Assigning Trust to Wikipedia Content

B. Thomas Adler[1] Jason Benterou[1] Krishnendu Chatterjee[2] Luca de Alfaro[3]
Ian Pye[1] Vishwanath Raman[1]

[1]Computer Science Dept.
UC Santa Cruz, CA, USA
{thumper, ipye, vraman, jbentero}@ucsc.edu

[2]EECS Dept.
UC Berkeley, CA, USA
c_krish@eecs.berkeley.edu

[3]Computer Engineering Dept.
UC Santa Cruz, CA, USA
luca@soe.ucsc.edu

November 2007

Technical Report UCSC-CRL-07-09
School of Engineering, University of California, Santa Cruz, CA, USA

Abstract

The Wikipedia is a collaborative encyclopedia: anyone can contribute to its articles simply by clicking on an "edit" button. The open nature of the Wikipedia has been key to its success, but has also created a challenge: how can readers form an informed opinion on its reliability? We propose a system that computes quantitative values of trust for the text in Wikipedia articles; these trust values provide an indication of text reliability.

The system uses as input the revision history of each article, as well as information about the *reputation* of the contributing authors, as provided by a reputation system. The trust of a word in an article is computed on the basis of the reputation of the original author of the word, as well as the reputation of all authors who edited the text in proximity of the word. The algorithm computes word trust values that vary smoothly across the text; the trust values can be visualized using varying text-background colors. The algorithm ensures that all changes to an article text are reflected in the trust values, preventing surreptitious content changes.

We have implemented the proposed system, and we have used it to compute and display the trust of the text of thousands of articles of the English Wikipedia. To validate our trust-computation algorithms, we show that text labeled as low-trust has a significantly higher probability of being edited in the future than text labeled as high-trust. Anecdotal evidence seems to corroborate this validation: in practice, readers find the trust information valuable.

1 Introduction

Collaborative, user-generated content is increasing in importance on the Web, with sites such as the Wikipedia, flickr.com, tripadvisor.com, and epinions.com relying on it for almost all their content. The creation of collaborative content does not depend on a restricted, pre-selected set of contributors, but rather can leverage the interests and abilities of people the world over. Exploiting this, many collaborative sites have experienced explosive growth rates that would have been impossible with more traditional ways of accruing content. The flip side of this is that the content of collaborative sites is of heterogeneous origin and quality, so that it can be difficult for visitors to assess the reliability of the content, and to "separate the wheat from the chaff". In this paper, we present an attempt to improve the situation in the context of the Wikipedia, via a trust system that provides Wikipedia visitors with a guide to the "trust" they can place in the text of the articles they read.

The Wikipedia is one of the most prominent collaborative sites on the Web. This on-line encyclopedia, available in multiple languages, has grown entirely due to user-contributed content, with contributors ranging from casual visitors to dedicated, volunteer, editors. This user-generated growth is at the basis of Wikipedia's remarkable breadth: as of October, 2007, the Wikipedia consisted of over two million articles, compared with approximately 120,000 for the online Encyclopedia Britannica [29]. On the other hand, the open process that gives rise to Wikipedia content makes it difficult for visitors to form an idea of the reliability of the content. Wikipedia articles are constantly changing, and the contributors range from domain experts, to vandals, to dedicated editors, to superficial contributors not fully aware of the quality standards the Wikipedia aspires to attain. Wikipedia visitors are presented with the latest version of each article they visit: this latest version does not offer them any simple insight into how the article content has evolved into its most current form, nor does it offer a measure of how much the content can be relied upon.

We introduce in this paper a system that computes trust values for the text of Wikipedia articles. The trust values

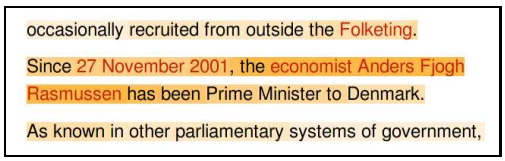

Figure 1: An example of coloring words by trust, from the Wikipedia article on *Politics of Denmark*, after the prime minister's middle name has been changed from "Fogh" to "Fjogh" by a low reputation user (revision id: 77692452).

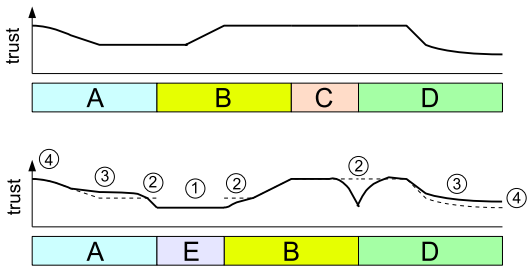

Figure 2: Update process for text trust. The text is shown before (top) and after (bottom) an edit, together with its trust. In the bottom figure, the new values of trust (continuous line) are obtained from the inherited values of trust (dashed line) as follows: 1: Trust value for newly inserted text (E). 2: Edge effect: the text at the edge of blocks has the same trust as newly inserted text. 3: Revision effect: old text may increase in trust, if the author reputation is higher than the old text trust. 4: The edge effect is applied at the beginning and end of the article only if text changes there (which is not the case here).

have two functions: they provide an indication of text reliability, and they flag recent text changes that have been insufficiently reviewed. While these two purposes are related, they do not coincide. For example, when text is deleted from an article, our trust values provide an indication that the deletion has occurred, in spite of the fact that the reliability of the remaining text may be unaffected. The first function of trust, as an indicator of reliability, has been studied in [19, 31]; the second function, text change flagging, has been discussed in [4]. Our evaluation indicates that the trust we compute is a good predictor of future text stability, a proxy we use for "truth". Following [19, 4, 31], we display the trust of Wikipedia text by coloring the text background: the color gradation goes from white, for fully trusted words, to orange, for fully untrusted ones. This coloring provides Wikipedia visitors with an intuitive guide to the portions of the articles they are reading that are most in need of critical scrutiny. An example of this coloring is shown in Figure 1; a demo of the trust coloring is available from `http://trust.cse.ucsc.edu/`.

1.1 Assigning Trust to Text

Our system computes the trust of text in an article on the basis of the reputations of the authors who, across time, contributed to the article [19, 31]. While we could have developed our trust system relying on various sources of author reputation (see, e.g., [12, 32, 2]), we chose to base our implementation on the *content-driven reputation system* for Wikipedia authors of [1]. In that system, authors gain reputation when their contributions prove long-lived, and they lose reputation when their contributions are reverted. The resulting author reputation has good *predictive* value: higher-reputation authors are more likely to make longer-lasting contributions to the Wikipedia.

We compute the trust of a word as a function of the reputation of the original author of the word, as well as of the reputation of all authors who edited the article in proximity of the word. Our text analysis is performed at the level of individual words: short edits constitute the majority of the edits on the Wikipedia, and many of them can have significant implications in terms of meaning. At the core of the trust system is thus a text tracking system, which follows the history of each word across the evolution of Wikipedia articles. The text tracking system is capable of dealing with deletions, and with reversions: it tracks not only the words present in each version of an article, but also the words that used to be present, and that have been deleted, so that if the words are later re-introduced, they can still be correctly attributed.

When an author of reputation r edits an article, the trust of article words is updated as follows (the precise algorithms are presented in the next section):

- New words inserted by the author have a starting reputation that is equal to $0.4 \cdot r$ (the constant 0.4, like all other constants used, was determined via an optimization process described later).
- When a block of text is moved, the edges of the block constitute discontinuities in the content, and inherit trust $0.4 \cdot r$; this *edge effect* gradually fades towards the interior of the block, whose trust value is unchanged.
- When a block of text is deleted, it is preserved as "dead text" associated with the article, in case it is later re-inserted. The trust value of the text in the block is lowered in proportion to the reputation of the author performing the deletion.
- Once the above insertion, move, and deletion effects have taken place, if the author reputation r is greater than the trust t of a word, the word trust is updated to $t + (r - t) \cdot \beta$, for $0 < \beta < 1$. We take $\beta = 0.36$ for

words in paragraphs that have been edited by the author, and $\beta = 0.2$ for other words; the latter constant being lower to model the non-uniformity of author attention during the edit.

An example depicting how text trust is updated is given in Figure 2. The algorithm has the following novel properties:

- *High trust requires consensus.* The only way text can become highly trusted is by surviving revisions by many authors, including high-reputation authors. Text just created by maximum-reputation authors has a trust value that is just 62% of the maximum trust value.[1]

- *Readers are alerted not only to content additions, but also to content re-arrangements and deletions.* When text is deleted, the margins of the "wound" where the text has been cut form the edges of text block-moves. These margins will thus be marked with less than full trust in the next version; their orange color will provide an indication that an edit has occurred (see, e.g., the deletion of block C in Figure 2). Text re-arrangements are also similarly flagged.

- *The algorithms are robust with respect to text deletion.* When deleted text is restored, its original authorship information is preserved. Moreover, if the deletion was performed by low-reputation authors, such as vandals, the original trust of the text is also restored. This ensures that vandalism does not have any long-term effect on an article.

- *The text coloring is smooth.* The rules for text insertion, and edge-effect of block moves, ensure that text within a sentence is assigned smoothly-varying amounts of trust. Smooth trust assignments result in more intuitive colorings, and reflect the fact that information, and its reliability, is not localized in single words.

We have implemented the proposed system, and we have used it to color, according to trust, thousands of long-lived articles of the English Wikipedia.[2] In our site `http://trust.cse.ucsc.edu/`, the display of Wikipedia text trust via coloring is augmented with text origin information [19]: when visitors click on a word in an article, they are redirected to the version of the article where the word was first introduced. The trust coloring and the text origin information complement each other: visitors are made aware of the less trusted portions of text by the coloring, and can then investigate the origin of such text via the text origin redirection. By using the on-line demo, we became convinced that the system provides useful information on the stability of text, highlighting text that has changed recently, and that has been insufficiently revised. A virtue of the system is that it makes it hard to maliciously and surreptitiously change the content of Wikipedia articles: every change, including text deletions, leaves a low-trust mark that fades in future revisions only as the text is further revised.

Our implementation is currently based on the batch processing of static Wikipedia dumps, made available by the MediaWiki Foundation. Our work towards an on-line system is briefly mentioned in Section 6.

[1] $0.4 + (1 - 0.4) \cdot 0.36 = 0.616 = 61.6\%$

[2] A coloring of the latest dump of the full English Wikipedia is in progress at the time of writing.

1.2 Evaluation

To evaluate the quality of our trust labeling, one idea is to measure the correlation between the trust values of the labeling, and the truth of the information encoded in the text, as assessed by human subjects. There are various problems with such a human-driven approach, however. "Truth" is a poorly-defined notion: indeed, accuracy investigations of Wikipedias and other encyclopedias have confined themselves to articles on science, where truth — or rather, scientific consensus, is easier to assess [7]. Furthermore, any human assessment of "truth" in Wikipedia articles would be very labor intensive, especially given the need for sampling a statistically significant set of article revisions. In particular, the human assessment approach would be infeasible for optimization purposes, where it is necessary to evaluate and compare multiple variants of the same algorithm.

For these reasons, we assess the quality of the trust labeling in a data-driven way, using the idea that *trust should be a predictor for text stability* [31]. If low-trust is correlated with imprecise information, then low-trust text should be more likely to be subject to edits than high-trust text, as visitors and editors seek to correct the information. We introduce four data-driven methods for measuring the quality of a trust coloring:

- *Recall of deletions.* We consider the recall of low-trust as a predictor for deletions. We show that text in the lowest 50% of trust values constitutes only 3.4% of the text of articles, yet corresponds to 66% of the text that is deleted from one version to the next.

- *Precision of deletions.* We consider the precision of low-trust as a predictor for deletions. We show that text that is in the bottom half of trust values has a probability of 33% of being deleted in the very next version, in contrast with the 1.9% probability for general text. The deletion probability raises to 62% for text in the bottom 20% of trust values.

- *Trust of average vs. deleted text.* We consider the trust distribution of all text, compared to the trust distribution to the text that is deleted. We show that 90% of the text overall had trust at least 76%, while the average trust for deleted text was 33%.

- *Trust as a predictor of lifespan.* We select words uniformly at random, and we consider the statistical correlation between the trust of the word at the moment of sampling, and the future lifespan of the word. We show that words with the highest trust have an expected future lifespan that is 4.5 times longer than words with no trust. We remark that this is a proper test, since the trust at the time of sampling depends only on the history of the word prior to sampling.

The above results were obtained by analyzing 1,000 articles selected randomly from the Wikipedia articles with at least 200 revisions. The requirement on the revision history ensures that each article has a lifespan that is sufficiently long to analyze the predictive power of trust with respect to text stability. We considered all text of all versions of the articles: overall, the selected articles contained 544,250 revisions, for a total of 13.74 GB of text. Taken together, these results indicate that the trust labeling we compute is a good predictor of future text stability.

To quantify the contribution of the reputation system to overall performance, we have compared the performance of a trust labeling that uses the reputation system of [1], with the performance of a trust labeling that does without a reputation system, and that assumes instead that everybody, from casual users to dedicated editors, has high reputation. Unsurprisingly, the trust system relying on the reputation system performed better, but the performance gap was narrower than we expected; the detailed results are presented in Section 5.

1.3 Related Work

The problem of the reliability of Wikipedia content has often emerged both in the press (see, e.g., [24, 11]) and in scientific journals [7]. The idea of assigning trust to specific sections of text of Wikipedia articles as a guide to readers has been previously proposed in [19, 4, 31], as well as in white papers [13] and blogs [18]; these papers also contain the idea of using text background color to visualize trust values.

The work most closely related to ours is [31]. That work introduced two ideas that proved influential for this work: that the trust of a piece of text could be computed from the reputation of the original author, and the reputations of the authors who subsequently revised the article, and that the quality of a trust labeling could be evaluated via its ability to predict text stability. In [31], the analisys is performed at the granularity level of sentences; all sentences introduced in the same revision form a *fragment*, and share the same trust. The trust values of the fragments are computed using a Bayesian network, which takes as input at every version the previous trust of the fragment, the amount of surviving fragment text, and the reputation of the version's author. Flagging individual text changes is not a goal of the algorithm: text insertions are not displayed at word level, and fragment reorderings and deletions are not flagged via the trust labels. Deleted text is not tracked: if text is deleted, and then re-inserted, it is counted as new. Among other things, this creates an incentive to vandalism: blanking an article suffices to reset its entire trust assignment. To validate the trust assignment, [31] computes the correlation between the trust of a fragment, and the probability that the fragment appears in the most recent version of the article. We refine this criterion into one of our evaluation criteria, namely, the predictive power of trust with respect to word longevity.

In [19], the trust of authors and fragments is then computed on the basis of the author-to-fragment and fragment-of-article graphs, together with the *link ratio* of article titles. The *link ratio* is the ratio of the number of times an article title appears as a link in other articles, and the number of times the title appears as normal text. The work provides trust values for some articles, but no comprehensive evaluation.

The white paper [13] focuses on the user interface aspects of displaying information related to trust and author contributions; we hope to include some of the suggestions in future versions of our system. Related work that relies on an analysis of revision information to infer trust has been performed in the context of software, where logs are mined in order to find revision patterns that point to possible software defects and weak points (see, e.g., [17]).

Other studies have focused on trust as article-level, rather than word-level, information. These studies can be used to answer the question of whether an article is of good quality, or reliable overall, but cannot be used to locate in an article which portions of text deserve the most careful scrutiny, as our approach can. In [32], which inspired [31], the revision history of a Wikipedia article is used to compute a trust value for the entire article. In [6, 21], metrics derived via natural language processing are used to classify articles according to their quality. In [16], the number of edits and unique editors are used to estimate article quality. The use of revert times for quality estimation has been proposed in [26], where a visualization of the Wikipedia editing process is presented; an approach based on edit frequency and dynamics is discussed in [30]. There is a fast-growing body of literature reporting on statistical studies of the evolution of Wikipedia content, including [26, 27, 22]; we refer to [22] for an insightful overview of this line of work.

The notion of trust has been very widely studied in more general contexts (see, e.g., [3, 9]. as well as in e-commerce and social networks (see e.g. [14, 23, 5, 12, 10, 8]); these notions of trust however are generally based on user-to-user feedback, rather than on algorithmic analysis of content evolution.

1.4 Paper Organization

We present the algorithm for assigning text trust in Section 2, and the methods for evaluating the quality of the resulting

trust values in Section 3. Our implementation of the trust system is described in Section 4, and the results on trust quality are given in Section 5. In Section 6 we discuss our work towards an on-line implementation of the proposed trust system for the Wikipedia.

2 Text Trust Algorithms

We base our computation of word trust on a reputation system for Wikipedia authors. The trust of a word will depend on the reputation of its original author, as well as on the reputation of all the authors who have subsequently revised the article. Thus, we compute text trust using a method modeled after the human process of text revision, in which different editors lend some of their reputation to the text they revise. In this section, all we assume of the reputation system is that whenever an author performs an edit to the Wikipedia, the reputation system provides us with a value for the reputation of the author in an interval $[0, T_{\max}]$, where the maximum reputation value $T_{\max} > 0$ is arbitrary. In the next sections, we will present experimental data on the performance of the proposed trust system that are based on the content-driven reputation system introduced in [1] by a subset of the authors.

We will present our algorithm for trust assignment in two steps. First, we will illustrate the basic idea via a simplified algorithm that does not cope with reversions, nor in general, with the situation when text is deleted, and later re-inserted. Next, we present an improved algorithm for assigning trust to Wikipedia content that deals with removed-and-reinserted text, and that also contains a tuned model of user attention during the process of article revision.

2.1 Notation

To present the algorithms, we use the following notation. We denote the sequence of revisions of a Wikipedia article by $v_0, v_1, v_2, \ldots$. Version v_0 is empty, and version v_i, for $i > 0$, is obtained by author a_i performing an edit $e_i : v_{i-1} \rightsquigarrow v_i$. When editing a versioned document, authors often save intermediate results, thus performing multiple consecutive edits. Before processing the versions, we filter them, keeping only the last of consecutive versions by the same author; we assume thus that for $1 \leq i < n$ we have $a_i \neq a_{i+1}$. Every version v_i, for $0 \leq i \leq n$, consists of a sequence $[w_1^i, \ldots, w_{m_i}^i]$ of *words*, where m_i is the number of words of v_i; we have $m_0 = 0$. Our system works at the level of the Mediawiki markup language in which authors write article content, rather than at the level of the HTML produced by the wiki engine; a *word* is a whitespace-delimited alphanumerical strings in the Mediawiki markup language.

2.2 A simplified text-trust algorithm

Our trust algorithms will assign a trust value in the interval $[0, T_{\max}]$ for each word of each article revision. We first present a simplified algorithm for trust computation. Given an edit $e_i : v_{i-1} \rightsquigarrow v_i$, a trust value $t_1, t_2, \ldots, t_{m_{i-1}}$ for each word of v_{i-1}, and a value $r \in [0, T_{\max}]$ for the reputation of the author a_i of the revision, the algorithm computes trust values $\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_{m_i}$ for all words of v_i. The algorithm first computes an *edit list* L_i detailing how v_i is obtained from v_{i-1} [25]. The edit list L_i consists of one or more of the following elements:

- $I(j, n)$: n words are inserted at position j of v_i;
- $R(j, n)$: n words are deleted at position j of v_{i-1};
- $M(j, j', n)$: n words are moved from position j in v_{i-1} to position j' in v_i.

Each word in v_i is part of exactly one of the above $I(\cdot)$ or $M(\cdot)$ elements, and the algorithm to generate edit lists tries to maximize text block matches [1]. The trust computation algorithm uses the following constants:

- $c_e > 0$ is the *edge effect constant:* it specifies how far from the edges of a block does the edge effect propagate.
- $0 \leq c_l < 1$ is the *trust inheritance constant:* it specifies how much trust should a word inherit from the reputation of its author.
- $0 \leq c_r < 1$ is the *revision constant:* it specifies how much trust does the author reputation confer to the overall text of the article (even text that was not due to the revision author).

The values of these constants are obtained via optimization techniques, with the goal of maximizing the correlation between text trust and future text stability, as detailed in later sections. We first compute preliminary trust values $t'_0, t'_1, \ldots, t'_{m_i}$ by considering all elements in the edit list L_i:

1. If $I(j, n) \in L_i$, then $t'_k := c_l \cdot r$ for all $j \leq k < j + n$: thus, inserted text is assigned a trust value equal to the reputation of the author inserting it, multiplied by the trust inheritance constant.
2. If $M(j, j', n) \in L_i$, then for all $0 \leq k < n$, k is the distance of the k-th word in the block from the beginning of the block, and $\bar{k} = n - 1 - k$ is the distance from the end of the block. We apply the edge effect to block endpoints whenever the endpoints *change context*, that is, whenever they are in contact with new text. The only situation in which block endpoints do not change context is when they are at the start (resp. end) of the article both before and after the edit. We use an exponential decay function to ensure the smooth transition between the edge-effect and block-interior behaviors.

(a) If $j \neq 0$ or $j' \neq 0$ then the left endpoint of the block has changed context, and we let:

$$t''_{j'+k} = t_{j+k} + (r - t_{j+k}) \cdot e^{-c_e k}$$

Otherwise, if $j = 0$ and $j' = 0$, we let $t''_{j'+k} = t_{j+k}$.

(b) If $j + n \neq m_{i-1}$ or $j' + n \neq m_i$, then the right endpoint has changed context, and we let:

$$t'_{j'+k} = t''_{j'+k} + (r - t''_{j'+k}) \cdot e^{-c_e \bar{k}}$$

Otherwise, if $j + n = m_{i-1}$ and $j' + n = m_i$, we let $t'_{j'+k} = t''_{j'+k}$.

If $R(j, n) \in L_i$, then the text is deleted, and there is no trust assignment to be made (the edge effect of adjacent blocks to $R(j, n)$ will take care of flagging the deletion in the new version). Once all elements of the edit list L_i have been processed, we have preliminary trust values $t'_1, t'_2, \ldots, t'_{m_i}$ which take into account of insertions, block moves, and edge effects. The final trust values $\hat{t}_0, \hat{t}_1, \ldots, \hat{t}_{m_i}$ of the words of v_i are then computed by accounting for the fact that the author a_i lends some of her reputation r to the revision v_i she just performed. For $0 \leq k < m_i$, we let:

$$\hat{t}_k = \begin{cases} t'_k & \text{if } t'_k \geq r \\ t'_k + (r - t'_k) \cdot c_r & \text{if } t'_k < r \end{cases} \quad (1)$$

The trust labeling computed by the algorithm enjoys the following properties:

- *Reordering warning.* Thanks to the edge effect, the algorithm flags with low or intermediate values of trust places where reordered blocks of text meet. Thus, readers are alerted via trust coloring when reordering has occurred, even though no new text has been added.

- *Deletion warning.* When text is removed it does not disappear silently: the "cut points" at the margin of the deletion receive trust that is lower than that of surrounding text, due to the edge effect of surviving blocks of text.

- *High trust requires consensus.* Text that is freshly inserted by an author with reputation r receives a trust of value smaller than r (in our case $0.62r$). This prevents high-reputation users from single-handedly creating trustable content: consensus and mutual revision is required in all cases to produce high-trust content.

This simplified algorithm, however, has a fatal flaw: it does not cope with text that is deleted in a revision, only to be reinserted in a later one. Deletion and reinsertion is a common phenomenon in the evolution of Wikipedia articles: it occurs in many disputes about article content, and even more devastatingly, it occurs when visitors deface articles by removing part or all of their text. The current algorithm would assign a trust to the re-inserted text that depends only on the reputation of the author who re-inserts it, thus losing all the trust that the text may have accumulated. If applied to the Wikipedia, especially in the context of an on-line system, such an algorithm would be disastrous: all a vandal would need to do to lower the trust of an article, and erase the effect of past revisions, would be to erase the whole article text — a type of attack that is common on the Wikipedia. Worse, the naïve trust algorithm presented above would give a strong incentive to such vandalism. Currently, when article text is destroyed, other authors quickly re-instate it, so that the effect on the Wikipedia, and the "reward" to the vandal, is small. However, if article trust was reset each time the article text was erased, the disruption would constitute a much larger "reward" for the vandal, and this type of vandalism would surely become more common.

2.3 An improved text-trust algorithm

We describe now an improved text-trust algorithm, which keeps track not only of the trust of the text present in an article, but also of the trust of the text that used to be present, but that has subsequently been deleted. The algorithm also models the attention focus of the author performing an edit, raising by a larger amount the trust of the text that is most likely to have been read by the author in the course of the edit.

2.3.1 Tracking deleted text.

We track deleted text by representing each article version v_i, for $1 \leq i \leq n$, as a non-empty list $C_i = [c_0^i, c_1^i, \ldots, c_{h_i}^i]$ of *chunks*, where each chunk c_k^i, for $0 \leq k \leq h_i$, is a sequence of words. The *live* chunk c_0^i corresponds to the words that are present in v_i; the *dead* chunks $c_1^i, \ldots, c_{h_i}^i$, if present, correspond to contiguous portions of text that used to be present in some prior version $v_0, \ldots, v_{i-1}$ of the article, but have been deleted. The chunks C_i are computed from the chunks $C_{i-1} = [c_0^{i-1}, c_1^{i-1}, \ldots, c_{h_{i-1}}^{i-1}]$ for v_{i-1} as described in [1]. Specifically, we match the text of v_i with the text of all the chunks in C_{i-1}, looking for the longest possible matches of contiguous sequences of words. We break ties in favor of matches between v_i and the text c_0^{i-1} that was present in v_{i-1}, thus preferring matches between v_i and the live text in v_{i-1}, to matches between v_i and the text $c_1^{i-1}, \ldots, c_{h_{i-1}}^{i-1}$ that was present before v_{i-1} but is "dead" in v_{i-1}. Furthermore, we allow the text in C_{i-1} to be matched multiple times, modeling the fact that an author can replicate existing text; the text in v_i can be matched at most once. The portions of unmatched text in C_{i-1} go on to form the new dead chunks $[c_1^{i-1}, \ldots, c_{h_{i-1}}^{i-1}]$ for v_i. In this matching process, lower bounds on the length of acceptable matches

ensure that common sequence of words (such as "the" or "in fact") appearing in new contexts are not considered as copied or re-introduced text.

We update the trust of deleted and reinserted text as follows.

- For text that is moved from the live chunk c_0^{i-1} to some dead chunk $c_{h'}^i$, $h' > 0$, we multiply the trust of the text by e^{-rc_k}. The idea is that when text is deleted, its trust is decreased in proportion to the reputation r of the author deleting the text. In particular, text does not lose trust when deleted by anonymous users or novices ($r = 0$). This ensures that when vandals remove all text of an article, once the text is re-inserted it has the same trust as before the vandalism occurred. In our implementation, we have taken $c_k = (\log 2)/T_{\max}$, so that the trust of a word is halved when deleted by an author of maximum reputation.
- For text that is moved from a dead chunk c_h^{i-1}, $h > 0$, to another dead chunk $c_{h'}^i$, $h' > 0$, we simply copy the trust.
- For text that is moved from a dead chunk c_h^{i-1}, $h > 0$, to the live chunk c_0^i, we update the trust in a manner completely equivalent to the one used for block moves $M(j, j', n)$ in the previous section, applying the edge effect to both text endpoints.

2.3.2 Modeling author attention.

In step (1) of the previous algorithm, we increase the trust of the text uniformly — this assumes that the author of the revision pays equal attention to the entire text being revised. This assumption is unlikely to be correct, as authors are more likely to pay greater attention to text that is closer to their edits; raising the trust of all the text in the article may impart too much trust to text that has not been the focus of author attention. We decided therefore to experiment with a variation of the algorithm that models author attention in a rudimentary fashion.

When parsing the text of the revision v_i, we split it into paragraphs, where section titles, items in a bulleted or numbered list, image captions, and table cell entries also count as "paragraphs". Our algorithm then follows the simple idea that authors are likely to pay more attention to the text in the same paragraph as the edits they are performing. To this end, we mark as *modified* all paragraphs where (a) either new text has been inserted (corresponding to an I element in the edit list), or (b) the paragraph contains the endpoint of a block move (elements M in the edit list) to which the edge effect applies. For modified paragraphs we apply, after (1), the following update:

$$\hat{t}_k := \begin{cases} \hat{t}_k & \text{if } \hat{t}_k \geq r \\ \hat{t}_k + (r - \hat{t}_k) \cdot c_p & \text{otherwise,} \end{cases} \quad (2)$$

where $0 \leq c_p < 1$ is the *paragraph constant*: it specifies how much additional trust the author reputation confers to the paragraph of the article she modified. Thus, text in modified paragraphs receives an additional trust increment.

3 Evaluation Criteria

There are many possible methods of associating trust with Wikipedia text. In the previous section, we have described one such method, and we have argued that, if not optimal, it is at least a reasonable attempt. The question is: how does one evaluate a trust labeling? A quantitative evaluation of a trust labeling is needed both to compare different versions of the algorithms, and to optimize the values of the various coefficients (c_e, c_l, c_r, c_k, and c_p) involved in the computation of the trust labeling.

The key idea we use to evaluate a trust labeling is that high trust should be associated with stability: if a piece of text is highly trusted, it ought to be less likely to change in future revisions than a piece of text which is labeled as low trust. By defining trust as being related to the stability of the text, we relate trust to the consensus that arises from group collaboration.

Based on this idea, we present various evaluation criteria that measure how well low-trust predicts future text changes. We note that this is a sound evaluation method: the trust labeling of a piece of text is computed entirely on the basis of the *past* history of the text;[3] thus, the correlation between text trust and future text change is entirely due to the ability of trust to be a predictor of text stability.

3.1 Low trust as a predictor of deletions

The most reliable indicator of text instability, in our experience, is text deletion. Not all text change is connected to deletions: text can also be reordered, or subject to insertions. However, when text reordering occurs, all words are preserved, and it is difficult to have an objective measure of how far the disruption carries over from the edges of the moved blocks. Deletions present no such ambiguity: each word is either present in the next version, or is deleted. Furthermore, all major content reorganizations involve text deletions, as merging new and old content requires rewording and restructuring the old content.

Thus, a basic evaluation criterion consists in measuring the precision and recall of low-trust with respect to text deletions. For each trust value $t \in [0, T_{\max}]$, we consider the fact of a word w having trust $t_w \leq t$ as a "warning bell", and we ask what is the recall, and the precision, of this warning bell with respect to the event of the word being deleted in the next revision. The recall $recl(t)$ measures the fraction of

[3] The computation uses author reputation, but author reputation can also be computed on the basis of the past history of the text; see, e.g., [1].

deleted text that had trust smaller than or equal to t immediately prior to deletion; the precision $prec(t)$ measures the fraction of text with trust smaller than or equal to t which is deleted in the next revision. More formally, let:

- $W^{\leq}_{i,p}(t)$ be the number of words in version i of article p that have trust no larger than t;
- $D^{\leq}_{i,p}(t)$ be the number of words in version i of article p that have trust no larger than t and which are deleted in the revision from version i to $i+1$;
- $D_{i,p} = D^{\leq}_{i,p}(T_{\max})$ be the number of words in version i of article p which are deleted in the revision from version i to $i+1$.

Then, we have:

$$recl(t) = \sum_{i,p} D^{\leq}_{i,p}(t) \;/\; \sum_{i,p} D_{i,p} \qquad (3)$$

$$prec(t) = \sum_{i,p} D^{\leq}_{i,p}(t) \;/\; \sum_{i,p} W^{\leq}_{i,p}(t)\;, \qquad (4)$$

where the summation is taken for all versions of all articles that are used to evaluate the quality of the trust labeling.

While recall and precision of low-trust are good indicators, they suffer from the fact that text can be deleted by vandals, only to be re-added in the next revision. This source of error can be significant: while people intent on improving an article often delete small amounts of text at a time, vandals often delete the entire text of an article. To obtain better measures, we would like to give more weight to deletions that happen due to well-thought-out editorial concerns, rather than vandalism. To this end, we employ the notion of *edit longevity* developed in [1]. The edit longevity $\alpha_{i,p} \in [-1, 1]$ is a measure of how long-lived is the change $e_i : v_{i-1} \rightsquigarrow v_i$ for article p. In particular, if $\alpha_{i,p}$ is -1, then the change e_i is reverted immediately, and if e_i is a deletion, then practically this should not be considered as a valid deletion. On the other hand, if $\alpha_{i,p}$ is close to 1, the change will live through many subsequent revisions, and if e_i is a deletion, then it should be considered as a valid deletion [1]. We use the *edit quality* $q_{i,p} = (\alpha_{i,p} + 1)/2$ to weigh the data points in (5)–(6), thus giving weight close to 1 to deletions that happen due to authoritative revisions, and no weight to deletions performed by vandals (which have longevity -1). We thus define the *quality-weighed* recall and precision of low-trust with respect to deletions as follows:

$$w\_recl(t) = \frac{\sum_{i,p} q_{i,p}\, D^{\leq}_{i,p}(t)}{\sum_{i,p} q_{i,p}\, D_{i,p}} \qquad (5)$$

$$w\_prec(t) = \frac{\sum_{i,p} q_{i,p}\, D^{\leq}_{i,p}(t)}{\sum_{i,p} q_{i,p}\, W^{\leq}_{i,p}(t)}\;. \qquad (6)$$

3.2 Trust distribution of general vs. deleted text

Another criterion for judging the quality of a trust labeling consists in considering the trust value distribution of all text, and of deleted text. Recall that, in our system, we display the text of revisions with a background color that reflects text trust, and which ranges from white for fully trusted text, to orange for text with trust 0. Site visitors are going to use the orange background as an indication that the information may be unreliable. If too much text on an article has orange background, the alert loses effectiveness, as visitors habituate to the constant flagging of text. Thus, we prefer trust labeling in which text, on average, is as trusted as possible. On the other hand, we clearly want text to be flagged as low-trust when it is about to be deleted.

To make these notions precise, we define the following quantities. Given a function $f : [0, T_{\max}] \mapsto \mathbb{R}$ with $\int_0^{T_{\max}} f(t)\, dt < \infty$, and $\rho \in [0, 1]$, we define the *ρ-median* of f the quantity a satisfying

$$\int_0^a f(t)\, dt = \rho \int_0^{T_{\max}} f(t)\, dt\;.$$

We also denote with $W^{=}_{i,p}(t)$ the amount of text having trust t in version i of article p, and we denote with $D^{=}_{i,p}(t)$ the amount of text in version i of article p having trust t which will be deleted in version $i+1$. We define the following notations:

$$tot\_txt(t) = \sum_{i,p} W^{=}_{i,p}(t)$$

$$del\_txt(t) = \sum_{i,p} D^{=}_{i,p}(t)$$

$$w\_del\_txt(t) = \sum_{i,p} q_{i,p} D^{=}_{i,p}(t)\;.$$

We assess the quality of a trust labeling via the following quantities, for $\rho \in [0, 1]$:

- The *ρ-white point* is the ρ-median of $tot\_txt(t)$.
- The *weighed orange average* is the average value of t for $w\_del\_txt(t)$.

We will use $W_{0.9}$ and Org_{avg} to denote the 0.9-white point and weighed orange average, respectively. Again, the weighing used in the definition of orange average is used to give more weight to deletions that occur in the course of higher-quality revisions.

3.3 Trust as predictor of text life-span

Our final criterion for judging the quality of a trust labeling consists in quantifying the predictive value of word trust with

respect to the subsequent life-span of the word. To measure this predictive value, we sample word occurrences from all versions uniformly at random (applying the algorithm to all words would be computationally very expensive), and we observe for how many consecutive article versions the words are present after their sampled occurrence.[4]

The simplest approach consists in studying the correlation between the trust t of the word at the moment it is sampled, with the life-span l of the word, measured as the number of consecutive subsequent versions in which the word is present. However, such a measurement would be biased by the *horizon effect* induced by the fact that we have only a finite sequence $v_0, v_1, \ldots, v_n$ of versions to analyze. Words sampled in a version v_i, and that are still present in the last version v_n, have a life-span of $n - i + 1$, even though they may live much longer once the wiki evolves and versions beyond v_n are introduced. This horizon effect causes us to under-estimate the true life-span of high-longevity words sampled close to the last existing version of an article.

To obtain a measurement that is unaffected by this horizon effect, we model the life-span of a word as a memoryless decay process, in which the word has a constant probability (dependent on the word, but not on its past life-span) of being deleted at every revision. Thus, we assume that the probability that a word that at v_i has trust t is still alive at v_k, for $k \geq i$, is $e^{-(k-i)/\lambda(t)}$, where $\lambda(t)$ is the half-life of the word under infinite-horizon. Our task is to measure the half-life $\lambda(t)$ as a function of t. Note that this definition of half-life eliminates the horizon effect due to the finite number of versions.

For every word sampled at v_i, and last present in v_k, with $i \leq k \leq n$, we output a triple (t, l, h) consisting of the trust t of the word in v_i, the life-span $l = k - i + 1$, and the observation horizon $h = n - i + 1$. To estimate $\lambda(t)$, we use the following observation: if $l < h$, then the word would have lived for l even under infinite horizon; if $l = h$, then the word has an average life-span of $l + \lambda(t)$ under infinite horizon, since the distribution is memoryless. Let A be the set of triples sampled for a trust level t. Let:

- m be the number of samples in A with $l < h$;
- $M = \sum\{l \mid l < h \wedge (t, l, h) \in A\}$;
- k be the number of samples in A with $l = h$;
- $K = \sum\{l \mid l = h \wedge (t, l, h) \in A\}$.

We can estimate $\lambda(t)$ via

$$\lambda(t) = \frac{M + K + k \cdot \lambda(t)}{m + k}.$$

This leads to the estimate $\lambda(t) = (M + K)/m$. A trust labeling will have high predictive value for life-span if larger values of t correspond to larger values of $\lambda(t)$.

3.4 Predicting stability vs. providing visual feedback

The evaluation criteria introduced above measure the quality of a trust labeling via its ability to predict text instability. While predicting instability is surely an important requirement of a trust system, a trust system in practical use also has another goal: providing visitors with visual feedback on the past edits to articles. While the goals of predicting stability, and providing visual feedback, are often compatible, there are instances when they are not. As an example, consider the case of an author removing a sentence from a paragraph. Our trust labeling will label low-trust both the end of the sentence preceding the removal, and the beginning of the sentence immediately following the removal. This low-trust labeling, and the resulting orange coloring, is used to make readers aware that some edit has occurred — that text was removed. The low-trust labeling is thus given for feedback purposes, and this use may be at odds with the goal of maximizing its power to predict instability. Indeed, sentences that precede and follow the removal are unlikely to be themselves deleted, so that from a prediction point of view, the labeling is inappropriate.

In our system, we strive for a mix of these prediction and feedback goals. However, our evaluation reflects only the predictive aspect of trust: we do not know how to algorithmically evaluate its feedback value.

4 Implementation

We have implemented a modular tool for computing author reputation and text trust on the Wikipedia. The tool, evolved from the one for author reputation described in [1], takes as input an XML dump of a Wikipedia, made available from the Wikimedia Foundation. An XML dump consists of all the articles of the Wikipedia, in sequential order. For each article, the dump lists all versions, along with meta-information for each version (such as author and date). The text of the versions is encoded in *Mediawiki markup language*, a markup language with tags and constructs denoting titles, tables, list items, and more. The tool traverses a Wikipedia XML dump, feeding the article versions to one of several analysis modules. We developed analysis modules for computing author reputation and text trust, as well as for analyzing a number of statistical properties of the Wikipedia; other modules can be easily added via a simple API. The tool is written in Ocaml [15]; we chose this language for its combination of speed and excellent memory management. On

[4] As we have seen in Section 2.3.1, a word in a version can correspond to multiple occurrences in the next version, when text is duplicated. When tracking a word to measure its life-span, whenever the word is duplicated, we track all occurrences separately.

an Intel Core 2 Duo 2 GHz CPU, the code is capable of processing and coloring versions of mature Wikipedia articles[5] at over 15 versions/second, or roughly 1.5 millions versions per day, an edit rate much higher than the one of the online Wikipedia [28]. We plan to make the tool available in open-source format. A demo of the trust coloring is available at `http://trust.cse.ucsc.edu/`; the code will be made available at the same URL.

4.1 Computing author reputation histories

Computing the trust coloring is a multi-step process, which begins with the computation of author reputation. We rely on *content-driven* reputation system for Wikipedia authors proposed in [1]. In this system, authors of contributions which prove long-lasting gain in reputation, while authors whose contributions are reverted lose reputation. Specifically, whenever an author A edits an article that had been previously edited by another author B, a change in reputation is generated for B: the reputation of B increases if A preserves B's contribution, and decreases if A undoes B's contribution. The reputation system is thus *chronological:* the reputation is computed from the chronological sequence of increments received by authors. To compute this content-driven reputation, we run the tool over the XML dump, extracting edit difference information among versions of each article. This information is then sorted according to the global chronological order of versions of all articles. Finally, it is fed to the process of reputation computation described in [1]. The outcome is a *reputation history* file containing, for each author, the chronological history of the author reputation in the Wikipedia, from the first edit performed by the author to the last. The reputation system is such that $T_{\max} = 9$.

4.2 Computing text trust and origin

We display the trust of each word by coloring the background of the word: white for fully trusted words, and increasingly intense gradations of orange for progressively less trusted text. While we compute trust using floating-point numbers, for display purposes we round it up into 10 levels, from 0 (lowest) to 9 (highest).

To color the article text, our tool first reads the reputation history file produced in the first pass, and then it takes a second pass over the XML dump file, using a trust-coloring module that implements the algorithm of Section 2. During the second pass, the tool computes the trust of all words of all versions of all Wikipedia articles. The tool also computes, for each word, the version where the word was first introduced, thus allowing site visitors to explore the provenance of the information they are presented. The information about text trust and origin is encoded by adding two tags to the Mediawiki markup language:

- the tag $\{\{\texttt{tt:}x\}\}$ indicates that the subsequent text has trust $x \in \{0, 1, \ldots, 9\}$;
- the tag $\{\{\texttt{to:}i\}\}$ indicates that the subsequent text was first inserted in version i (Mediawiki assigns to each version a global integer identifier).

To save storage, these tags are not added for each word, but only when the information changes from one word to the next. The extended markup language is then output by the tool as a "colorized" XML dump that is identical to the input dump, except for the presence of the trust and origin tags. This colorized dump can be loaded in any Mediawiki installation, using the standard procedures for reproducing the Wikipedia (Mediawiki [20] is the software package responsible for implementing the wiki behind Wikipedia). A plugin we developed for Mediawiki interprets the additional markups and enables site visitors to see the computed trust information as the color background of text.

4.3 Displaying trust and origin information

Adding the trust and origin tags to the Mediawiki markup language without breaking the visual formatting of Wikipedia articles is a minor challenge in itself. The markup language is position sensitive: for instance, the title (==) and bullet (*) markups only work when they occur precisely at the beginning of a line, and tables have complex rules that determine where extra markup can be added without breaking the table formatting. Furthermore, there is no complete documentation of the language, especially as authors often abuse it: "everything that renders fine, is fine". Inserting the markup properly involved developing a parser for the markup language occurring in practice in Wikipedia articles (including errors and abuses), with the purpose of identifying the places where the tags could be safely inserted.

The additional tags are then interpreted by Mediawiki extensions we developed, following the Mediawiki extension framework. The extensions intercept the Mediawiki translation from markup language to HTML and translate the coloring and origin tags into appropriate HTML span elements. The trust span elements are mapped to a text background color via Cascading Style Sheets. For text origin, we define a on-click action in JavaScript, so that when a user clicks on a word, the user is sent to the article version where the word was first inserted. The two types of information, trust and origin, augment each other, and together provide Wikipedia visitors with effective tools to judge the accuracy of article contents. The trust coloring focuses visitors' attention to the portions of an article which are less reliable, either because they are very recent, or because they were introduced by low-reputation authors and have been insufficiently revised. The

[5] Measured on a randomly-selected subset of articles with at least 200 versions each.

origin labeling can then be used to explore how the unstable information was added to the article.

5 Results

Our first step in the performance evaluation of the trust labeling consisted in choosing values for the constants appearing in the trust labeling algorithm. Choosing values for the constants involves balancing the recall and precision of the trust labeling: the recall is a measure of the trust labeling's ability to flag unreliable content, and the precision is a measure of how likely it is that something flagged will turn out to be unreliable. Thus, obvious candidates for optimization were the weighed recall $w\_recl(t)$ and the precision $w\_prec(t)$, for $t \in [0, T_{\max}]$ defined in Section 3. However, this approach was difficult to follow in practice. First, the particular value of $t \in [0, T_{\max}]$ that should be picked for optimization was not clear: which value of trust is low enough, or which shade of orange is dark enough, to constitute a warning? Second, it was not clear to us what would constitute acceptable values of recall and precision.

We found it much easier to reason about how "white" a mature article should be on average, and about how "orange" the deleted text should be: thus, we performed the optimization using the white point and orange average, as defined in Section 3. We let $W'_{0.9} = W_{0.9}/T_{\max} \in [0, 1]$ be the normalized 90%-white-point, and we let $Org'_{avg} = (T_{\max} - Org_{avg})/T_{\max} \in [0, 1]$ be the normalized weighed orange average, where $T_{\max} = 9$ for our system. We wanted to find parameter values that would make the article, overall, as white as possible (maximize $W'_{0.9}$), while ensuring the deleted text was as orange as possible (maximize Org'_{avg}). To this end, we used linear search on the space of the parameters to optimize the value of the *weighed harmonic mean* of $W'_{0.9}$ and Org'_{avg}, i.e., we optimize $F(W'_{0.9}, Org'_{avg}) = 2 \cdot W'_{0.9} \cdot Org'_{avg}/(W'_{0.9} + Org'_{avg})$, for a set of 100 articles used for training. We use the weighed harmonic function since it weighs both of its arguments evenly. This led to the following values for the parameters:

$$c_r = 0.2 \quad c_l = 0.4 \quad c_e = 2 \quad c_p = 0.2 \quad c_k = (\log 2)/T_{\max} .$$

With these parameters, we proceeded to evaluate the performance of the trust coloring on a set of 1,000 articles selected uniformly at random among the articles with at least 200 revisions; the articles comprised 544,250 versions all together, for a total of 13.7 GB of text. We focused on articles with long revision histories for two reasons. From a technical point of view, the long revision history enables us to better estimate the predictive power of trust with respect to text stability. From a user-interface point of view, our trust is especially useful for mature articles: it is relatively easy for visitors to conclude that incomplete articles, with short revision history, cannot (yet) be trusted.

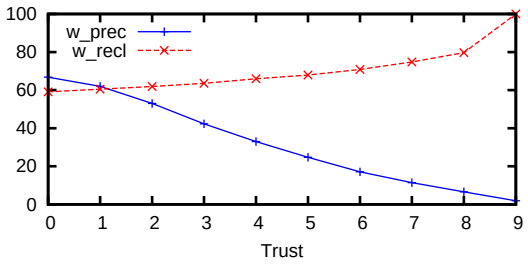

Figure 3: Low-trust as a predictor of deletions: quality weighed precision and recall.

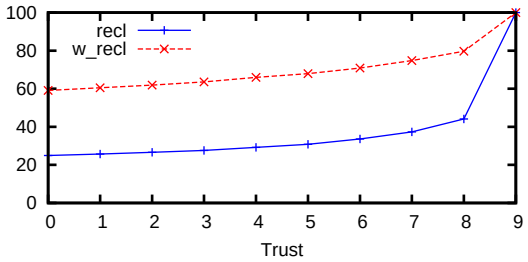

Figure 4: Comparison of recall and weighed recall.

Figure 3 gives the quality-weighed precision and recall of low trust with respect to text deletions. We see that the recall is always at 60% or above; in practice, a mid-range orange background, which is sure to attract a reader's attention, is able to warn the reader to 2/3 of the text that will be deleted in the next revision. We believe that this is a good performance figure, given that text can be deleted for many reasons other than poor quality, such as rewording: thus, some deletions are never likely to be anticipated by low trust. The precision figures give the probability that text marked as low-trust will be deleted in the very next revision; low precision figures would be a sign of excessive warnings to visitors. We see that text with trust 0 has a 2/3 probability of being deleted in the next revision, and text with mid-level trust has a 1/3 probability of deletion; we consider this to be an acceptable level, especially since not all text that will be deleted is going to be deleted in the very next revision. In Figure 4 we compare weighed and unweighed recalls: as we see, if we also include deletions due to vandalism ($recl$), our recall drops, reflecting the fact that such vandalistic deletions are hard to predict.

The color profiles of general and deleted text are compared in Figure 5. We can see that deleted text, on average, is much lower in trust: indeed, the average trust of deleted text was 2.96, while 90% of text had a trust above 7.60 (out of a maximum of $T_{\max} = 9$).

Figure 6 depicts the correlation between the trust of a word occurrence, and the subsequent life-span of the word. The data is obtained by random sampling of word occurrences, and tracing the future of the word from the sampling

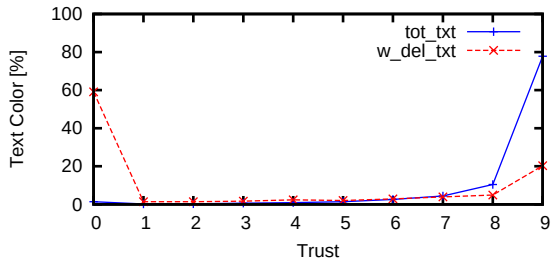

Figure 5: Color of general and deleted text.

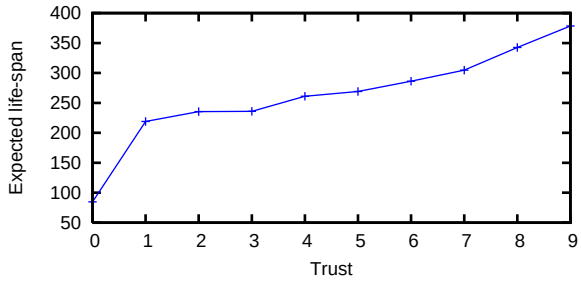

Figure 6: Expected future life-span $\lambda(t)$ of words as a function of their trust label t.

point. We note that the trust is the trust of the word *occurrence:* over the subsequent life-span, the word trust may well vary (and typically increases, as the article is revised). We see that there is a clear correlation: higher trust corresponds to a longer expected life-span. We also see that there is a sharp increase in expected life-span as we go from words labeled with trust 0 to words labeled with trust 1. This can be explained by the high "early mortality" of words with trust 0: over 60% of them, as indicated by the recall graph in Figure 3, do not make it to the next version.

We also evaluated the magnitude performance improvement due to the use of the author attention modeling presented in Section 2.3.2. To our surprise, we discovered that the author attention modeling does not appreciably improve the results, in spite of introducing additional degrees of freedom in the trust algorithms. We believe this is due to the fact that authors usually edit the sections of an article that have received the most recent edits. Thus, outside of the paragraph being edited, there is not much text which can benefit from a trust increase, and distinguishing between edited and non-edited paragraphs has little effect.

We also experimented with using the trust algorithm without a reputation system, instead assigning everybody, from anonymous visitors to well-established editors, the maximum value of trust. Fresh text, as well as block-move edges, received initially trust 0,[6] and the trust of text would then increase according to the algorithms of Section 2 (no change

[6]Had we used a trust value greater than 0 as initial value, no text would ever get trust 0.

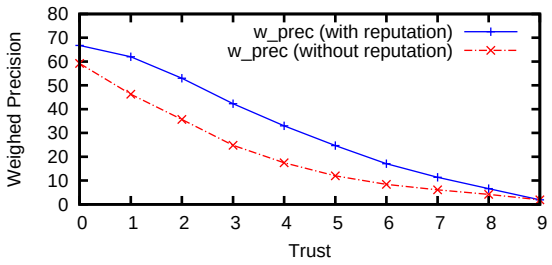

Figure 7: The weighed precision with and without reputation systems.

was performed to the trust algorithms). We chose coefficients for the trust computation that would yield an weighed orange average similar to the one obtained using a reputation system. The trust labeling computed without the aid of a reputation system performed worse than the one that made use of the reputation system of [1], but the performance gap was narrower than we expected. The performance gap was most noticeable with respect to the precision, as illustrated in Figure 7: for trust 4, for instance, the precision was nearly double (33%) with the reputation system than without (17.5%). The gap for recall was narrower: for trust 4, the quality-weighed recall was 66% using a reputation system, and 72.5% without. Furthermore, while deleted text had similar colors, the average text was noticeably more orange in the tests not using the reputation system: the 90% white point went from 7.6 using reputation, to 5.43 when reputation was not used.

This performance difference can be explained as follows. One of the benefits of using a reputation system is that text which is inserted or moved by high-reputation authors receives a non-zero initial value of trust (in our system, $0.616 \cdot 9 \approx 5.5$). This reflects the fact that high-reputation authors are statistically more likely to perform good contributions [1]. If we do without a reputation system, all newly inserted or rearranged text instead has trust 0 initially. This makes the text lower-trust overall (thus the lower 90% white point), and this decreases precision, since among the low-trust text is plenty of text that is due to authors who are statistically likely to perform good contributions.

The use of a reputation system has another benefit: it limits how much authors can increase the trust of article text. Without a reputation system, any author could repeatedly edit the text of an article, causing most of the text (except for the edited portion) to raise in trust. When a reputation system is used, on the other hand, authors can cause the non-edited text on articles to raise in trust only up to their reputation value. This difference is not apparent in our data, gathered on a dump, but would come into play in an on-line implementation of our trust system.

6 Future Work

The implementation described in this paper is based on batch-processing Wikipedia dumps. We are currently working on an on-line version of the system, which would provide Wikipedia visitors with real-time information about text trust. While this requires a reorganization of the data flow in the system, all main algorithms will be unchanged: in fact, our trust algorithm is purely chronological, so that the trust of the latest version of an article involves only the consideration of the previous version of the article, along with information about the author. We do not expect the computational power to process edits to be a challenge, as remarked in Section 4.

One of the challenges in developing an on-line trust system consists in making the system hard to attack. We are particularly concerned about attacks that aim at inserting misleading information and causing the information to be labeled with high trust. We believe that several features of the proposed trust system contribute to making it resistant to this and other types of attack. In the proposed system, only high-reputation authors can cause text to gain the maximum trust value. Furthermore, each high reputation author can affect an article only in limited fashion. First, consecutive edits by the same author are collapsed into one when computing trust (see Section 2.1). This prevents high-reputation authors from raising the article trust in accelerated fashion via multiple edits: edits by other authors are also needed. We remark that it would be difficult for authors to have multiple separate identities all with high reputation, due to the slow way in which authors gain reputation in our content-driven reputation system [1]. Second, even high reputation authors who edit a page leave some track in the form of medium-trust text, as discussed previously. We are currently analyzing various types of attacks, and we believe a reasonably robust system is attainable in practice.

References

[1] 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.

[2] J. Blad. Article quality and user creditability, 2006. From http://meta.wikimedia.org/wiki/User:Agtfjott.

[3] C. Castelfranchi and eds. Y. Tan. *Trust and Deception in Virtual Societies*. Kluwer Academic Publishers, 2001.

[4] T. Cross. Puppy smoothies: Improving the reliability of open, collaborative wikis. *First Monday*, 11(9), September 2006.

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

[6] W. Emigh and S. Herring. Collaborative authoring on the Web. In *Proc. of HSCC*, 2005.

[7] J. Giles. Internet encyclopaedias go head to head. *Nature*, pages 900–901, December 2005.

[8] J.A. Golbeck. *Computing and Applying Trust in Web-Based Social Networks*. PhD thesis, University of Maryland, 2005.

[9] T. Grandison and M. Sloman. A survey of trust in internet application. *IEEE Comm. Surveys Tutorials*, 3(4), 2000.

[10] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proc. of the 13th Intl. Conf. on World Wide Web*, pages 403–412. ACM Press, 2004.

[11] M. Hickman and G. Roberts. Wikipedia — separating fact from fiction. *The New Zealand Herald*, Feb. 13 2006.

[12] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proc. of the 12th Intl. Conf. on World Wide Web*, pages 640–651. ACM Press, 2003.

[13] R. King. Contributor ranking system, 2007. White paper available from http://trust.cse.ucsc.edu/Related_Work.

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

[15] Xavier Leroy. Objective caml. http://caml.inria.fr/ocaml/index.en.html.

[16] A. Lih. Wikipedia as participatory journalism. In *Proc. 5th International Symposium on Online Journalism*, 2004.

[17] V.B. Livshits and T. Zimmerman. Dynamine: Finding common error patterns by mining software revision histories. In *ESEC/FSE*, pages 296–305, 2005.

[18] P. Massa. Wikipedia trust network, 2007. http://www.gnuband.org/2007/06/26/wikipedia_trust_network/.

[19] D.L. McGuinness, H. Zeng, P.P. da Silva, L. Ding, D. Narayanan, and M. Bhaowal. Investigation into trust for collaborative information repositories: A Wikipedia case study. In *Proceedings of the Workshop on Models of Trust for the Web*, 2006.

[20] http://www.mediawiki.org/.

[21] B. Mingus, T. Pincock, and L. Rassbach. Using natural language processing to determine the quality of wikipedia articles. In *Wikimania, Taipei, Taiwan*, 2007.

[22] F. Ortega and J.M. Gonzales-Barahona. Quantitative analysis of the wikipedia community of users. In *Proc. of Wikisym*. ACM Press, 2007.

[23] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kiwabara. Reputation systems. *Comm. ACM*, 43(12):45–48, 2000.

[24] R. Stross. Anonymous source is not the same as open source. *The New York Times*, Mar. 12 2006.

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

[26] 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.

[27] J. Voss. Measuring wikipedia. In *Proc. of ISSI*, 2005.

[28] http://stats.wikimedia.org/EN/TablesDatabaseEdits.htm.

[29] http://en.wikipedia.org/wiki/Wikipedia:Size_comparisons.

[30] D. Wilkinson and B. Huberman. Cooperation and quality in Wikipedia. In *Proc. of WikiSym*. ACM Press, 2007.

[31] H. Zeng, M. Alhossaini, R. Fikes, and D.L. McGuinness. Mining revision history to assess trustworthiness of article fragments. In *Proc. of the 2nd Intl. Conf. on Collaborative Computing: Networking, Applications, and Worksharing (COLLABORATECOM)*, 2006.

[32] 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.