Technical Overview of Relevant Protocols

Slava Balasanov

February 7, 2018

Abstract

The purpose of this document is to provide a general technical overview of the reputation and incentive protocols on the Relevant platform. For more context and an introduction, please consult the Relevant Mission Statement blog post.

1 Introduction

The Relevant news-sharing app uses reputation and incentive protocols to promote the curation of quality information. The *reputation protocol* is used to identify and rank experts for any number of topics. This enables us to rank content based on the level of expertise of those who share and endorse it. The *incentive protocol* encourages all users to increase the overall quality of information on the platform. Expert users are incentivised to share and rate information they deem relevant, while non-expert users are encouraged to identify *predictably* popular (or poor) content that does not require extra attention. Relevant protocols are designed to accommodate general use cases and can be used in a variety of other apps and services that require effective methods for sourcing quality content.

2 Reputation Protocol

Reputation is the information we use to assess someone's trustworthiness [14]. Platforms like Airbnb, Uber and eBay rely on reputation systems to mitigate counterparty risk and enable strangers to transact with one another [8]. At Relevant, we are building a reputation system that will help filter and rank content, enabling users to discover quality articles recommended by the community.

2.1 Reputation Context

Reputation is always context-specific [1]. In a marketplace we want to know whether the seller is honest and reliable — so a marketplace reputation system would ideally have a metric for honesty and reliability. A general reputation protocol thus needs to take into account the context of its reputation. On the Relevant platform, we want to be able to rate user's expertise in relation to a specific topic. If I'm interested in the topic 'blockchain', I want my feed to contain articles suggested by users that are experts on blockchain technology.

In order to bootstrap our reputation system, we define a set of topics, each of which will serve as a context for an expertise score plus an additional "global" context we can use to track global reputation:

```
reputation-contexts: [
  "global",        // global category (can be used to nominate admins)
  "politics",
  "technology",
  ...
]
```

Work in Progress: We are researching ways to compute reputations for dynamic ontologies in order to accommodate a diverse and evolving set of topics and subtopics. At the moment this is possible with non-sybil-resistant reputation systems but can become computationally expensive with the manipulation-resistant algorithm we propose.

2.2 Computing Reputation Scores

2.2.1 Direct Trust and Localized Reputation

We can use an algorithm similar to PageRank [11] to compute user reputation scores. We construct our reputation graph by looking at how users rate user-generated content they see on the platform (see Fig. 1). For example, Alice sees Bob's commentary on an article about blockchain tech and upvotes it. We now know that Alice trusts Bob's 'blockchain' expertise.

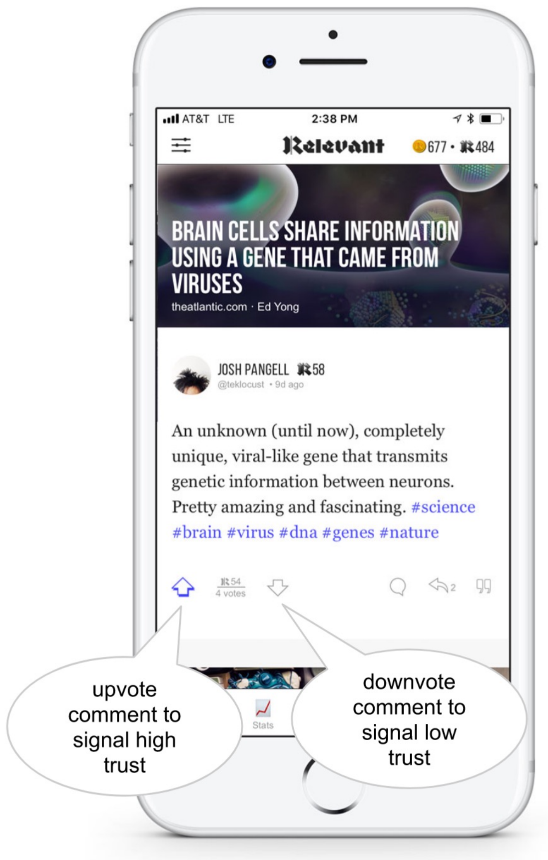


Figure 1: UX for upvoting user commentary.

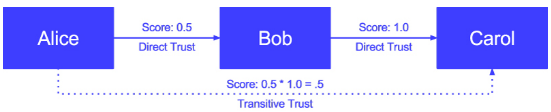
2.2.2 Transitive Trust and Distributed Reputation

After computing local reputation scores for all users, we combine them to determine transitive reputation scores. Because Alice trusts Bob, she 'transitively' trusts those who Bob trusts [6]. Assume we know that Bob trusts Carol. If we let t_{XY} denote how much X trusts Y, then

$$t_{AliceCarol} = t_{AliceBob} * t_{BobCarol}$$

Say Alice gave Bob a trust score of .5 and Bob gave Carol a trust score of 1.0. We can calculate how much Alice trusts Carol:

$$t_{AliceCarol} = .5 * 1 = .5$$



We then look at all the people Carol trusts, and so on. As we compute transitive trust scores for further and further nodes, the scores converge[1] to a global trust score for each user independent of who we start with.

In this way we can compute the global reputation score for every user, on every topic. One of the most popular existing algorithms to do this is EigenTrust, which was originally designed to determine the reputation of nodes in a peer-to-peer network [7]. EigenTrust++ is an improved version that is more appropriate for a social network by virtue of resiliency to a variety of manipulation attacks [2].

2.3 Manipulation - The Russian Bots Problem

An important quality of a reputation system is resistance to manipulation. This is especially true for a decentralized system where it's not easy to manually exclude malicious actors.

[1] This is a property of the EigenTrust algorithm

We have seen how easy it is to manipulate and game various systems based on quantifiable metrics. TripAdvisor is full of fake reviews, there are millions of Russian bots on Twitter, it's trivial to buy Instagram followers and so on. The most popular attack method, and one that plagues most online reputation systems is the Sybil attack — where an adversary creates a large quantity of fake accounts in order to submit fraudulent ratings and manipulate the reputation system. The first step to preventing this kind of attack is through identity verification. One way to accomplish this is to require users to register and authenticate their accounts with a decentralized identity service like Civic or Uport, or by asking them to provide verification via a phone number or social media accounts.

However we don't need every user to be a verified and trusted individual. Eigentrust++ uses an initial set of trusted users as a starting point for propagating reputation scores. Multiple connecting paths to the initial set are required to build up a positive reputation. This makes it difficult for Sybil nodes to build up reputations by repeatedly upvoting one another and also makes it trivial for users to flag suspicious activity. One limitation is that we can only let users earn reputation from activities that are difficult to accomplish programmatically — like text-based commentary and opinions. If one could earn reputation from simply sharing an article, an adversary could potentially deploy content-sharing bots that earn reputation and then manipulate the reputation system.

Even if bots are able to spoof text comments and fool some, more discerning users have a recourse to downvote or flag the bots. And if a bot is good enough to fool most users and provide value to the platform by automating curation, then the bot deserves to have a reputation and subsequent curation rewards.

2.4 Decentralized Reputation Protocol Overview

2.4.1 Setup - Trusted Seed

We continue bootstrapping the reputation system by defining a set of trusted experts for each topic. These are users we can trust to be non-malicious. For example, users whose identity can be verified via identity providers or social media. This initial set does not necessarily have to consist of "top" experts. Candidates should have some expertise in the given subject, should be deemed to be a reliable person and should be willing to be a participant in the network. In addition we create one "global" reputation set that we can use for selecting users to fulfill general governance tasks — a role similar to the admins in Wikipedia's governance system [9].

```
reputation-contexts: [ "global", "politics", "technology", ...]
experts: {
  global: ["Alice", "Bob", ...],          // global admins
  politics: ["Carol", "Dan", ...],
  technology: ["Erin", "Frank", ...],
}
```

2.4.2 Reputation Events

Users earn reputation when other users rate their commentary on an article, blogpost or other content. We define these as reputation events. For example on the Relevant app, any user can 'upvote' or 'downvote' another user's comment about a particular topic. **Users do not earn reputation from 'sharing' an article, only from commenting on an article.**

Ex. — Alice upvotes Bob's comment on an article about politics and culture:

```
"reputation-event": {
  "payload": {
    "app": "0x123..."                          // App public key
    "source": "0x123...",                        // Alice's public key
    "target": "0x123...",                        // Bob's public key
    "tags": ["global", "politics", "culture"],  // reputation context
```

```
    "content": "/ipfs/asd..."                    // content address
    "rating": 1,                                 // normalized rating 0 or 1
    "sample-size": 1,
    "generated": 1492205001                      // unix timestamp
  },
  header: {
    "source-signature": "QuAsa..."               // Alice's signature
    "app-signature": "WuaXs..."                  // Application signature
  }
}
```

These reputation events are recorded on the blockchain and the reputation algorithm is able to access them in order to compute the global trust scores for all nodes.

Reputation flow example:

1. Bob shares an article about ‘blockchain’ along with some insightful commentary
2. Alice, who has a high reputation in the ‘blockchain’ category upvotes Bob’s comment
3. Bob’s ‘blockchain’ reputation increases
4. Bob is a big fan of Carol’s posts about ‘blockchain’ and has often upvoted her annotations
5. Carol’s ‘blockchain’ reputation increases as a result of transitive trust

2.5 Reputation Governance

In the future we may need to add or remove users from the initial trusted set of experts. For example, we may want to remove users that have not been active, or add new members. In this case, the list of users can be managed by a modified version of Mike Goldin’s Token Curated Registry (TCR), "a decentrally-curated list with intrinsic economic incentives for token holders to curate the list’s contents judiciously" [5].

Users manage the list by proposing and voting on the addition or removal of the list’s members. The person that proposes a new member of the list must also provide a small deposit to prevent spam. Relevant will use a variation of TCR to balance both on-chain and off-chain governance.

Some notable properties of Relevant’s TCR implementation include:

- A user’s vote weight will be determined by both reputation and token holdings. Ex: vote weight = normalized reputation * tokens. Users with a reputation of 0 will not be able to ‘buy in’, but will have to spend time building up their reputation. Malicious users can also be ‘blacklisted’, forcing them to start building up their reputation from scratch.
- Some of the high-level decisions will be made by admins only (users in the global expert set). This process will resemble Wikipedia’s governance structure — off-chain discussions and voting will be open to all users, however the admins will have the final say to account for low voter participation and vote buying.
- Users will always have the ability to fork the Reputation system as a check on admins, as well as propose the removal or addition of admin members.

  Reputation-based governance is more conducive to hard forks then stake-based governance. In the latter, forking governance becomes impossible without modifying token balances since whales hold the majority of vote weight. In our protocol user can be blacklisted and reputation annulled.
- This governance methodology can be extended to cover other blockchain decisions including economic and incentive parameters.

2.6 Why Stake-based Curation is Often a Bad Idea

Steemit, Curation Markets [12], and Token Curated Registries are a few of the projects in the crypto community that are centered around stake-based curation of information. Users buy 'curation tokens' and use them to vote on content. The more 'curation tokens' you hold, the more your vote counts. One aspect of this dynamic that is intuitively unsettling is that wealth supersedes merit; i.e., a not-so-great curator with a million tokens can have a tremendous impact on ranking, but a great curator with one token will have virtually none.

We can see this problem compounded when we add incentives to the system. If the only metric for rating information is stake votes, and we allocate rewards for the highest rated content, we are invariably rewarding participants that already hold a large supply of tokens. This makes it hard for a great curator with few tokens to earn a greater share of influence, and further entrenches the wealthy as the most influential curators — a vicious cycle also known as the Matthew Effect [10].

We believe stake-based voting only works when the context for 'great curation' is aligned with wealth, for example: investment advice, or bidding on an advertising space. In cases where quality is not aligned with wealth, stake voting will not be useful, and is likely to be harmful. We have seen a clear illustration of this on Steemit. It is a great place to discover technical information about the Steem platform, token and community — but not much else (especially not material critical of Steem, which is censored by the token cartel).

At Relevant, we are interested in curating qualities that are not aligned with wealth, but with human values. This is why we have developed a non-transferable reputation metric that reflects the human values that the community holds in high esteem. Content ranking is determined via reputation voting only — curators with the highest reputation make the biggest impact.

We believe that by detaching the act of curation from wealth we will be able to create a meritocratic reward system that incentivises quality curation. However, users still have an opportunity to stake on their curation activities in order to have a chance to earn more tokens. Our *incentive protocol*, described in the next section, allocates rewards based on reputation and content quality first, and only then takes the number of staked tokens into account.

3 Incentive Protocol

At Relevant, we are developing an economic system that rewards users for curating quality content. Our goals are twofold: we want to create a stable currency that will reflect the value of the network and appreciate as the network grows. But we also want to create a meritocratic distribution and circulation system that rewards those who provide the most value to the ecosystem. These two goals are not necessarily contradictory, but can easily become so if we don't balance the incentives to hold and spend coins.

Bitcoin has demonstrated that a system optimized for store of value does not make for a healthy redistribution and transaction ecosystem. At the same time, bootstrapping a utility token (currency that derives its value purely from the flow of transactions), is hard. If there is no incentive to HODL a currency it can quickly drop in value as users convert to store-of-value coins or fiat. Lastly, even if we have a stable liquid currency, we still need to incentivise the discovery and curation of *quality* information. If the system can be gamed at the expense of quality, our protocol won't live up to its promises.

The Relevant incentive protocol is modeled after Steem, with one important improvement — the integration of a reputation system to facilitate a meritocratic redistribution of wealth. We believe the protocol will be able to facilitate both a store of value and a robust redistribution dynamic. Initially, the Relevant Token will have a 10% inflation rate that will be gradually adjusted to meet the demands of the platform. In order to earn newly minted tokens, users will have to vest their Relevant Tokens. The more tokens one vests, the more one stands to earn. The vesting period — how long it takes to 'withdraw' all your tokens — is 3 months. Once vested, tokens become *Curation Tokens* and enable participation in various aspects of the curation protocol.

Definition: *Curation Tokens* are vested Relevant Protocol Tokens and are required for participation on the Relevant platform.

Work in Progress: With the addition of multiple reputation systems there will be multiple Curation Tokens, one for each reputation system. This will allow users to stake tokens on the reputation system of their choice.

3.1 Desired Properties of Incentive Protocol

- Incentivise curation of quality information, where quality is defined by a given community
- Reward the curation of unpredictably relevant content (as opposed to a trending NYT article)
- Limit ability to game the system / make gaming the system useful

3.2 Prediction Market

A recent Reddit study [4] found that "most users do not read the article that they vote on" and that out of all votes, "73% of posts were rated (i.e. upvoted or downvoted) without first viewing the content."

We can imagine this problem being exacerbated with the potential to earn rewards for rating content. It requires very little work to roughly predict which content will be popular — based on the headline, the poster's follower count, their reputation and the source of said content. So it is trivial simply upvote all the posts that are likely to be popular and suck up a large portion of the curation rewards without providing any real value to the network.

Instead of fighting this problem we can channel it into a productive activity that will help us curate information. One proposed method for doing this is implementing a prediction market. The market will determine what content is likely to become the most popular and curation reward payouts will be adjusted to account for predictability. **This will enable the protocol to reward users for discovering and curating content that is unexpectedly relevant.**

Because there is no reputation requirement, bots can compete to participate in the prediction market. This kind of bot activity will not be detrimental, on the contrary, it will provide benefit to the platform by providing predictability metrics for various content.

3.2.1 How it works

Users decide what score an article should get on a range of -1 and 1 and are able to stake some Curation Tokens on their prediction (see Fig. 2). (We can even experiment with designing a UI where users are not aware whether they are rating content or participating in a prediction market.)

Prediction market properties:

- Open for 30m after creation of a post (as ratings start to come in, predictions become less useful)
- Participation doesn't require and doesn't take into account reputation — great for bots or users with no reputation
- UX detects whether the user has read a post and decides on prediction / rating mode
- Prediction is tied to UX activity such as "save to read later"
- Users can choose how many tokens they want to stake on their guess
- First users to make a prediction have the best chance to win rewards
- The more tokens one stakes the better the odds of winning rewards

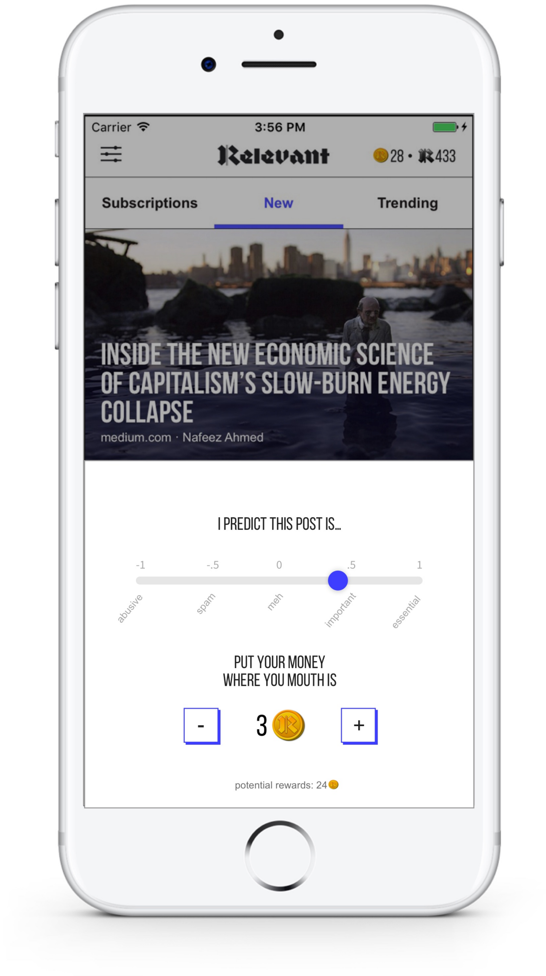


Figure 2: Prediction UX

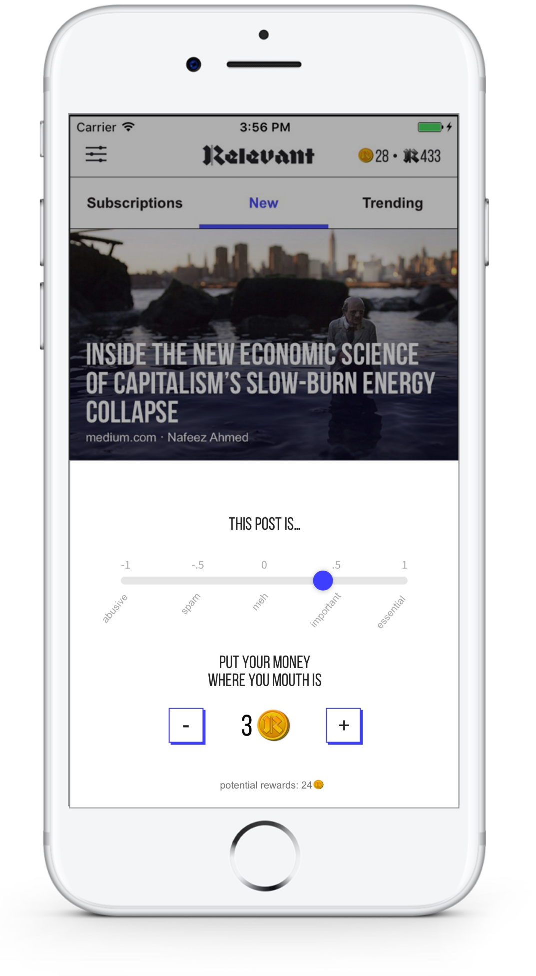


Figure 3: Rating UX

*note: prediction market is only necessary in the case where the content being rated requires time to consume — ex: long-form text posts or videos. A curation market for images, memes, tweets does not necessarily require a prediction market because the content can be rated immediately. However a similar dynamic can still be useful to separate 'n00b' and expert opinions.

3.3 Rating Content

When the app detects that users have read an article, they are prompted to rate it (see Fig. 3). Users' ratings are weighted by their respective reputation scores and summed to produce a rating for the post.

$$postRating = \sum_{i}^{n} rating_i * userReputation_i$$

Like in the prediction market, users rating posts can stake coins on their rating. This has no effect on the article rank, however does increase the users 'share' in the potential reward payouts for the post. This allows us to rely on a non-fungible reputation system for rating and organizing posts and at the same time encourages users to HODL Relevant Curation Tokens in order to earn greater portions of the rewards.

3.4 Computing Rewards

Every hour, we mint new Relevant Protocol Tokens based on a 10% yearly inflation rate, convert them to Curation Tokens and allocate a portion to the *curationRewardFund*, a portion to the *predictionRewardFund* and a final portion as reputation miner rewards.

We then select posts that have reached their *payoutTime* (currently set to 3 days after post creation) and calculate rewards allocated to each post and distributed them to 'predictors' and 'raters'.

3.4.1 Prediction Rewards

We would like to reward predictions that surface the best and worst content (worst because we would like to flag spam content early on). So if predictions are made on a scale of -1 to 1, we do not reward predictions of 0 and give the most rewards for accurate predictions of -1 or 1. We allocate rewards to each post independent of the number of predictions in order to encourage predictions for all posts. To track negative and positive rewards, we split the *predictionRewardFund* into positive and negative funds.

Computing prediction rewards

1. If the post rating is positive, compare it against other positive posts to determine its share of the positive prediction rewards
2. If the post rating negative, compare it against other negative posts to determine its share of negative prediction rewards
3. Distribute post prediction rewards to participants in the post's prediction game based on the following criteria:
   (a) Accuracy of prediction
   (b) Time of prediction (earliest prediction are rewarded first)
   (c) Number of staked curation tokens (the more tokens a user stakes, the bigger the share)

3.4.2 Curation Rewards

Each post is allocated a portion of the *curationRewardFund* based on its rating compared to other posts in the recent time window. Only positively ranked posts are considered for curation rewards.

Computing curation rewards: (see exact formulas in section 6)

1. Adjust rating for predictability (if post has low prediction score, increase rating, if post has high prediction, decrease rating)
2. Compute the post's share of rewards by comparing its rating to other posts in the recent time window
3. Distribute post rewards to users that rated the post based on the following criteria:
   (a) Time of rating (earliest ratings get the most rewards, last ratings get least)
   (b) Number of staked curation tokens (the more tokens a user stakes, the bigger the share) — this encourages users to HODL tokens

3.4.3 Rewarding Commentary

There is no additional financial rewards for users to add 'commentary' or 'text review' for a post. This would create a strong incentive to game the system and result in a large quantity of poor-quality text submissions [3]. After all, our goal is to improve curation of information, not to make it harder. Users who offer commentary and analysis have a chance to earn reputation if their contributions prove to be useful, but they also risk losing reputation. This way we have a clear distinction between social/altruistic benefits and commercial rewards. Of course once users earn a high reputation they are incentivised to participate in rating other's contributions and earn rewards.

3.5 Incentive Attack Scenarios

3.5.1 Bots

It is possible for users to try to game the system in order to win a greater share of the curation rewards. This is bad because it could skew the our quality metric. We can avoid this threat by making it more profitable and easy for bots to participate in the prediction market and difficult to participate in rating content.

Possible solutions:

- Require a minimum reputation score to participate in curation rewards. Since reputation can only be earned by creating posts with quality commentary, it will be difficult for bots to earn enough reputation to participate in curation activities.
- Reputation penalties for high-ratings of poor quality content. If bots are attempting to rate a large quantity of content, they will end up accruing more penalties than rewards.
- Rate-limiting is an effective way to limit bot activity. Users are only allowed to contribute 10-20 ratings a day.

3.5.2 Voting Rings

It is possible that users will attempt to form voting rings and coordinate high ratings of specific posts in order to reap high curation rewards. This attack is mitigated for the following reasons:

- It will take time and energy for participants to earn enough reputation to make a significant impact.
- Posts that have unnaturally-high ratings will be easily detectable and high-reputation users will be able to downvote them to correct the behaviour.
- Users will be able to analyze blockchain data and detect voting rings. Participants risk being exposed and blacklisted via governance mechanisms.

4 Experiments and Case Studies

We plan on building a few case studies in order to test protocol assumptions. To truly test economic assumptions, we plan on launching several experiments using continuous bonding curves [13]. Participants will be able to deposit Ethereum into a smart contract in exchange for newly-minted Curation Tokens. New tokens will be created according to a predefined bonding curve, with the price increasing as more users join the market. Additional tokens will be minted based on a 10% annual inflation rate and distributed to participants as curation rewards.

Curation game dynamics:

1. Deposit Ether into contract in exchange for curation tokens based on a predefined price
2. Receive curation tokens
   (a) Do nothing
      i. lose 10% a year to inflation
      ii. earn Ether as token price increases when more users join
   (a) Participate as a mediocre curator
      i. lose 10% to inflation

ii. earn Ether as token price increases when more users join

iii. earn rewards proportional to 10% of investment

(a) Participate as a great curator

i. lose 10% to inflation

ii. earn Ether as token price increases

iii. earn rewards greater than 10% of investment

3. Withdraw Ether based on a vesting time period and price dictated by the bonding curve

As a first experiment, we will create a feed of blockchain-related news and invite the Ethereum community to curate content.

5 Future Research

Interoperability between multiple reputation systems

- Reputation for reputation systems
- Explore using a single Eigentrust community-wide reputation metric and derive topic reputations from karma-based voting. (karma reputation is easier to track in real time, but more susceptible to manipulation)

Governance mechanisms for combining reputation systems

- UX for creating, managing and interfacing with multiple reputation systems
- Dynamic ontologies — how to account for evolving and niche reputation context
- Eigentrust can only be used to compute a limited set of topics/contexts. As the number of topics increases, the computation time will approach O(n2)
- Can we safely combine Eigentrust and karma-based reputations?

Alternative Reputation Systems

- Personalized EigenTrust
- ML models to compute reputation and detect malicious nodes

6 Appendix: Computing Curator Rewards

We first compute the post share of the curation rewards

$$postShare = \frac{postRating * predictabilityAdjustment}{totalPostRank}$$

where $totalPostRank$ is a sum of all post ranks computed for a rolling time window of 6 days (older post's relevance decays to 0 after 6 days).

We then calculate the post payout:

$$postPayout = rewardFund * postShare$$

The users that rated the post then share in the post's rewards: $curatorWeight$ is computed at the time the user submits their rating and is based on the order of the rating and the amount of coins staked on the rating.

$$curatorWeight = \frac{coinsStaked}{totalCoinsStaked}$$

So if we assume each voter stakes 1 coin for their vote, their $userWeight$ would decrease as follows:

$$userWeight_1 = 1, userWeight_2 = \frac{1}{2}, ..., userWeight_5 = \frac{1}{5}, ...$$

If Alice was the 5th one to submit her rating, her curatorWeight will be 1/5. However if she chooses to stake 10 coins, she can increase her weight to $\frac{10}{(4+10)} = \frac{10}{14}$

To get the final user share of the post's payout we divide user weight by the sum of all weights:

$$userShare = \frac{userWeight}{\sum_i^n (userWeight_i)}$$

And finally, compute the user reward payout:

$$userPayout = userShare * postPayout$$

References

[1] V. Cahill, E. Gray, J. M. Seigneur, C. D. Jensen, Yong Chen, B. Shand, N. Dimmock, A. Twigg, J. Bacon, C. English, W. Wagealla, S. Terzis, P. Nixon, G. Di Marzo Serugendo, C. Bryce, M. Carbone, K. Krukow, and M. Nielson. Using trust for secure collaboration in uncertain environments. *IEEE Pervasive Computing*, 2(3):52–61, July 2003.

[2] Xinxin Fan, L. Liu, Mingchu Li, and Zhiyuan Su. EigenTrust++: Attack resilient trust management. In *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, pages 416–425, October 2012.

[3] Randy Farmer and Bryce Glass. *Building web reputation systems.* O'Reilly Media, Inc., 2010.

[4] M. Glenski, C. Pennycuff, and T. Weninger. Consumers and Curators: Browsing and Voting Patterns on Reddit. *IEEE Transactions on Computational Social Systems*, 4(4):196–206, December 2017.

[5] Mike Goldin. Token-Curated Registries 1.0, September 2017.

[6] Audun Jøsang, Roslan Ismail, and Colin Boyd. A Survey of Trust and Reputation Systems for Online Service Provision. *Decis. Support Syst.*, 43(2):618–644, March 2007.

[7] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The Eigentrust Algorithm for Reputation Management in P2p Networks. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 640–651, New York, NY, USA, 2003. ACM.

[8] Peter Kollock. The Production of Trust in Online Markets. In E. J. Lawler, M. Macy, S. Thyne, and H. A. Walker, editors, *Advances in Group Processes*, volume 16. JAI Press, 1999.

[9] Geert Lovink, Nathaniel Tkacz, Joseph Reagle, Dan O'Sullivan, Lawrence Liang, Amila Akdag Salah, Cheng Gao, Krzystztof Suchecki, Andrea Scharnhorst, R.Stuart Geiger, Edgar Enyedy, Peter B. Kaufman, Johanna Niesyto, Hans Mathews, Scott Kildall, Nathaniel Stern, Nicholas Carr, Alan Shapiro, Florian Cramer, and Shun-Ling Chen. *Critical Point of View: A Wikipedia Reader.* June 2012. DOI: 10.2139/ssrn.2075015.

[10] Robert K. Merton. The Matthew Effect in Science. *Science*, 159(3810):56–63, January 1968.

[11] Larry Page, Sergey Brin, R. Motwani, and T. Winograd. *The PageRank Citation Ranking: Bringing Order to the Web.* 1998.

[12] Simon de la Rouviere. Introducing Curation Markets: Trade Popularity of Memes & Information (with code)!, May 2017.

[13] Simon de la Rouviere. Tokens 2.0: Curved Token Bonding in Curation Markets, November 2017.

[14] Toshio Yamagishi. *Trust: The Evolutionary Game of Mind and Society.* Springer, Tokyo ; New York, 2011 edition edition, November 2011.