An Analysis of Reputation: Currency of the Stack Exchange Network

Apoorva Dornadula
apoorvad@stanford.edu

Megha Jhunjhunwala
meghaj@stanford.edu

Bryce Tham
bjtham@stanford.edu

1. Abstract

The advent of online discussion forums has given rise to a new kind of collaboration-based network known as social question-and-answer (SQA) networks. In a typical SQA network, users post questions and seek responses from other users, after which the community votes on which of the responses adequately answers the provided question. This results in a voting-based system by which user answers are ranked, which contributes to the status of the user. However, such a system is activity-based, which means computing the reputation of users is impossible on large networks without an established reputation system. Or is it? This report analyzes various features of a subset of SQA sites on the Stack Exchange network and determines multiple different metrics to evaluate user status. Our results show that user reputation can be reasonably modelled by the quantity (not just the quality) of their answers and demonstrate a meaningful way to model user reputation on other SQA networks (like Piazza or Quora) using only the structural characteristics of the networks themselves. We used a number of features such as the fraction of upvotes and downvotes, timerank, Stack Exchange reputation, and centrality measures to determine whether an answer is chosen as the accepted answer for a question. We were able to predict the answer with 67% accuracy.

2. Related Work

There has been extensive previous research on the subject of reputation and user status analysis within SQA networks. The first two papers below discuss ways in which user status can be defined, while the latter two papers discuss how reputation affects user behavior. These papers present prior ideas for modeling reputation using network properties and how these results might be evaluated using existing activity-based reputation systems currently employed by SQA networks.

2.1. A Model of Collaboration-Based Reputation for the Social Web

This paper by McNally et al.[1] offers one way to model user-to-user interactions in SQA networks. In this study, a user-to-user collaboration network was created to represent user responses to user questions, and weights were added to each question-answer pair corresponding to the proportion of votes received. Various models were used to evaluate the reputation of each user including PageRank and weighted sum approaches, both of which we will use for our own analysis. The paper then compares these models correlate to Stack Exchange's own reputation system according to some ground-truth metric and found that the weighted sum approach performed the best among the models evaluated. In our analysis, while we do not intend to find a model that exceeds Stack Exchange's reputation system, the methods discussed in this paper will be used to produce a computational model from which reputation can be evaluated. It would also be interesting to further explore these approaches using other models and evaluation metrics.

2.2. An Empirical Analysis of a Network of Expertise

This paper by Le et al.[2] focuses on how to model and characterize a user's expertise based on the network and its centrality measures. They analyzed the network using power-law degree

distributions, reciprocity patterns among users, linear models, and PCA (Principle Component Analysis). This paper performed a linear regression analysis to determine which centrality measures best correlates to reputation. The best centrality measures that correlated with the reputation were the number of answers a user gave and the in degree of the node. Second best correlation came from the PageRank and HITS Authority measures. We also decided to use centrality measures used in this paper to correlate it to Stack Exchange's reputation. We took it a step further by taking the results of these measures as features of a logistic regression classifier to identify whether or not an answer is an accepted answer of a question.

2.3. Analysis of the reputation system and user contributions on a question answering website: Stackoverflow

This paper by Attias et al.[3] does a thorough analysis of the participation patterns of high and low reputation users in Stack Overflow. Their study concluded that the user contribution in the first months of activity is a good indicator of their contribution to the community in the future. This analysis was used to design a machine learning model to predict the long-term contributors to the question and answering community. First, they analysed the distribution of number of users over user reputation in the last four years of data. As expected, the distribution follows a log-linear relationship where very few users have very high reputation and a large number of users have very low reputation. Next, the authors performed a network analysis to study user interaction in the SO network. They considered networks resulting from three different types of interactions between users, considering edges between the user who asked the question and: 1) any user who answered 2) the user whose answer was accepted and 3) any user whose answer was upvoted. We experimented with similar user interaction networks in our study of stack overflow reputation. As concluded in this paper, the reputation of stack overflow was highly correlated with the centrality scores of the user interaction networks.

2.4. Discovering Value from Community Activity on Focused Question Answering Sites: A Case Study of Stack Overflow

This paper by Anderson et al.[4] focuses on understanding the community activity, in this case Stack Overflow, that results in a set of answers. They explore metrics such as reputation and speed of answering a question to predict two main questions. One, is whether a question and its answers have long term value and another is whether a question has been sufficiently answered. Predicting these tasks was done by constructing 27 features (such as questioner's reputation, total number of page views, average answerer's reputation, etc). These features were used to construct a logistic regression classifier to predict the long-term value of a question as well as whether a question has been sufficiently answered. They found that the number of answers was a strong indication that a question would be long term. We are interested in further investigating this space by first constructing our own measure of status. Then, we would like to see how well it works when predicting which answer will be accepted. We will be using some of the prediction techniques that this paper uses in accomplishing our task. This paper was the motivation behind our interest in looking at answers to a question and its relationship to reputation.

3. Dataset & Representation

Before analyzing the network properties used to evaluate reputation, it is important to first understand the details behind the underlying data provided. This section gives a full description of the data, how the data is being parsed and cleansed, and how it is being represented as a graph. Summaries of each dataset are also detailed below.

3.1. Dataset Description

The dataset we wish to use in our project is the Stack Exchange Data Dump provided by Stack Exchange released on August 31, 2017. The data dump includes anonymized information from over 300 Stack Exchange sites. For each site, data is available for badges, comments, posts, post history, post links, users, and votes. For the purposes of this project, only information on posts, users, and votes will be used. The point of using only a limited amount of information from each Stack Exchange site is to be able to generalize our results to non-SQA networks with the constraint that these networks must have some sort of voting-based system in place.

- *Posts*. Posts can be either questions or answers. Users can upvote/downvote, favorite, or comment on posts. In addition, a user can mark a post as "accepted" if it has satisfactorily answered a question. The data dump includes all the above information about each post as well as the author of each post, its content, and the time it was created.
- *Users*. Each user has the opportunity to provide supplemental information about themselves. Most importantly, each user has a reputation score that describes how "trusted" they are on the site. The data dump includes all the above information as well as the number of views and votes each user has gotten.
- *Votes*. Users can upvote/downvote individual posts. Votes also include favorites and bounties, among others. The data dump includes information about which uses voted for which posts and what type of vote was cast.

The full data dump (including complete schema information) can be viewed on the Internet Archive page where it is currently being hosted (https://archive.org/details/stackexchange).

Because the data dump contains information on hundreds of Stack Exchange sites, it is important to narrow down the domain and select only a few sites for which to test our models. In choosing which sites to use, we wanted to identify sites with various backgrounds so as to take into account any community differences that might exist between them. The final set of sites that we have chosen for our network analysis are biology (https://biology.stackexchange.com), cs (https://cs.stackexchange.com), and movies (https://movies.stackexchange.com).

3.2 Data Cleansing & Summary

The data dump provided by Stack Exchange is in XML format and therefore needs to be processed and converted into readable form. There are also instances in which important data was missing rendering some information unusable. We explain how the data from each site was parsed and cleansed below.

	#users	#posts	#votes
biology	24059	38039	205557
cs	59394	48003	232729
movies	36143	46966	399443

Table 1. Number of users, posts, and votes for each Stack Exchange site (before data cleansing).

Using Python's built-in ElementTree XML API, each of the three XML files (posts.xml, users.xml, and votes.xml) for each site was converted into Python dictionary form with every entry being a row defined by the XML. Through this process we were able to determine individual statistics about the activity level of each site including number of users, posts (questions and answers), and votes (all types). These statistics are summarized in *Table 1*. Notice how the cs site contains many more users than posts, implying a high inactive user rate. Analyzing this site will provide some insight into the community structure of networks with a high number of inactive users (like Twitter). Also notice how the

movies site contains significantly more votes compared to the biology or cs sites, implying that voting activity is relatively high within that community. Analyzing this site will provide some insight into the community structure of networks in which users vote much more often than they post (like Reddit). Overall, the biology site seems to have the lowest amount of activity out of the three.

With each site in Python dictionary form, the data must then be organized into classes. The following classes were used for each site:

- *User*. Each user is identified by a unique user ID. The key statistic stored for each user is their reputation. A set of each user's answers and questions are also stored by ID.
- *Answer*. Each answer is identified by a unique post ID. Information stored includes the user ID of the answer author, the question ID of the parent post, the number of upvotes and downvotes, and the creation date. A separate flag is set to True if the answer is accepted.
- *Question*. Each question is identified by a unique post ID. Information stored includes the user ID of the question author, the list of answers responding to the question by ID, the number of upvotes and downvotes, and the creation date. The list of answers is sorted by timestamp.
- *Dataset*. There is one dataset object for each site containing three dictionaries: one for users, one for answers, and one for questions (all by ID).

The classes described above were populated first by iterating through each of the converted dictionaries and constructing empty User, Answer, and Question objects for each Dataset, then filling in any missing information by iterating through each item in the Dataset a second time. Some posts and votes were discarded due to incomplete data; a total of 1719 (4.5%) of posts from the biology site, 1211 (2.5%) of posts from the cs site, and 4725 (10.0%) of posts from the movies site were missing critical information that was vital to our analysis. The final number of users, questions, and answers for each site after data cleansing can be found in *Table 2*.

	#users	#questions	#answers
biology	24059	16524	19796
cs	59394	20786	26006
movies	36143	16409	25832

Table 2. Number of users, questions, and answers for each Stack Exchange site (after data cleansing).

3.3. Graph Representation

We will be modeling user's questioning and answering behavior by representing users, questions, and answers in two different graphs.

	Nodes	***Edges (i → j)***
Graph 1	User ID	j answered i's question
Graph 2	User ID	j's answer to i's question was accepted

Table 3. Description of graph representations of the dataset.

We will also be experimenting with setting the following edge weights on the graphs:

- *No Weight*. A graph with no edge weights simply means that the centrality of each node is directly correlated with its degree.
- *Proportion of Upvotes*. The paper by by McNally et al. found the proportion of upvotes an answer receives is a strong indicator for user reputation. Let $u(q_i, a_j)$ be the number of upvotes received for answer a_j to question q_i where i is the author of the question and j is the author of the answer. Then for every question q_i the weight of each edge is:

$$w_{q_i}(i \to j) = \frac{u(q_i, a_j)}{\Sigma_k u(q_i, a_k)}$$

- *Reciprocal of Timerank*. The paper by Attias et al. found that high-reputation users are more likely to respond first to a question. Thus, it may be possible to model reputation as a function of time rank (i.e. the relative order of responses to a particular question). Let $t(q_i, a_j)$ be the timerank of answer a_j to question q_i where i is the author of the question and j is the author of the answer. Then for every question q_i the weight of each edge is:

$$w_{q_i}(i \to j) = \frac{1}{t(q_i, a_j)}$$

4. Reputation Metrics & Methods

Below are a description of metrics and methods we used to evaluate user reputation using the network properties of the graph representation described above. We will be using the CS site as the example dataset used to illustrate our metrics and methods, but note that we ran these evaluations on our other datasets as well. First, we will describe the properties of our networks in terms of reputation and degree distribution. Then, we will introduce the algorithmic methods we used to evaluate user reputation. Note that all the plots below are normalized.

4.1 Reputation & Degree Distribution

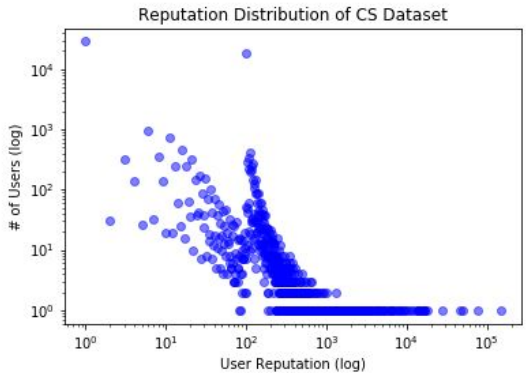

Figure 1a. User reputation vs. number of users with that reputation.

Figure 1a shows the distribution of reputation across all users in the network. Notice how the top 2 highest frequencies in the figure are 1 and 100. This can be explained by the fact that new Stack Exchange users receive a starting reputation of 1 and returning Stack Exchange users who are new to a specific site receive a starting reputation of 100. Note how by ignoring all users with reputation less than 100, the reputation distribution resembles what we might expect, with a majority of users having low reputation and few users having high reputation.

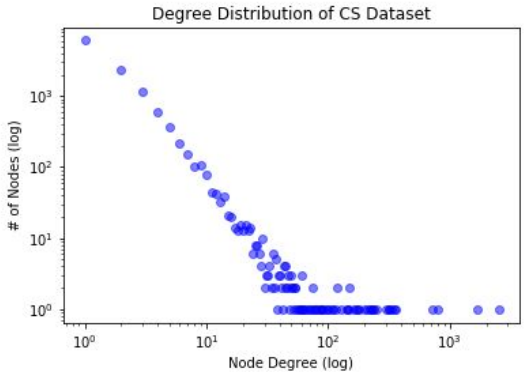

Figure 1b. Node degree vs. number of nodes with that degree.

Figure 1b shows a similar trend with the distribution of degrees across all nodes in the network, with a majority of nodes having low degree and few nodes having high degree. From these 2 figures we can infer that there are only a handful of users who are highly active. We can use this information to further explore the reputation evaluation methods described below.

4.2 Weighted Sum Approach

McNally et al. described a weighted sum approach that they used to model user status against the Stack Exchange reputation system. The weighted sum reputation of a node is simply the sum of the weights of all incident edges of that node. This measure of reputation is directly proportional to the degree of each node and the value of the edge weights assigned.

$$WeightedSum(i) = \sum_{j \to i} W(E_{j \to i})$$

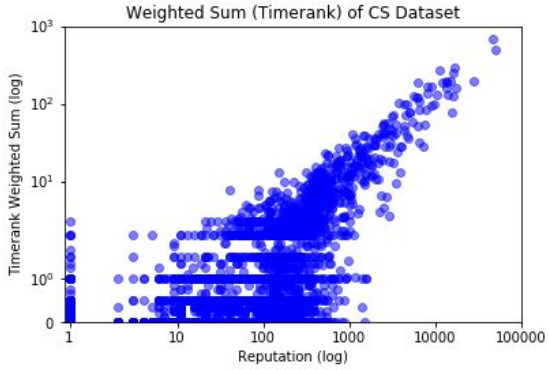

Figure 2a. Reputation vs. timerank weighted sum.

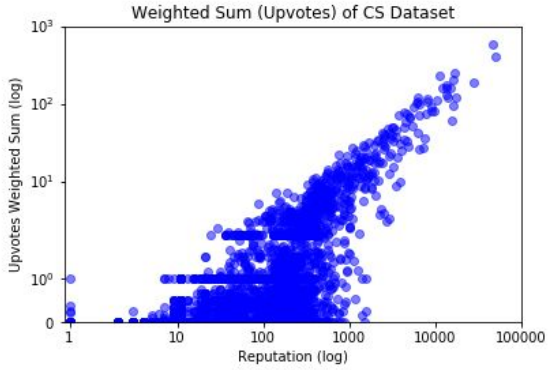

Figure 2b. Reputation vs. upvotes weighted sum.

Figure 2a was generated on the network whose edges are determined by time rank and *Figure 2b* was generated on the network whose edges are determined by upvotes. Both methods correlate well with Stack Exchange's reputation system for users who score highly in this method. This means that high reputation users who answer often, answer first, and have their answers voted up are more likely to have a high reputation. Another interesting result that we observed from the figures 2a and 2b is that a larger number of anomalous users (with stack exchange reputation 1 but a comparatively higher centrality score) were detected when we used time rank as the edge weights. This is possibly due to the fact that such anomalous users were using a spamming strategy to answer first on many questions. When we used fraction of upvotes as the edge weights, our reputation scores were much more correlated with the actual stack exchange reputation scheme.

4.3 PageRank

Both McNally et al. and Le et al. used the PageRank[5] algorithm as a candidate measure of reputation. PageRank is known as a "flow" model because the status of the user who responds to a question depends in part on the status of the user who asks the question.

$$PR(i) = \sum_{j \to i} \beta \frac{PR(j)}{degree(j)} + (1 - \beta)\frac{1}{n}$$

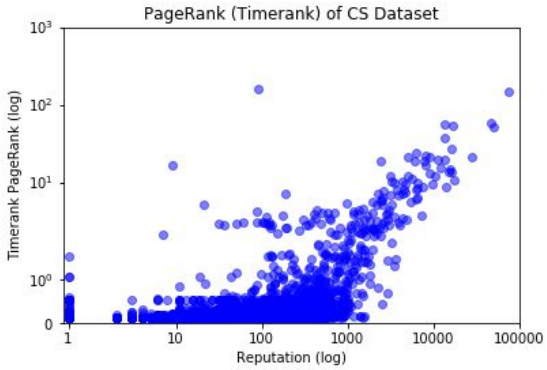

Figure 3a. Reputation vs. timerank PageRank.

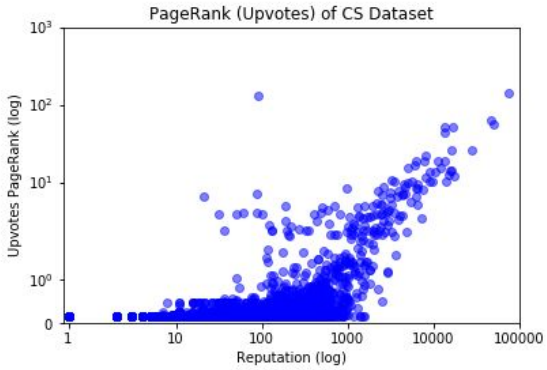

Figure 3b. Reputation vs. upvotes PageRank.

Figure 3a was generated on the network whose edges are determined by timerank and *Figure 3b* was generated on the network whose edges are determined by upvotes. For both plots, the correlation does not seem strong for users with low reputation. This is likely due to the fact that users with high reputation do not discriminate between low-reputation users and high-reputation users in terms of whose

questions they respond to. As a result, the status of low-reputation users "flow" down to high-reputation responders. That being said, the correlation between PageRank and user reputation seems to be strongest among users with high PageRank, even if the number of such users is relatively small, possibly because low-reputation users are more hesitant to answer questions asked by these users. Notice how similar the correlations in the two plots are regardless of whether we choose timerank or upvotes as edge weights.

4.4 HITS Authority

In addition to PageRank, Le et al. uses the HITS[6] algorithm to model reputation. The HITS algorithm outputs two different scores for each node: a hub score and a authority score. We use only the authority score for our analysis. *Figures 4a-4b* show how well HITS Authority correlates with user reputation.

$$auth(i) = \sum_{j \to i} hub(j) \; ; \; hub(i) = \sum_{i \to j} auth(i)$$

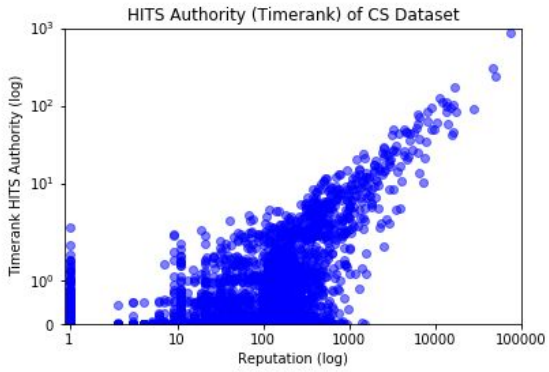

Figure 4a. Reputation vs. timerank HITS Authority.

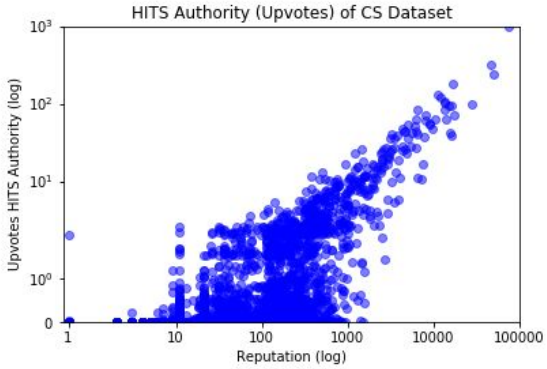

Figure 4b. Reputation vs. upvotes HITS Authority.

Figure 4a was generated on the network whose edges are determined by timerank and *Figure 4b* was generated on the network whose edges are determined by upvotes. While correlation in both plots are strong, there seems to be a lot of noise for users with low HITS Authority. This is expected, as users who have only answered a few questions may have highly differing reputations depending on how their answers were received, and a single answer may affect the reputation of these users greatly. As with the PageRank algorithm, high HITS Authority tends to correlate with high reputation.

4.5 Degree Centrality

In addition to the methods described above, there are several other centrality measures that might be used to evaluate reputation. Le et al. uses betweenness centrality as a candidate measure of reputation, but our results show that this measure has low overall correlation (a similar measure, known as closeness centrality, also correlated poorly). We will use degree centrality instead. *Figure 5* shows how well degree centrality correlates with user reputation.

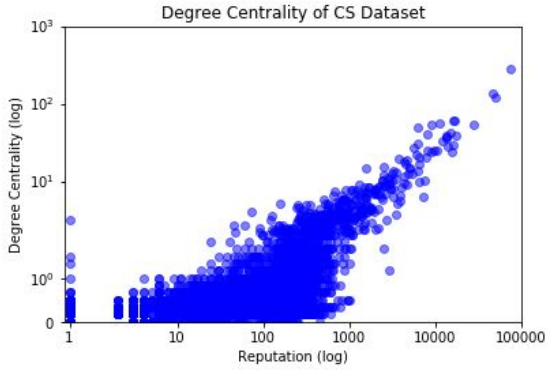

Figure 5. Reputation vs. degree centrality.

Because degree centrality depends only on the degree of each node, *Figure 5* was generated on the graph with no edge weights. The degree centrality of a node is simply the number of edges incident to that node. This results in a centrality measure that only looks at how often a user answers questions. The correlation to user reputation is quite strong, which one might expect because user reputation is directly derived from user activities. However, it is interesting to note that this implies the quantity of answers, and not necessarily the quality of answers, is what drives user reputation.

5. Reputation Correlation Analysis

Given the reputation metrics and methods described above, we now need to take a closer look at which methods best correlate with user reputation overall. This section discusses which of the methods are better at modelling reputation for each of the 3 sites (biology, cs, movies) we are interested in. For each method, let $M(n)$ be the set of top n users according to that method, and let $R(n)$ be the top n users according to Stack Exchange's reputation system. Then, the correlation with reputation can be computed by the following equation:

$$C_n(M,R) = \frac{|M(n) \cap R(n)|}{n}$$

Figures 6a-c show the overall correlation of each method across the top n users for each 100 users up to 1000 and reveal several trends about our methods across all three sites.

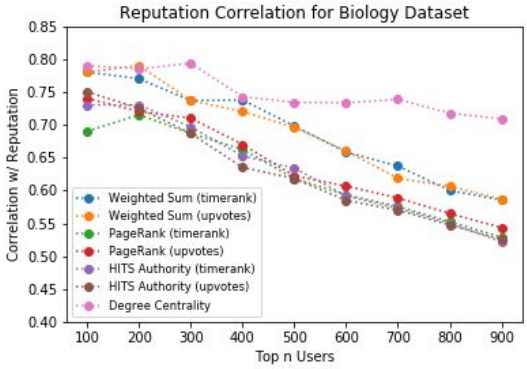

Figure 6a. Reputation correlation for top 1000 users of the biology site.

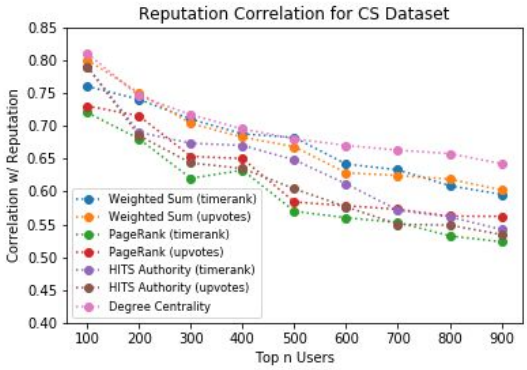

Figure 6b. Reputation correlation for top 1000 users of the cs site.

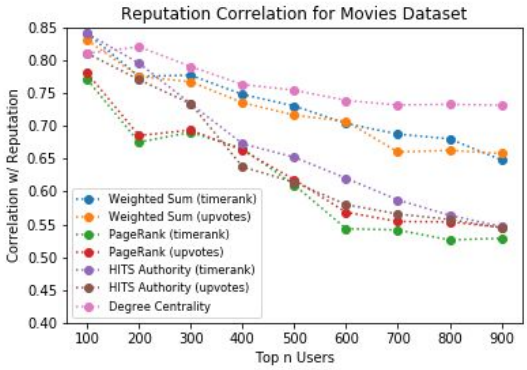

Figure 6c. Reputation correlation for top 1000 users of the movies site.

First, degree centrality correlates best. The weighted sum approach correlates second-best. This result is interesting, as degree centrality is essentially the weighted sum approach if all edge weights were 1. This implies a

somewhat surprising characteristic of these networks: how much a user contributes, regardless of other metrics such as timerank and upvotes, is what correlates with user reputation. Note that in some rare cases, edge weights do matter: the weighted sum (upvotes) approach slightly outperforms degree centrality for the top 200 users of both the biology and cs sites, and both weighted sum approaches slightly outperform degree centrality for the top 100 users of the movies site.

Second, timerank differs little from upvotes. Regardless of which one of weighted sum, PageRank, or HITS Authority methods is used, choosing either timerank or upvotes as edge weights yield similar results. This is likely due to the fact that these two measures are correlated; users who respond to questions first tend to receive more upvotes. Unfortunately, neither one of timrank or upvotes consistently outperforms the other; timerank performs worse in weighted sum and PageRank but performs better in HITS Authority. This does, however, raise another interesting research question for another paper, namely whether we can predict the quality of a response using only its timerank.

Third, correlation decreases as n increases. Certainly, no method is perfect, and the plots in the previous section show that these methods correlate best for users with high reputation. This is an acceptable result,, as we are often only interested in identifying users with the highest reputation anyways. Moreover, long-term analysis reveals that all measures retain a correlation of at least 0.5 even after looking at the top 4000 users. Still, it would be interesting to find a method with consistently high correlation with user reputation.

6. Predicting Accepted Answers

We used a logistic regression classifier to predict which answer was accepted for a particular question. The features we used are listed in Table 4, along with their respective weights as learned during training. Our features were inspired by [2] and [4]. The features correspond to the answerer.

Feature	**Weight**
# Questions Asked	0.000835
# Questions Answered	-0.000703
Stack Exchange Reputation	1.847e-05
HITS - Hub Score	-2.10e-05
HITS - Authority Score	0.00858
PageRank of the Answer	0.000463
Time Rank of the Answer	-0.510
Fraction of Upvotes of the Answer	1.67
Fraction of Downvotes of the Answer	-0.914

Table 4. Features for our logistic regression classifier.

We found that the fraction of upvotes, fraction of downvotes, and the timerank of an answer were the most influential features based on the weights learned by the model.

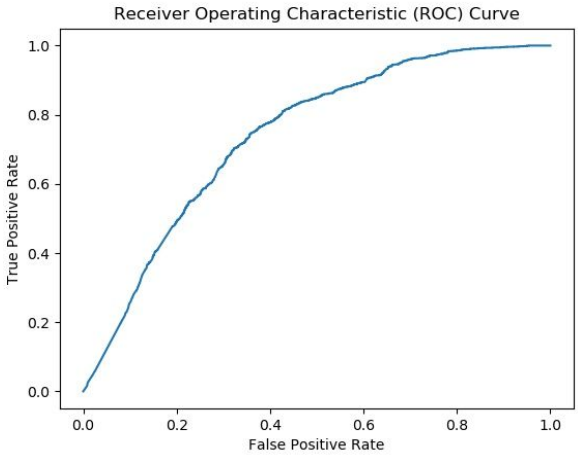

Figure 7. RoC curve of our logistic regression classifier.

The dataset we used for this classifier had 25k labeled question and answer pairs. 80% of our dataset was used for training and the remaining 20% was used as our test set. Our classifier achieved an accuracy of 68.25% on the test set. The RoC curve for our classifier is displayed in Figure 7. Our results show that the fraction of upvotes and the fraction of downvotes received by an answer are more influential than the actual reputation of the answerer. This was contradictory to our initial intuition that accepted answers were chosen based on a user's reputation.

7. Conclusion

In this report, we used a number of centrality measures to analyze how well each method correlates with Stack Exchange's user reputation system. We found the following trends in our results. First, the degree centrality of each node correlates best with reputation, followed by the weighted sum approach, suggesting that the quantity of answers a user provides is what matters most. Second, because timerank and upvotes trend similarly, neither one results in a better correlation with reputation. Third, all our measures depreciate in effectiveness when considering lower-reputation users, meaning our methods work best when analyzing the top users in the network. We also created a classifier to predict which answer becomes an accepted answer for a question based on features such as HITS score, fraction of upvotes, and timerank.

8. References

1. McNally, Kevin, Michael P. O'Mahony, and Barry Smyth. "A model of collaboration-based reputation for the social web." *Seventh International AAAI Conference on Weblogs and Social Media*. 2013.

2. Le, Truc Viet, and Minh Thap Nguyen. "An empirical analysis of a network of expertise." *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM, 2013.

3. Movshovitz-Attias, Dana, et al. "Analysis of the reputation system and user contributions on a question answering website: Stackoverflow." *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM, 2013.

4. Anderson, Ashton, et al. "Discovering value from community activity on focused question answering sites: a case study of stack overflow." *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.

5. Page, Lawrence, et al. *The PageRank citation ranking: Bringing order to the web*. Stanford InfoLab, 1999.

6. Kleinberg, Jon M. "Hubs, authorities, and communities." *ACM computing surveys (CSUR)* 31.4es (1999): 5.