Rocovo : Robust Communal Publication Scheme

Thomas Largillier
Normandie Univ
UNICAEN, GREYC
Caen, France

Guillaume Peyronnet
Nalrem Medias, France

Sylvain Peyronnet
Normandie Univ
UNICAEN, GREYC
Caen, France

ABSTRACT

We study the problem of designing a robust validation method for publishing content on social websites. In such websites, users can propose their content to the community. Since the data is user generated, a huge volume of content is published every day on a successful platform. An important problem arising in such websites is that users have to filter all the published content to find something interesting. This filtering can be assisted by the website, thanks to a recommendation mechanism. An even more important problem is that the context is adversarial: some malicious users try to trick the system in order to either publish their own content or make sure that specific content is not published. Therefore, users might not find the content they are interested in either because it is diluted in an ocean of spam or because it was maliciously blocked from publication. In this paper, we present Rocovo (RObust COmmunal VOting), a two phases mechanism that allows for the automatic management of the publication process in a robust and communal manner.

1. INTRODUCTION

The social Web has emerged more than a decade ago, as the term Web 2.0 goes back to 1999. Nowadays, nobody can deny that the way people interact with each others on the Web has drastically changed with the emergence of social sites such as social networks, blogging platforms, customer reviews websites, etc. All these websites provide an aggregation of user-generated content. Most of the time, they offer filtered information thanks to social recommendation methods. Most of these methods are based on collaborative filtering: relevant documents are suggested to users based on the taste and behavior of other users.

There are many examples of such websites. *Digg*[1] (launched in 2004) was one of the first to appear. Digg is a social news website: users share content they found on the Web through its interface, then they can digg (vote up) or bury (signal as inappropriate) any news published on the website. Digging a news is then considered as a recommendation, and news with a sufficient number of recommendations are displayed on Digg's front page. Also if a news receive a high enough number of bury it will be manually examined and eventually removed from the website. We mention Digg because it provoked the emergence of numerous Digg clones (generally denoted as *Digg-like*). This huge success can be explained by the amount of traffic such a website aggregates and redistributes. Since most websites follow an economic model based on advertisement, obtaining visitors is the best way to improve the income. It is then tempting for a user to use malicious techniques to obtain a good visibility for his websites on Digg or any Digg-like. A typical malicious technique is explained in [10] where it can be read that, in 2006, the top 30 users of Digg were already responsible for the majority of the front page of the site. Since 2006, malicious users became more and more efficient on social sites (see for instance the paper of Heymann *et al.* [6]). At the same time, more and more social sites redirect more and more traffic to other websites, being thus targets of choice for malicious webmasters. For instance, Facebook represents 5.39% of all Web referrals in August 2013[2].

To the best of our knowledge, no social website implements a robust non-intrusive mechanism that allows for the community to publish user-generated content while preventing manipulations by malicious users. Spotrank [9] targets the same problem. It is built over *ad-hoc* statistical filters, a collusion detection mechanism and the notion of *pertinence* of voters and news. The drawback of this approach is that antisocial behavior are taken into account *a posteriori.*

This paper is using a completely different approach in order to obtain a more robust voting scheme. More precisely, we use a collaborative approach based on the concept of "games with a purpose" of Luis Von Ahn [16, 17], where players are representative committees of the community. This way, we can ensure that the community will always be capable of avoiding bad content to be published.

The main contributions of this paper are:

- **The design of the Rocovo algorithm.**
  Rocovo is a two steps algorithm. The first step consists in creating two randomly sampled committees representative of the whole community. The second step decides, depending on the votes of the committees' members, if a content is to be published or not. Using random sampling for creating the committees

[1] http://digg.com

[2] http://www.netmarketshare.com/2011/12/01/Google-as-a-Referral-Source-Continues-Downward-Trend

SAC'14 March 24-28, 2014, Gyeongju, Korea.
Copyright 2014 ACM 978-1-4503-2469-4/14/03 ...$15.00.

and a 2-player game between committees for validating the content, we ensure that published content is both representative of the taste of the community and not pushed by malicious users.

- **A strong experimental analysis.**
  We present results obtained through simulation to give evidence of the efficiency of our approach. We show that the behavior of a social website that uses Rocovo is the one expected, even in presence of various types of malicious users.
  We do not use any existing dataset to validate our approach because our technique filter the content *a priori*, while all existing dataset that we know contain already published items.

The structure of the paper is as follows. In section 2 we give some insight about the related work. Then, in section 3, we present our modeling and describe in detail the Rocovo algorithm. In section 4, we discuss our hypothesis and the feasability of the method in a real life framework. Last, we give in section 5 an experimental analysis of our method.

2. RELATED WORK

In this paper we introduce a robust voting mechanism whose main application is to filter out spam on social websites. There are a few research papers that are dedicated to the analysis of such websites. The team around Kristina Lerman has done extensive work in this field [10, 12, 11]. These papers analyse the behavior of users and content in social sites such as Digg. A user modeling approach is used to predict which news can obtain good ranking according to the first votes [12]. Even if this work is concerned with social websites, their goal is different from ours. Their research provides a detailed analysis of those websites while we, on the other hand, aim at designing a robust voting scheme in an adversarial environment (we have a normative approach).

In [6], the authors present a survey on spam countermeasures for social websites. They present a classification that separates such countermeasures into three categories. In particular, they mention spam prevention through interface-based methods. The idea is to harden the publication of spam using specifically designed mechanisms. Our approach falls into the scope of these methods since we reduce the chances that bad content is published using a specific voting mechanism.

In [1], the authors describe a machine learning based ranking framework for social media that is robust to some common forms of vote-based spam attacks. Other work focuses on manipulation-resistant systems [14], and use a notion close to the one of pertinence, introduced in [9].

Our present work is using a completely different approach in order to obtain a robust voting scheme. More precisely, we use a collaborative approach based on the concept of "games with a purpose" of Luis Von Ahn [16, 17], where players are representative committees of the community. This way, we can ensure that the community will always be capable of avoiding bad content to be published.

This approach has been already used for different purposes. For instance, a two-player game to fight webspam is presented in [5]. Also, spam assessment tasks (*i.e.* the human task of labelling webpages in order to find spam) are often done using some kind of game-inspired approach [2].

The problem of providing users of a community with a good selection of news can also be seen as a recommendation problem [15]. One of the earliest attempts to design recommendation systems was given in [4] for collaborative filtering techniques. Cosley *et al.* [3] study the relation between recommendation systems and users, while Lam and Riedl [8], and O'Mahony *et al.* [13] address the problem of malicious users and the robustness of systems. The first step of Rocovo borrows ideas from recommendation systems: we ask a committee to evaluate the content suggested for publication by some users. The committee decision to allow the publication of some content can be seen as a recommendation of this content to the whole community. However, Rocovo is not a recommender since it only assess the fact that the published content is acceptable by the community and do not attempt at ranking items.

3. ROCOVO ALGORITHM

Here, we first present the framework on which Rocovo is built, together with its principle. Then, we describe each part of the mechanism.

Framework and principle.

We assume that our publishing system Rocovo is used by a community of n users $\mathcal{U} = \{u_1, \cdots, u_n\}$. Any member of the community can ask for the publication of a document. The set of all documents is denoted by $\mathcal{D}$. Rocovo is based on rounds or periods. A period is a time interval during which a bucket of documents is submitted and processed by the algorithm. The set of all periods is $\mathcal{P} = \{p_1, \cdots, p_{|\mathcal{P}|}\}$. $\mathcal{D}$ can be partitionned into subsets $\mathcal{D} = \cup_{p_i \in \mathcal{P}} D_{p_i}$. D_{p_i} is thus the set of documents that are submitted for publication for the period p_i. For the sake of clarity, we present in the paper the behavior of the algorithm for one period. This way we can ease the notation by removing references to p_i. Moreover, we will consider that only one document is submitted during this unique period. this document will be denoted by d.

Figure 1 depicts the internal process followed by Rocovo to decide on the publication of one suggested document. Rocovo works in several steps in order to avoid the publication of unwanted (by the whole community) or spammy documents with high probability.

The first idea (random committee selection) is borrowed from recommendation systems. The document to be published should be acceptable for the whole community but it is not tractable to ask each community member to validate each document. Thus, we use the opinion of a few users to decide on the acceptance of a document. In recommendation systems, these few users constitute a committee. The key point in constituting committtees is to make sure that they are representative of the community. We address the specifics of Rocovo's committee members selection later in the paper.

To decide if a document is suitable for publication, each committee member is voting for acceptance or rejection of the document. The vote is done with a $\frac{2}{3}$-majority rule. An important point is that not all voters have the same "weight". If someone is constantly agreeing with the rest of the community, she is rewarded and her weight increases (up to a certain bound). If someone is disagreeing with the rest of the community, her weight decreases. This means that when we mention a $\frac{2}{3}$-majority rule one should keep in

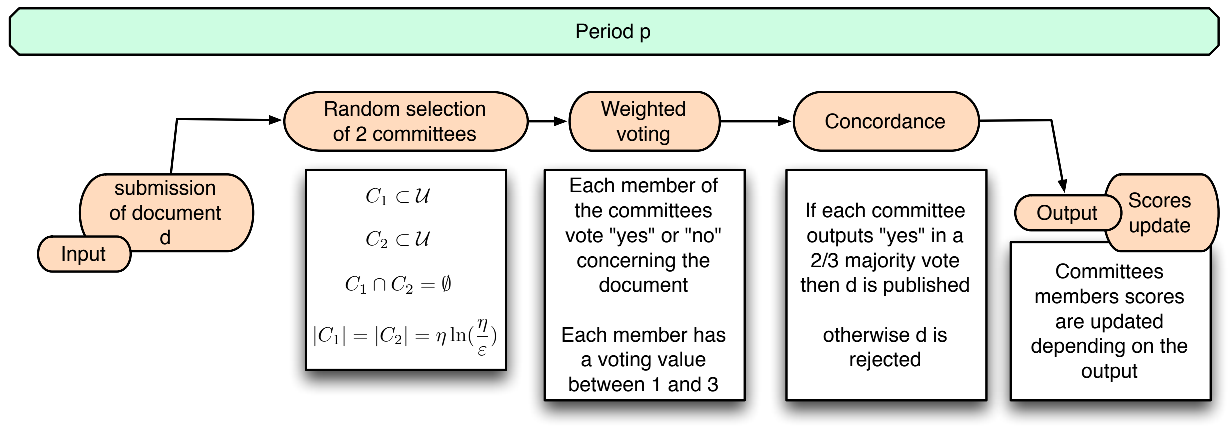

Period p
Input
submission of document d
Random selection of 2 committees
$C_1 \subset \mathcal{U}$
$C_2 \subset \mathcal{U}$
$C_1 \cap C_2 = \emptyset$
$|C_1| = |C_2| = \eta \ln(\frac{\eta}{\varepsilon})$
Weighted voting
Each member of the committees vote "yes" or "no" concerning the document
Each member has a voting value between 1 and 3
Concordance
If each committee outputs "yes" in a 2/3 majority vote then d is published
otherwise d is rejected
Output
Scores update
Committees members scores are updated depending on the output

Figure 1: Rocovo's algorithm

mind that each committee member has a different weight and thus the vote of one member can be worth the added votes of several others. Moreover, the threshold of the vote is, of course, a parameter of the method. Here we use the specific value $\frac{2}{3}$ to ease the readability of the paper.

Having only one committee is not enough if one wants to obtain a robust mechanism. This is why we use two committees. At the end of the period, a document is accepted if the two committees, selected for this item, output the same positive recommendation. In all other cases (disagreement, or agreement on a negative recommendation), the document is rejected. This use of an agreement is similar to what is done in "games with a purpose" [16, 17] (see [5] for an application to webspam detection). However, our method is not strictly a "game with a purpose" since it is not a game.

At the end of the period, the document is either accepted or rejected. All committees member weights are updated, as well as their scores.

Committees.

The first step of Rocovo is the setting of two committees C_1, $C_2 \subseteq \mathcal{U}$. A committee is built upon a small number of community's members. Its goal is to made a proposal on the acceptance or rejection of a submitted document. If the two committees agree on the acceptance, then the document is accepted, otherwise it is rejected.

In order not to overwhelm users with the publication process , we need the committees to be as small as possible, but we also have to make sure that the chosen committees can decide for the whole community. This means that we want *representative* committees. In order to give a clear definition of this notion, we assume that the community is partitioned into a constant number of equivalence classes E_i. Two users are said to be equivalent if they agree on the decision to be made on a document. Note that the assumption that users can be categorized in classes that strongly differ one from each other is very common in recommendation systems. In our framework, the categorization of users mainly depends on their ability to distinguish between honest and spam content independently from their personal interest in the content. In the rest of the paper, we denote by η the number of equivalence classes. For the sake of simplicity, we also assume that all classes have the same size (this assumption is discussed in section 4)

Definition 1. *A committee C is representative if and only if it contains with high probability at least one member of each equivalence class. That is, C is representative if and only if $\forall E_i$ $\exists u \in C$ s.t. $u \in E_i$, with high probability (1 - ε), where $\varepsilon \in\ ]0,1[$.*

The less costly way to set up a committee is to pick uniformly at random members of the community. The probability that a randomly picked user belongs to an equivalence class E_i is $1/\eta$ for all i since all the classes have the same size. The question that arises then is to decide how many members must be picked at random in order to obtain a representative committee. The following proposition addresses this problem.

Proposition 1. *If we pick uniformly at random $\eta \ln(\frac{\eta}{\varepsilon})$ members of the community, then we obtain a representative committee with probability at least $(1-\varepsilon)$*

Proof. Let NOK be the event that the committee is not representative. After picking uniformly at random m users, the union-bound principle gives:

$$P(NOK) \leq \sum_{i=1}^{\eta}(1-\frac{1}{\eta})^m.$$

Since $\sum_{i=1}^{\eta}(1-\frac{1}{\eta})^m = \eta(1-\frac{1}{\eta})^m \leq \eta e^{\frac{-m}{\eta}}$ (by convexity reason), we have that:

$$P(NOK) \leq \eta e^{\frac{-m}{\eta}}.$$

We want $P(NOK) \leq \varepsilon$, so we obtain that:

$$m = \eta \ln(\frac{\eta}{\varepsilon})$$

is sufficient for our purpose. $\square$

In practice, we will take $\alpha \times \eta \ln(\frac{\eta}{\varepsilon})$ members at random, α being a well chosen constant to avoid a deadlock due to a small proportion of users being late to respond (or not responding) about the status of the submitted document. Typically it will be between 1.5 and 2. Since we need two committees, we also have to take the same number of users amongst those that are not in the first committee. Typical practical values for η and ε are 3 and 0.05, meaning committees of size 13. We set η to 3 following the idea that we have three classes: spammers, honest users capable of detecting spam content, and honest users not capable of detecting it.

Votes.

Once a committee is set, the document is submitted to its members. Each committee member vote for acceptance or rejection of the document. The committe is proposing the document for publication if the majority of the expressed votes accept the document, otherwise it is rejected.

Note that in our election process, members are not equal. More precisely, each member got a specific weight during the vote. This weight is a natural number less than 3 in our practical experiments. The bound on the weight is a parameter whose value depends on the strength of the targeted filtering of bad content. We will see later how we can give a specific weight to a specific user.

The vote is thus made in the following way: we sum up the weights of the supporters of the document, we sum up the weights of the detractors of the document. The document is accepted if more than $\frac{2}{3}$ of the weighted votes are in favor of it. Again, the value $\frac{2}{3}$ is the one we use in practice, but can be tuned to obtain a different behavior for the whole process.

With these specific values for the weights and for the majority threshold, it is easy to see that it is very difficult for malicious users to manipulate the outcome of the vote. Let us consider the extreme case where all malicious users vote in the same way and have the maximum weight (two very unlikely events), assuming that we have 1000 users in the community, 20% of them being malicious persons acting together and that a committee have size 13, they can manipulate the vote if and only if they represent more than 46 % of the committee. In this case, the probability of a successful manipulation of one committee is less than 3%. In a more reasonable case where a malicious user acts like an honest user long enough so he have the same weight, the probability of a successful manipulation is less than 0.00015%[3]. All these probabilities are computed as cumulative probabilities of a hypergeometric distribution. This means that this probability of a successful manipulation is given by:

$$\sum_{i=\beta|C|}^{|C|} \frac{\binom{n/s}{i}\binom{n-n/s}{|C|-i}}{\binom{n}{|C|}}$$

Here, β is the proportion of malicious users needed in the committee to ensure the success of a manipulation. n is the size of the community, $|C|$ the size of the committee and $1/s$ is the probability of finding a malicious user when picking at random a community member. Giving a closed form of the cumulative probability above is a difficult problem, and even approximations are difficult to obtain [18].

Validating the publication.

A document is accepted for publication if and only if the two committees agree on the acceptance, otherwise it is rejected. This step can be seen as a game with a purpose where players are the committees. The goal here is to amplify the probability that a bad behavior is ineffective. In the previous examples, the probabilty of success for a manipulation becomes less than 0.1% for the extreme case, and 10^{-7}% in the reasonable case.

Updating the values of the users.

Once a document has been either accepted or rejected, weights and scores are updated.

The weight of a user is his value during the vote. At first, the weight of any user is 1. The value evolves vote after vote, but is bounded (by 3 in our setting). If the committees reached an agreement, then all users that voted for the actual output have their weight increased by 1. Those that are already at 3 stay at 3. Other users (that for instance voted for acceptance while the document was rejected) have their weight set to 1. If the committees did not reach an agreement, then the weight of users that voted for acceptance is set to 1. The weight of other users is unchanged.

The score of a user is a way to reward him for his good behavior. Each time a user is in a situation where his weight should have been increased, his score increases. In practice, score is a sort of money that users can exchange for various things (following the gamification idea [19]). In our experiments, users earn 10 credits each time their score increases. Every 100 credits they can buy a publication token.

The number of tokens is the last value attached to a user. A token is a right to submit a document for publication. At first, a user has 1 token. When a user submit a document, he loses a token. If the document is accepted, he retrieves the token, otherwise the token is lost permanently. The only other way to gain tokens is to achieve good behavior during the voting phase.

Rocovo is strategy-proof: there is no interest for an honest user to vote for a document he does not want for the community. More precisely, an honest user can gain from a bad behavior only if he is in a situation where he knows that there is enough malicious users in both committees so that the outcome of the votes will not be the one that fits the need of the community. Since committees members are randomly selected, making the probability of having a sufficiently large set of malicious users in both committees very low, the score expectation is higher for someone that plays honestly.

4. DISCUSSION

In this section, we discuss the hypotheses made in this paper. We also elaborate on some comments we had from early readers of the paper.

Parameters' values.

The method uses many parameters: the outcome of the vote depends on a 2/3 threshold, the weight of a specific user ranges in $\{1, 2, 3\}$, etc. All the values that we use in this paper are likely to be modified in a real-life framework. We chose to use actual values instead of symbols for the sake of simplicity and to ease the reading of the paper.

Security issues.

Our method assume that it is possible to generate random samples. More specifically, it means that we assume that we have a strong enough pseudorandom generator. The exis-

[3]See `http://www.berbiqui.org/thomas/sac2014/`

tence of pseudorandom generators sufficient for monte carlo methods is known since 1990 [7]. We can use such a generator for our method.

We don't assume that identities are stable. More precisely, white-washing is possible: a detected spammer can come back with a new identity. This is of course an overhead for the method, but it won't preclude the behavior of the publication system. Moreover, in a practical setting, we will identify people using a third party service (such as openID, facebook or twitter identification), and will thus be as strong as these services when it comes to multiple avatars.

We implicitly limit Sybil attack with our hypothesis that the users classes are of equivalent size. Indeed, a Sybil attack is a large group of users sharing the same behavior, this is thus a unique class of users. Either this class is the only one, and it's no longer a corruption of the community since it is the entire community, or there are at least two classes, and with a 2/3 majority vote, the impact of the Sybil attack is limited.

Users' behaviour and classes in a real-life context.

The assumption that is the most surprising is probably the fact that we assume that all classes have the same size. First, have the same size must be understood as have similar sizes. Second, the main interest of this hypothesis is that it makes the paper easier to read (less technicalities). Moreover, we can always comply with this assumption by dividing big classes into smaller classes. We then obtain more classes, but all of similar sizes.

The main concerns we have for practical settings are:

- The system depends on committees putting in time to review posts. It means that the community must be very motivated in order to obtain a constant stream of published information. Because of this, the method is best fitted for expert communities, or for enterprise social networks. For instance, an intranet where all engineers of a very large international company share their best practices.
- When there are very few users, the review burden is high. This is the case during the startup phase of a social network. Our first real-life experiments lead us to think that during the startup phase, it may be necessary to lower the number of users in the committees.
- The punishment for attempting to post spam is draconian. To overcome the fact that people may leave the network because of that, we advise to use additional mechanisms to give credits to users after some time.

5. EXPERIMENTS

In this section we present the simulations we conducted to assess the good behavior of the proposed scheme from an empirical point of view. We run 7 scenarios. In the first we consider only honest users, in scenarios 2 to 6 the scheme is tested against a specific adversary while in the last scenario, it is tested against all adversaries altogether. In those experiments, users start with 3 tokens and win 10 credits every time they do a "good" vote. When a user gather 100 credits, he buys an additional token.

Each experiment is composed of 50 rounds. The user set consists of 1000 users, 125 of which are malicious in adversarial scenarios. In each round all users get a chance to publish a new item. The probability that any user will post an item is determined by a power law. For any honest user, the probability that the published item is "good" is denoted by p. In the following experiments $p = 90\%$. Once we have the list of all items for a round, we select two committees for each item and proceeds to the votes. In these experiments any honest user has a 70% probability of voting. If they decide to vote for an item, they have a 85% probability to vote according to this item's category, *i.e.* voting yes for a "good" item and refusing a "spam" item. In all the following experiments, $\eta = 5$ and $\varepsilon = .05$. The presented results are the average values over 22 repetitions of each experiment.

The results are summed in Tabs. 1, 2 and Fig. 2. Tab. 1 shows the average publication ratio for posts over 10 rounds regarding their nature. Every scenario is depicted through a line in this table. Tab. 2 represents the minimum, average and maximum number of tokens accumulated by honest users for the first scenario and each kind of adversaries in the following ones after 10, 20, 30, 40 and 50 rounds. Fig. 2 depicts the "weight" of the vote for honest users and adversaries after each round $i \geq 1$

First scenario.

The behavior of the scheme without any adversary will be used as a reference to determine the impact of malicious people on the system. The first line of Tab. 1 shows the acceptance rate of posts regarding their nature. We can see that when there are only honest users in the system, no spam post is ever published while the publication rate of good posts is around 90%. This is the reference behavior for our system and we expect to see as few deviation from it as possible in the presence of adversaries.

The first line of Tab. 2 shows the number of tokens accumulated by the users over time. We can see that only a few users have less than one token at every moment since the average value grows together with the maximum. The presence of users with less than one token could be explained by the fact that some users may post few and bad quality articles even if their intent is good. On the other hand, the maximum value represents users with a high voting rate that propose legitimate content.

The evolution of the vote's value of users is represented in Fig. 2a. As it can be expected most users have a vote value of 3 which is the maximum. Others users are those that did not vote or chose the wrong option when voting for a proposed post. The proportion of those users is in accordance with the error probability we set for the votes.

Second scenario.

In this scenario, we consider adversaries that are trying to push their content or content proposed by their friends to the website. They will propose "bad" content and will always vote yes for bad content. Regarding "good" content, those adversaries behave like honest users. This is a stronger version of the real life adversary since here all malicious users are part of the same cabal while in reality they will be divided into smaller groups with a smaller probability of success. It can be seen on the second line of Tab. 1 that despite their number, those malicious users do not succeed in publishing spamming content. Their behavior also does not affect the publishing ratio of good content as expected.

The second line of Tab. 2 shows the number of publication tokens own by the malicious users over time. As one can see,

	1 - 10		11 - 20		21 - 30		31 - 40		41 - 50	
	G	S	G	S	G	S	G	S	G	S
Sc 1	89.50	0.00	90.21	0.00	90.22	0.00	90.11	0.00	89.55	0.00
Sc 2	90.14	0.00	90.06	0.00	90.66	0.00	89.81	0.00	89.86	0.00
Sc 3	82.82	0.00	87.67	0.00	88.75	0.00	88.73	0.00	87.44	0.00
Sc 4	88.74	0.00	89.48	0.00	89.15	0.00	89.44	0.00	89.00	0.00
Sc 5	90.41	0.00	90.17	0.00	89.46	0.00	89.29	0.00	89.75	0.00
Sc 6	90.36	0.00	90.38	0.00	89.90	0.00	90.49	0.00	90.56	0.00
Sc 7	90.08	0.00	90.55	0.00	90.13	0.00	89.88	0.00	90.02	0.00

Table 1: Percentage of posts acceptation regarding their type

	10			20			30			40			50		
	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min
Gen	18	8.30	3	33	13.96	3	45	19.72	2	60	25.58	2	73	31.26	0
Adv1	17	3.36	0	33	5.23	0	44	7.15	0	56	9.01	0	67	10.94	0
Adv2	8	3.63	3	12	4.59	3	16	5.57	3	20	6.59	3	24	7.58	3
Adv3	9	4.29	3	15	5.97	3	19	7.70	3	24	9.41	3	29	11.12	3
Adv4	13	7.87	3	23	13.08	5	32	18.38	7	41	23.73	8	50	29.03	10
Adv5	35	13.75	6	66	25.04	11	92	36.29	16	121	47.75	21	150	59.13	28
Adv6	23	6.49	0	43	10.67	0	61	14.89	0	79	19.09	0	97	23.24	0

Table 2: Number of tokens per user through time

most of them possess very few tokens. The maximum value is similar to the honest users value since they behave as good voters, they are able to buy a publication token every 100 credits earned. We can see that even if malicious users can regain publication tokens through a genuine voting strategy on non spam content, they cannot publish any spam content as shown on Tab. 1.

Almost all malicious users in this scenario possess the smallest vote value as seen on Fig. 2b. This is logical since they try to push spamming content. The few users that sometimes achieve a higher vote value are those who were not recently on the committee involving spamming content. Since they behave as regular users to avoid detection they vote accordingly to the content value and can see their voting value rise. We can also see that even with users having the maximum voting value it is not possible for those spammers to push content.

Third scenario.

The adversaries implemented in this scenario do not attempt to promote their content but rather to crash the website by always voting no for "good" content. We can see on the third line of Tab. 1 that their attack is ineffective on the system. They hardly decrease the probability of accepting good content. The third line of Tab. 2 shows the evolution of their number of credits. Some users succeed in getting some credits, but they are far from the number of credits obtained by honest users (see the first line of the table). A more interesting value for those users is the value of their vote. One can see on Fig. 2c that almost all those users have the smallest vote value. This is explained by the fact that mostly good content is proposed during those simulations and since they always vote to refuse they very often contradict the community opinion and so their vote has only a reduced impact.

Fourth scenario.

These adversaries share the objective of the ones in the third scenario, they aim at crashing the site rather than promoting their own content. They are the opposite of genuine users since they vote no for every "good" content and yes for "bad" content. The fourth line of Tab. 1 shows that their action is inefficient regarding the promotion of "bad" content or the demotion of "good" content. Since they always vote against the nature of the content, it is really hard for them to increase their vote value. Most of them have the lowest possible vote value as seen on Fig. 2d. The fourth line of Tab. 2 shows the evolution of their number of tokens. Since they always vote against the nature of an item, when they correctly evaluate it, it is hard for them to earn credits and purchase more publication tokens. This explains why they achieve numbers close to the ones of the third scenario.

Fifth scenario.

The adversaries of this scenario use a random voting strategy tossing a coin each time they need to vote. This behavior can be observed for users that do not take the time to look at the item before voting. As seen on the fifth line of Tab. 1, the action of these users do not affect the acceptance ratio for both "good" and "bad" content. It is only logical since even if there is a large enough number of them inside the same committee, they do not cooperate. Regarding the evolution of their number of tokens shown on the fifth line of Tab. 2, we can see that few users can purchase many tokens, some users cannot purchase any token and that most users can purchase some tokens. This is explained by their voting strategy. Since it is random, a few users will always agree with the committees decision, a few will always disagree and most will alternate correct and incorrect decisions. This effect can also be seen on the vote value of those users on Fig. 2e. We can see that around half those users have the lowest voting value and the remaining half is split almost equally between voting value 2 and 3. This is because there are mostly "good" content proposed to the system and therefore tossing a coin is not the optimal strategy to vote blindly.

Sixth scenario.

In this scenario, the adversaries always vote yes, regardless of the content quality. Again those adversaries are trying to promote "bad" content and to crash the website. This behavior might also be adopted by users relying on the rest of the community to do the filtering and see the committee's role as too much work. This blind voting strategy, even when used by an association of those users in a committee do not increase the acceptance ratio of "bad" content as shown on the sixth line of Tab. 1. Since mostly "good" content is proposed in this scenario, these users tend to agree with the committee most of time, therefore most of them have a high vote value and can purchase many publication tokens as seen on the sixth line of Tab. 2 and Fig. 2f.

Seventh scenario.

In this scenario all kinds of adversaries are represented. Adv1 represents half the adversaries while the remaining classes are equally distributed. All those adversaries have divergent objectives and thus it is interesting how their combined action might affect the system. As seen on the seventh line of Tab. 1, their combined behavior do not lower the acceptance ratio of "good" content nor increase the "bad" content one. Also, most of them only possess a low number of publication tokens and the lowest vote value as shown on the seventh line of Tab. 2 and Fig. 2g. The system is then robust even when facing a great number of diverse adversaries that either try to promote their own content or alter the experience of the genuine users by promoting "bad" content or shunning "good" content.

6. CONCLUSION

In this paper, we presented a robust validation method for publishing content on social websites. This mechanism, called Rocovo, uses representative committees of the whole community and a two steps voting algorithm in order to obtain both representativity of the tastes of the community, and robustness to the behavior of malicious users. A set of experiments show that the method is indeed robust, even if a large number of various malicious users are in the community.

We have implemented Rocovo in an actual social website[4] to assess its real-life feasibility and efficiency.

[4]`http://www.krinein.fr`.

7. REFERENCES

[1] J. Bian, Y. Liu, E. Agichtein, and H. Zha. A few bad votes too many?: towards robust ranking in social media. In *Proc. of the 4th international workshop on Adversarial information retrieval on the web*, pages 53–60. ACM, 2008.

[2] C. Castillo and B. Davison. *Adversarial web search*, volume 4. Now Publishers Inc, 2011.

[3] D. Cosley, S. K. Lam, I. Albert, J. A. Konstan, and J. Riedl. Is seeing believing?: how recommender system interfaces affect users' opinions. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 585–592. ACM, 2003.

[4] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, pages 61 – 70, 1992.

[5] M. Goodstein and V. Vassilevska. A two player game to combat web spam. Technical report, Technical Report, Carnegie Mellon University, 2007.

[6] P. Heymann, G. Koutrika, and H. Garcia-Molina. Fighting spam on social web sites: A survey of approaches and future challenges. *IEEE Internet Computing*, 11(6):36–45, 2007.

[7] F. James. A review of pseudorandom number generators. *Computer Physics Communications*, 60(3):329–344, 1990.

[8] S. K. Lam and J. Riedl. Shilling recommender systems for fun and profit. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 393–402, New York, NY, USA, 2004. ACM.

[9] T. Largillier, G. Peyronnet, and S. Peyronnet. Spotrank: a robust voting system for social news websites. In *Proceedings of the 4th workshop on Information credibility*, WICOW '10, pages 59–66, New York, NY, USA, 2010. ACM.

[10] K. Lerman. User participation in social media: Digg study. In *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, pages 255–258, Washington, DC, USA, 2007. IEEE Computer Society.

[11] K. Lerman. Dynamics of a collaborative rating system. pages 77–96, 2009.

[12] K. Lerman and A. Galstyan. Analysis of social voting patterns on digg. In *WOSP '08: Proceedings of the first workshop on Online social networks*, pages 7–12, New York, NY, USA, 2008. ACM.

[13] M. O'Mahony, N. Hurley, N. Kushmerick, and G. Silvestre. Collaborative recommendation: A robustness analysis. *ACM Trans. Internet Technol.*, 4(4):344–377, 2004.

[14] P. Resnick and R. Sami. The influence limiter: provably manipulation-resistant recommender systems. In *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, pages 25–32, New York, NY, USA, 2007. ACM.

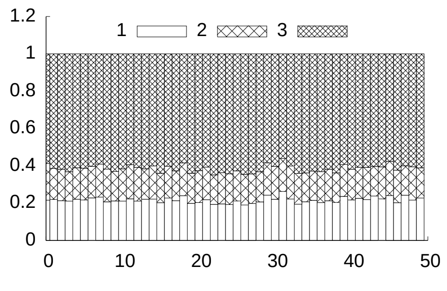
[15] P. Resnick and H. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[16] L. Von Ahn. Games with a purpose. *Computer*, 39(6):92–94, 2006.

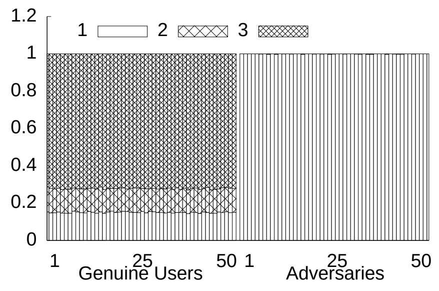
[17] L. Von Ahn and L. Dabbish. Designing games with a purpose. *Communications of the ACM*, 51(8):58–67, 2008.

[18] T. Wu. An accurate computation of the hypergeometric distribution function. *ACM Transactions on Mathematical Software (TOMS)*, 19(1):33–43, 1993.

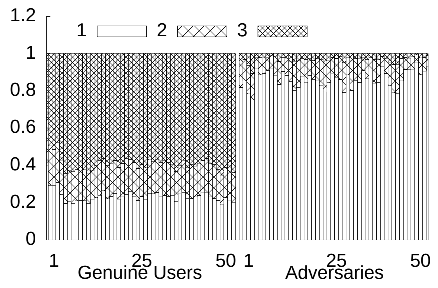
[19] G. Zichermann and C. Cunningham. *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps.* O'Reilly Media, 2011.



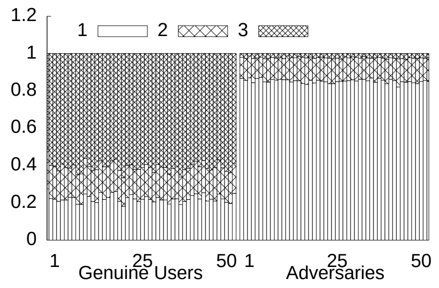
(a) Regular users



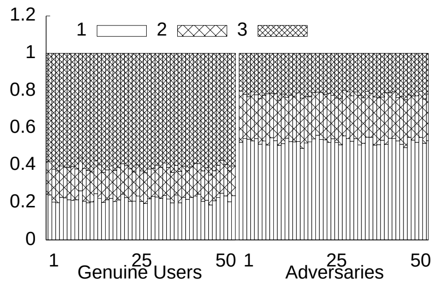
(b) Adversary 1



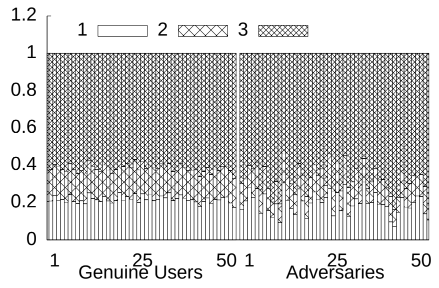
(c) Adversary 2



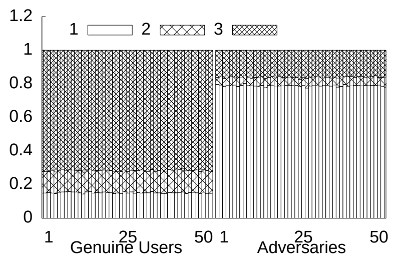
(d) Adversary 3



(e) Adversary 4



(f) Adversary 5



(g) All adversaries

Figure 2: Vote's value of users after each round in the experiments