j for local feedback.*

Proof. We consider the two evolution histories σ and σ'. In σ' the edit e_j is replaced by multiple edits by identities in U, otherwise σ and σ' coincide. Let v_{j-1} be the version before the edit e_j in σ, and we denote the versions produced by edits of identities in U in σ' as $\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_f = v_j$ (i.e., the final version $\hat{v}_f$ of the edits by U is the version v_j of σ). The following analysis shows that the maximal reputation gain in σ' is no more than the maximal reputation gain in σ. Consider a version $\hat{v}_i$ produced by an identity in $u \in U$ with maximal reputation r. Let the version $\hat{v}_i$ be judged by a later version v. If v is produced by an edit of an identity in U, then the reputation of the author of the version judging $\hat{v}_i$ is at most r (since the maximal reputation of identities in U is r). Thus the reputation of u does not increase. Hence we consider the case when the judging version is a version v_k after v_j, and show the total reputation increment in σ' is bounded by the reputation increment by local feedback in σ. We consider a triple (v_i, v_j, v_k) for reputation increment in σ, with $i < j < k$, and $k - i \leq m$. The sum γ of the reputation increment in σ' for edits by U producing v_j as judged by v_k is given as follows:

$$\begin{aligned}
\gamma &= c_s \cdot d(v_{j-1}, \hat{v}_1) \cdot \mathsf{qual}(v_{j-1}, \hat{v}_1, v_k) \cdot w(r_k(a_k)) + c_s \cdot \textstyle\sum_{l=1}^{f-1} d(\hat{v}_l, \hat{v}_{l+1}) \cdot \mathsf{qual}(\hat{v}_l, \hat{v}_{l+1}, v_k) \cdot w(r_k(a_k)) \\
&= c_s \cdot w(r_k(a_k)) \cdot \big(d(v_{j-1}, v_k) - d(\hat{v}_1, v_k)\big) + c_s \cdot w(r_k(a_k)) \cdot \textstyle\sum_{l=1}^{f-1} \big(d(\hat{v}_l, v_k) - d(\hat{v}_{l+1}, v_k)\big) \\
&= c_s \cdot w(r_k(a_k)) \cdot \big(d(v_{j-1}, v_k) - d(\hat{v}_f, v_k)\big).
\end{aligned}$$

We obtain the first equality because in (5) the reputation increment is bounded by the local feedback, and the second equality follows since

$$\mathsf{qual}(v_{j-1}, \hat{v}_1, v_k) = \frac{d(v_{j-1}, v_k) - d(\hat{v}_1, v_k)}{d(v_{j-1}, \hat{v}_1)}; \qquad \mathsf{qual}(\hat{v}_l, \hat{v}_{l+1}, v_k) = \frac{d(\hat{v}_l, v_k) - d(\hat{v}_{l+1}, v_k)}{d(\hat{v}_l, \hat{v}_{l+1})}.$$

Algorithm	ALGO-07	BASIC	REPUTATION-CAP-NIX	LOCAL	LOCAL-GLOBAL
Precision	31.7 %	30.5 %	31.7 %	29.8 %	31.5 %
Recall	93.1 %	93.2 %	92.9 %	93.4 %	93.1 %

Table 1: Precision and recall of low reputation for bad edits.

Since $\hat{v}_f = v_j$, it follows that the above sum is equal to $w(r_k(a_k)) \cdot \big(d(v_{j-1}, v_k) - d(v_j, v_k)\big)$. In the evolution history σ, the reputation increment for the triple $(j-1, j, k)$ is given by

$$\begin{aligned} \textit{IncLocal}^p(j-1,j,k) &= c_s \cdot d(v_{j-1}, v_j) \cdot \mathsf{qual}(v_{j-1}, v_j, v_k) \cdot w(r_k(a_k)) \\ &= c_s \cdot \big(d(v_{j-1}, v_k) - d(v_j, v_k)\big) \cdot w(r_k(a_k)). \end{aligned}$$

It follows that $\gamma = \textit{IncLocalGlobal}^p(j-1, j, k)$. Since $c_s \cdot w(r_k(a_k)) \cdot \big(d(v_{j-1}, v_k) - d(v_j, v_k)\big)$ is the local feedback for v_j compared against v_k, the desired result follows. ∎

Hence if the global feedback reputation increment exceeds the local feedback increment, and the LOCAL-GLOBAL algorithm is followed, then the rational policy for reputation increment is the truthful policy. The only case when an edit can possibly benefit from splitting is as follows: the immediate previous edit is from a high reputed user, but in a wrong direction as compared to the following edits of the high reputed user (i.e., a high reputed user performs a bad edit for the article). In this case, the global feedback is lower as compared to the local feedback, and since the immediate previous edit is from a high reputed user, the REPUTATION-CAP-NIX algorithm also allows for reputation increment. However, we argue that the case when the LOCAL-GLOBAL algorithm violates the truthfulness property is rare and hard to implement for an user. First, it is rare that a high reputed user performs a bad edit for an article, and second, since the reputation of authors is not public, an author who wishes to make an edit does not know whether the previous bad edit was from a high reputed user. Hence for all practical purposes the LOCAL-GLOBAL algorithm is truthful, robust against sock-puppet and denial-of-reputation attacks.

4 Evaluation

The robust reputation algorithms we proposed in this paper have not been deployed yet on a large and dynamic wiki, so that it is not possible at this point to report on their real-world behavior. While the theorems presented in this paper provide absolute guarantees of robustness, only a real-world deployment will make it possible to judge the impact of the algorithms on user satisfaction, and quality of on-line collaboration.

Our present evaluation focuses on the *quality* of the reputation computed by the algorithms: specifically, we show that the changes required to obtain robust algorithms do not lead to lower-quality reputation. Following [2], we evaluate the quality of content-driven reputation via its ability to predict the quality of future contibutions. We consider all edits e_j^p in the history of a wiki, and we study the correlation between the reputation $r_j^p(a_j^p)$ of the author of e_k^p at the time t_k^p when the edit was made, and the future *longevity* of e_j^p, defined as in [2] by:

$$\textit{Long}(e_j^p) = \frac{1}{m-1} \sum_{k=j+1}^{j+m-1} \mathsf{qual}(v_{j-1}^p, v_j^p, v_k^p) \ .$$

The longevity of e_j^p is a measure of how long the change introduced in e_j^p lasts in the future. As the reputation $r_j^p(a_j^p)$ is accrued in the past of e_j^p, the correlation between $r_j^p(a_j^p)$ and $\textit{Long}(e_j^p)$ provides a meaningful statistical quality criterion for our content-driven reputation. Following [2], we say that e_j^p is *short-lived* if

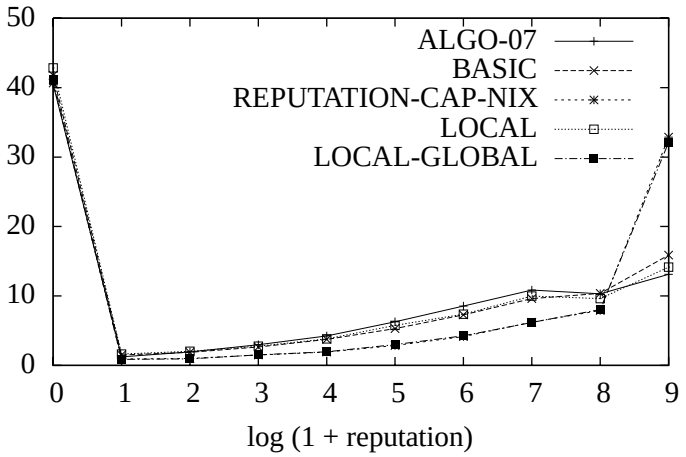


Figure 3: Percentage of edits from authors of a given reputation range. The large number of edits from reputation 0 are due to novices and anonymous users. Data from a 100,000-article sample of the French Wikipedia, up to March 2008.

Long$(e_j^p) \leq -0.8$, indicating that the edit has been almost entirely reverted, and we say that $r_j^p(a_j^p)$ is *low-reputation* if $r_j^p(a_j^p) \leq 0.2 \cdot T_{\max}$, that is, if the author is in the lowest 20% percentile at the time of the edit. To estimate the quality of the reputation systems, we assign to each edit e_j^p the relative weight $d(v_{j-1}^p, v_j^p)$, and we consider the precision and recall that low-reputation provides with respect to short-lived edits:

- The *precision* is the probability that e_j^p is short-lived, given that $r_j^p(a_j^p) \leq 0.2 \cdot T_{\max}$;
- The *recall* is the probability that $r_j^p(a_j^p) \leq 0.2 \cdot T_{\max}$, given that e_j^p is short-lived.

We have evaluated the performance of the proposed reputation algorithms over 100,000 articles of the French Wikipedia, corresponding to 56,229,855 revisions, with end date March 23, 2008.[2] The nix interval was 1 day, and 0.07% revisions were nixed. The algorithm ALGO-07 is the one of [2]. Table 1 shows precision and recall measurements for the basic reputation algorithm, and for the robust versions. We see that the performance of the algorithms is only slightly affected by the changes that are required to make them resistant to attack. The graphs in Figure 3 give the distribution of author reputation The main difference among the algorithms is that the algorithms which consider only triples of the form $(j-1, j, k)$ for $1 < j < k$ with $k - j \leq m - 1$ confer less reputation to users than the algorithms that consider triples of the form (i, j, k), for all $0 < i < j < k$ with $k - i \leq m$. This is simply due to the fact that the latter algorithms consider more triples, in total, to update the reputation value of a version author. The performance of the algorithms can thus be equalized simply by choosing different re-scaling factors c_s for the algorithms.

[2]The data for the whole French Wikipedia will be available to us soon; we do not expect significant changes due to the size of this sample.

References

[1] B.T. Adler, J. Benterou, K. Chatterjee, L. de Alfaro, I. Pye, and V. Raman. Assigning trust to Wikipedia content. Technical Report UCSC-CRL-07-09, School of Engineering, University of California, Santa Cruz, CA, USA, 2007.

[2] 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.

[3] Wikipedia article:. Flagged revisions / Sighted versions, 2008.

[4] 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.

[5] E.H. Clarke. Multipart pricing of public goods. *Public Choice*, 8:17–33, 1971.

[6] G. Cormode and S. Muthukrishnan. The string edit distance matching problem with moves. *ACM Trans. Algorithms*, 3(1):2, 2007.

[7] 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.

[8] T. Groves. Incentive in teams. *Econometrica*, 41(4):617–631, 1973.

[9] K. Hoffman, D. Zage, and C. Nita-Rotaru. A survey of attack and defense techniques for reputation systems. Technical Report CSD TR #07-013, Purdue University, 2007.

[10] A. Kittur, B. Suh, B.A. Pendelton, and E.H. Chi. He says, she says: Conflict and coordination in Wikipedia. In *Proc. of CHI*, 2007.

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

[12] M.J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[13] R. Priedhorsky, J. Chen, S. K. Lam, K. Panciera, L. Terveen, and J. Riedl. Creating, destroying, and restoring value in wikipedia. In *GROUP '07: Proceedings of the 2007 international ACM conference on Supporting group work*, pages 259–268, New York, NY, USA, 2007. ACM.

[14] J.-M. Seigneur, A. Gray, and C.D. Jensen. Trust transfer: Encouraging self-recommendations without sybil attack. In *Trust Management*, volume 3477 of *Lect. Notes in Comp. Sci.* Springer-Verlag, 2005.

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

[16] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[17] F. Viégas, M. Wattenberg, and K. Dave. Studying cooperation and conflict between authors with history flow visualizations. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 575–582, 2004.

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