**Project Milestone: Chatous Network Analysis**

Alex Fandrianto, Aman Neelappa, Negar Rahmati (Group 36)

## Abstract

We analyze various properties of the Chatous online chat network. First, we attempt to predict user conversation polarity given the word IDs used. As ground truth, friend and reported conversations were used. We determined that logistic regression and Naive Bayes performed comparably to taking the average conversation length as the primary polarity metric. If the actual words were given, NLP-techniques like sentiment analysis could be used to improve performance. Next, we tried logistic regression on two different classes features to predict the edge signs. The first class of feature based on the signs of 2-edge long paths between nodes, and the second class is based on the positive and negative degree of the nodes. Finally, we examined the report network with link-based ranking algorithms to determine, which users were toxic to the system. While the highest ranked users were spam bots, a manually evaluated set of dirty users were indistinguishable from clean users using these algorithms.

## 1. Introduction

Chatous is an online chat platform where people are randomly paired up with a stranger based on some underlying matching algorithm. After each conversation, users have the choice to form a mutual friendship, report the other user, or simply move on to the next chat. The data collected by Chatous presents an interesting opportunity to study various aspects of an online chat network. From our discoveries, we hope to improve Chatous's matching algorithm to optimize each user's experience. To do so, we will attempt to define a metric of positive vs negative user interactions. This will allow us to predict which users will have a good experience with each other and suggest that they be paired together. Additionally, by identifying users with primarily negative interactions, we can verify whether link-based ranking algorithms can be used on the report network to identify bad users in the system.

To define the notion of positive and negative interactions, we begin with the assertion that friendly conversations should be considered positive while reported conversations should be considered negative. However, this still leaves a huge number of interactions without a sign. We attempt to solve this problem by training a classifier that classifies conversations as positive and negative. This would then create a signed network for use in subsequent algorithms.

Currently, Chatous matches users randomly with the preference of matching opposite genders. However, If Chatous could match people intentionally based on users' past conversations with other people, users would be more likely to enjoy their conversations. In order to come up with an appropriate matching algorithm for Chatous, which predicts whether or not two users will enjoy chatting, we train on users' past behavior using two different classes of features and logistic regression classifier. The features are extracted from the conversation network, an undirected graph in which nodes are users and edges are conversations between each pair of users. In the conversation network the edges are labeled positive if the users liked chatting and negative if they didn't. Identification of the positive and negative conversations is based on the algorithms explained above. The first developed method trains on triads, and the second trains on node polarity, the number of positive and negative edges of each node. Later on, the accuracy of training on each of these feature classes are evaluated.

The Chatous community has good and bad users. It would be preferable to detect and isolate the latter set of users from the system as early as possible. To do so, we employ link-based ranking algorithms, which

have been used successfully to rank the authoritativeness of nodes. Generally, directed edges imply trust or endorsement. However, for Chatous's report network, the edges imply distrust. We intend to explore how well these algorithms work on a network predicated on negative endorsements and determine what adjustments should be made. The goal is to find the users that are detrimental to the chat system with a metric more sophisticated than taking simple in-degree (i.e. the number of times a user has been reported). To evaluate the rankings produced, we hope to match the results of manually labeled Chatous users.

## 2. Related Work

In Leskovec et al.[2], the author tries to predict signs of edges in a signed network. In the network, the positive sign indicates a friendly relationship and negative sign indicates an antagonism. A logistic classifier is trained on two different set of features and is used to predict the edge signs. The first class of features is based on triads in the network and the second class of features are based on positive and negative degrees of nodes. In this project, both of these methods are implemented and tested on the Chatous conversation network in order to come up with an appropriate matching algorithm.

Pagerank, HITS, and SALSA are standard link-based ranking algorithms used for directed graphs where directed edges imply trust. They were first proposed for modeling importance in the web graph and have been effective when combined with information retrieval systems. When taken alone, however, it was found that SALSA outperformed the other two algorithms by a large margin [3]. While similar in structure, the algorithms are fundamentally different at what they attempt to model [4]. As Borodin et al. discover, many variations of these algorithms can be proposed to suit the graph one is modeling.

## 3. Data and General Statistics

The Chatous dataset is a 2-week snapshot of user activity corresponding to ~9 million chats from 80,000 users along with their profiles. Since most chats are empty, we decided to focus on the ~1.9 million chats where users actually chatted. For privacy concerns, instead of the full chat log, each chat only has a histogram of words used by each user.

Some approximate schemas for the data follow:

- Chat(**Chat ID,** Fuser ID, Suser ID, Friendship Status, Chat Created Date, Chat Finished Date, Length of Chat, Disconnector, Reported User ID, Reason for Reporting)
- ChatContent(**Chat ID, UserID**, Profile ID, Word Histogram)
- Profile(**Profile ID**, User ID, Time Created, Age, Gender, Location, Location Flag)
- User(**User ID**)

We can identify three kinds of networks in Chatous:

1. **Conversation Network**: an undirected graph with multiple edges. Edges can be turned into weighted edges with weights as a function of the conversation of the two users (eg: length of the conversation, sentiment of the conversation).
2. **Friends Network**: an undirected graph where an edge exists between two people if and only if they are friends.
3. **Report Network**: a directed graph where an edge (a->b) exists iff a reports b. 22038 users are involved in this network, with 3985 reporters and 20266 reportees. Note that 2213 users both received and generated reports.

## 4. Methods

**Positivity Metric**

In order to analyze the positive and negative triads in the network, we need a notion of positivity of edges in the network. We reason that conversations between friends should be deemed to be positive (+1) and the conversations that lead to people being reported should be deemed to be negative (-1). This gives us a tagged dataset that can be used to train an edge classifier. While training is done on only extreme conversations, the hope is that when applied to all conversations, non-friend and non-report conversations will receive intermediate weights on the scale of positivity (between -1 and 1). Again, for this task, we are only considering the conversations which are non-empty. The conversations that are empty can be assigned a polarity based on the profile information, which can be investigated separately.

**Positivity Classification**

We tried out a few different methods to classify conversations as positive and negative.

a. **Logistic Regression based classifier**

The classifier used the following features :

1. Length of the conversation: We expected that positive conversations in general should tend to be longer than negative conversations.
2. 'Balance' of the conversation: We hypothesized that positive conversations should be those where both the users participate commensurately.
3. Jaccard similarity between the word vectors of the two users: We expected that positive conversations would tend to have similar word counters for both the users and reported conversations would not.
4. Jaccard similarity with average 'friendly' and 'reported' conversations: We computed the jaccard similarity of the word vector to the word vector averages over all 'friend' conversations (positive) and the same for all 'report' conversations (negative).

b. **Conversation Length based classifier**

The above classifier gave a much higher weight to the length of conversation feature as compared to the other features. This motivated us to try using that for classification. We plotted the the fraction of conversations below a given length for both friendly and reported conversations and manually inspected them to find a reasonable cutoff.

c. **Using Tf-Idf features**

We noticed that conversation similarity based features were given low weights. We wanted to enrich our length based classifier by also incorporating some information about the conversation content. We realized that it might be worthwhile to focus on certain subset of words in the conversations rather than all the words in the conversation. However, as we only had access to word ids (due to privacy concerns), so we decided on Inverse Document Frequency as a measure of importance. Words were ranked by their Inverse Document frequencies in friendly/reported conversations, retaining only those words above a certain Inverse Document frequency with a suitable cutoff.

d. **Naive Bayes based Classifier**

We finally decided to use a technique often used in spam classification - Naive Bayes. In a nutshell, we compare the conditional probabilities of the conversation being between friends or reporters/reportees and assign the conversation a sign based on which has a higher probability. As is usual in Naive Bayes

assumption each word in the conversation is considered independent of the rest. We used Laplace smoothing for dealing with unseen words and avoiding zero probabilities.

**Edge Sign Prediction**

In order to come up with an accurate matching algorithm which matches users who will enjoy chatting, we analyzed the conversation network. In the conversation network, each node is a user and undirected edges represent a chat between the users. These edges are labeled positive if the two users liked chatting and negative otherwise. As a baseline, the conversations that lasted longer than the median of all conversation lengths (15 words), are labeled positive with all others labeled negative. In the next phase, the conversations are labeled positive and negative using a Naive Bayes classifier explained in the previous section.

The logistic classifier and two different classes of features are used to predict the sign of a potential conversation between two users. Since we hope to match users that our system predicts will have a positive interaction, our system will need to optimize for precision in order to avoid mismatches.

In the first class of features, triads concerning edge (u,v) are analyzed. Assume nodes v, w, and u form a triad. If edge (u,w) is positive and edge (w,v) is negative, there is a positive-negative path between u and v. Then, there would be 4 different cases of paths of length two (positive-positive, positive-negative, negative-positive, negative-negative) from node u to node v based on edge signs. The embeddedness of each edge is also added to the feature vector of edges. In the first feature class, the number of each type of path from u to v is calculated and the feature set is labeled by the sign of edge (u,v). The logistic classifier is used to train on this feature class. This method is called the 8Triads.

The second class of features used to predict the edge sign is based on the negative and positive degrees of each node. Thus for each edge (u,v), the number of the positive and negative edges of each of the nodes u and v, the degrees of nodes u and v, as well as the embeddedness of edge(u,v) are calculated and added to the feature vector. This method is called Degree Polarity. The label is based on the sign of edge(u,v).

Each model was tested by two different approaches. The two approaches for testing the models are explained in the results section.

**Link-based algorithms and the Report Network**

We use the following link-based algorithms to analyze the report network and construct authoritativeness scores. The higher the score, the more toxic the user is to the Chatous network. Except for Pagerank, each algorithm also produces a hubbiness or ‘reporter accuracy’ score, which can be used to determine users who excel at identifying bad users.

Let F(i) be the out-degree of node i, and B(i) be the in-degree of node i.

<u>Pagerank Algorithm</u>

Initialize **p** to be a vector of 1’s

While not converged

$$\forall p_i := \frac{1-d}{N} + d \sum_{j \in B(i)} p_j / |F(j)|$$

Note that d is between 0 and 1, usually 0.85, and N is the # of nodes in the graph.

HITS/SALSA/AverageHub Algorithm
Initialize **a** and **h** to be a vector of 1's
While not converged:

HITS $\forall a_i := \sum_{j \in B(i)} h_j \qquad \forall h_i := \sum_{j \in F(i)} a_j$

SALSA $\forall a_i := \sum_{j \in F(i)} \frac{h_j}{|F(j)|} \qquad \forall h_i := \sum_{j \in F(i)} \frac{a_j}{|B(j)|}$

AverageHub $\forall a_i := \sum_{j \in B(i)} h_j \qquad \forall h_i := \frac{\sum_{j \in F(i)} a_j}{|F(i)|}$

Normalize **a** and **h**

After producing the scores, we will compare the scores with various metrics of user quality. By plotting the two simultaneously, we can see if there is a strong correlation between the two. The stronger the correlation, the more useful the metric is at identifying truly bad users.

## 5. Results and Discussion

**Positivity Classification**
The logistic classifier achieved a 5-fold cross validation accuracy of about 80% over the training set. As the data is significantly large, we tested this over a subset of the test set. We noticed that the classifier was to be more accurate for positive('friend') conversations than for negative ('report') conversations. This was probably due to the skew in the number of positive and negative samples.

For the length based classifier, we then chose a suitable cut off (15 words) by manual inspection of the graphs as the cutoff length beyond which the conversations are to be considered friendly. This simple classifier gave us training error similar to the logistic regression classifier while allowing us to quickly classify the 2 million non-empty conversations.

The Naive Bayes classifier reported a training accuracy of about 65 percent. While the overall accuracy was lower than that obtained for the length based classifier, however it achieved a better balance in making predictions for both positive (70% accuracy) and negative(50% accuracy) classes.

Ideally, we would have liked access to the actual words spoken by the users and run sentiment analysis methods on them. But we could not gain access to those due to privacy concerns and instead had to deal with word vectors where the actual word was replaced with an id. Thus, when we computed the Idfs of the words we found a few surprising results. One was the number of unique words in each conversation. In about 78,000 conversations which were either friendly/reported, we found around 380,000 unique words were used. Further, what was more surprising was the distribution of these words across conversations. We found that about 260,000 words appeared in only one conversation! We sent a random sample of these 260,000 words to the Chatous team for manual inspection. They found that these words were often mis-spellings, links,

foreign words and the like. We realized that we would find it difficult to choose what words are relevant based on the Idf scores alone and that we would need some sense of what these words actually are.

**Edge Sign Prediction**

The conversation network used for testing the error rate of the sign prediction method has 78733 nodes (users), 139180 triads, and 957690 edges (conversations). We first trained on 90% of the feature set and tested the model on the remaining 10% (cross validation). The test set and training set were selected randomly. The overall prediction error as well as the precision for positive edges (arguably, the more relevant metric for our task) are reported in Table 1. This estimation is based on the edge prediction we derived from the behavior of users but in order to have a better estimation about how the algorithm predicts friendly relationships, 10% of edges between friends and enemies, which are explicitly defined by monitoring user behaviors in Chatous, are removed and the logistic classifier is run to train on the new data set. Then, the model is tested on the pairs of friends and enemies that were removed in the training process. This precision estimation is reasonable since it is based on real data about the likability of conversations. The positivity and negativity of the network reported in Table 1 is derived using a Naïve Bayes classifier. We also ran the test on the network that was signed based on the length of conversations. The results of the latter analysis are demonstrated in Table 2.

Table 1. Edges Are Signed Using The Naive Bayes Classifier.

| Feature | Test Method | Error Rate | Positive Edge Precision |
|---|---|---|---|
| 8Triads | Cross Validation | 0.29 | 0.66 |
| Degree Polarity | Cross Validation | 0.15 | 0.86 |
| 8Triads | Tested on Verified friends | 0.29 | 0.71 |
| Degree Polarity | Tested on Verified Friends | 0.29 | 0.80 |

Table 2. Edges Are Signed Using The Conversation Length.

| Feature | Test Method | Error Rate | Positive Edge Precision |
|---|---|---|---|
| 8Triads | Cross Validation | 0.20 | 0.84 |
| Degree Polarity | Cross Validation | 0.31 | 0.65 |
| 8Triads | Tested on Verified friends | 0.28 | 0.72 |
| Degree Polarity | Tested on Verified Friends | 0.21 | 0.84 |

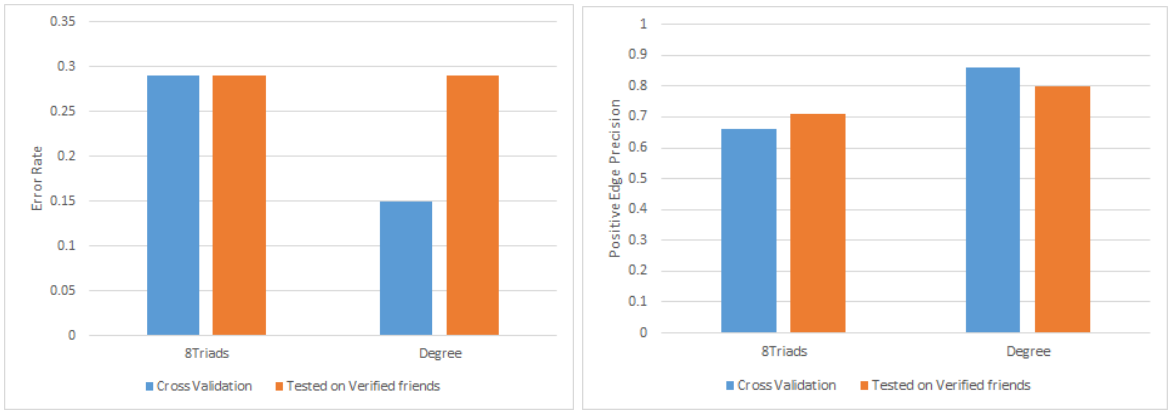



Figure 1. Edges Are Signed Using The Naive Bayes Classifier.

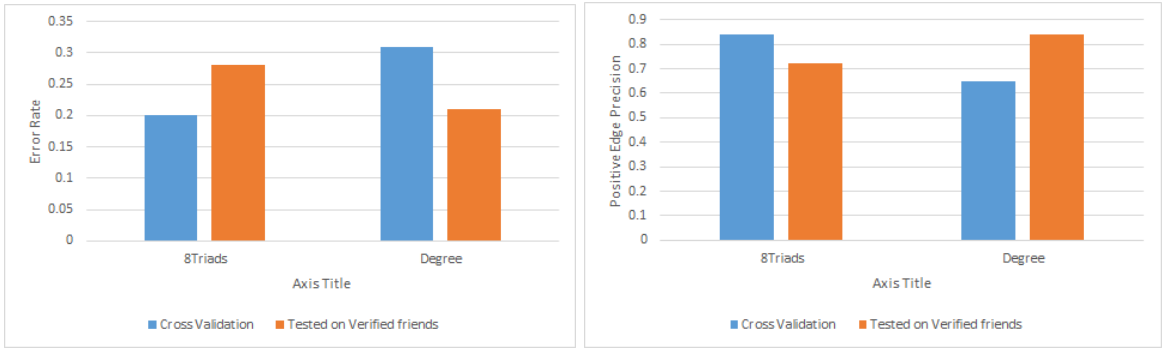



Figure 2. Edges Are Signed Using The Conversation Length.

The cross validation test is only a metric for showing how well the the logistic classifier has predicted the edge signs. However, the error rate and positive edge precision derived from testing on verified friends and enemies, is a metric for showing how successful the “positivity classification” and “edge sign prediction” methods have been in predicting whether or not two people will enjoy chatting. The highest precision of positive edges is gained by running the degree polarity method on a network signed based on conversation length. Generally, the error rate is below 31%, which means approximately 7 out of 10 edges are predicted correctly. Our aim was to predict the positive edges and match people to chat based on this prediction. The positive edge sign prediction tested on the verified friends has a precision higher than 70% in all cases. So, the devised methods are appropriate for predicting positive edges and the best proposed method is the degree polarity method run on a network signed based on conversation length.

**Link-based algorithms and the Report Network**

After implementing Pagerank, HITS, SALSA, and AverageHub, we plotted a user’s reported rank vs the average length of their conversations (Figure 3). While we do see some negative correlation, it is very weak and noisy. The best we can conclude is that reports on users with high average conversation length should be discounted.

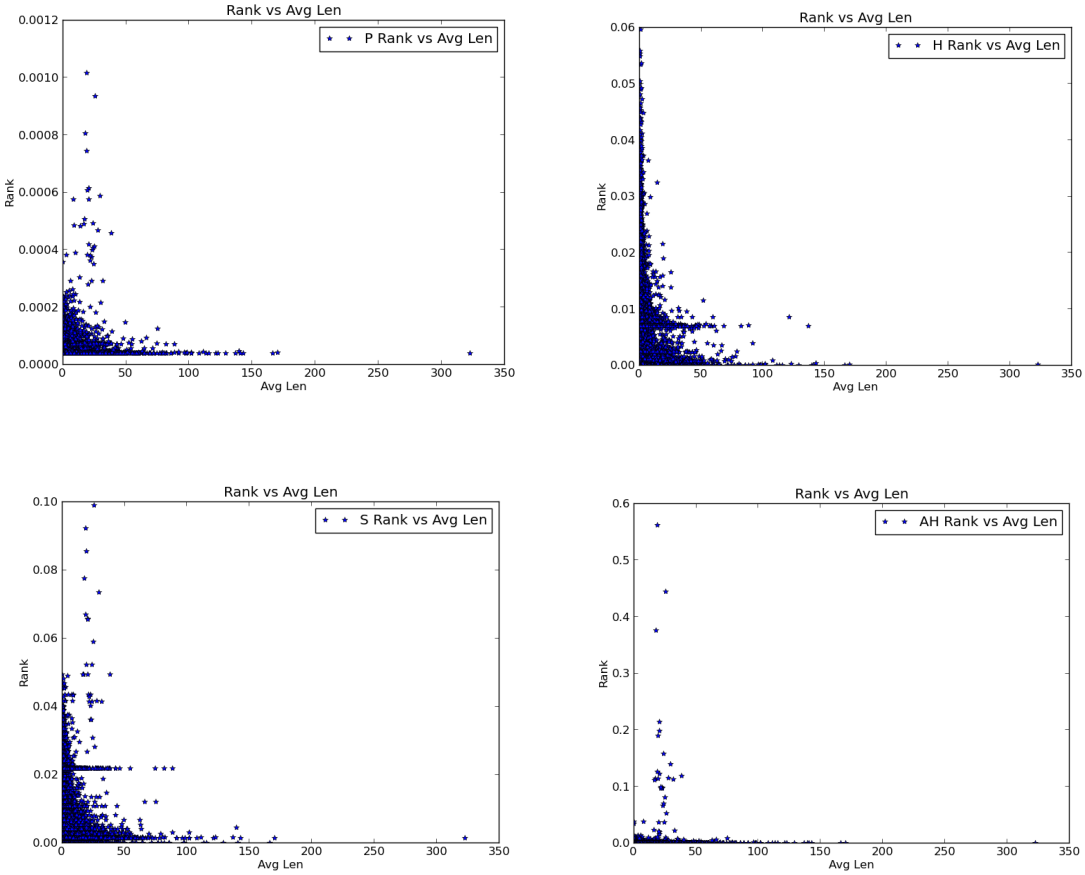
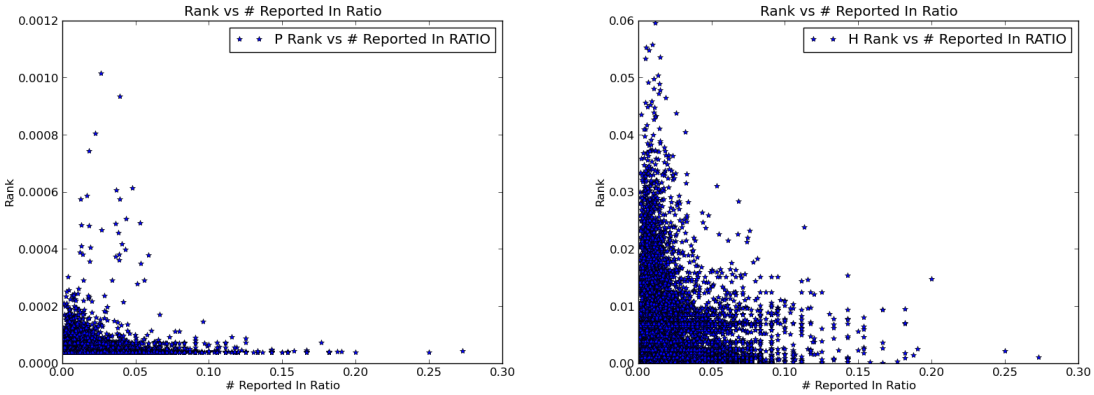



Figure 3. Report Network Rank vs Average Conversation Length

We then plotted rank vs the ratio of conversations a user was reported in (Figure 4). We used the ratio instead of using the raw count of reports because the more conversations a user has, the more times he/she is going to be reported. For this part, we only showed users who had at least 10 conversations.



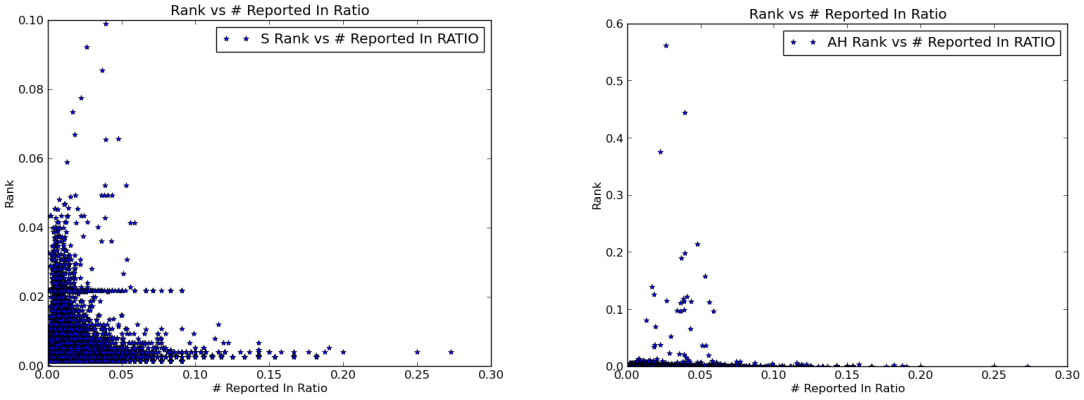



Figure 4. Report Network Rank vs # Reported In Ratio

Again, there does not seem to be any clear correlation. However, notably Pagerank (at 0.00032) and AverageHub (at 0.02) have a clear cutoff for rank (HITS and SALSA do too to a lesser extent). Users with rank above this cutoff are outliers and were proposed to be toxic. The users in this group were reported many times. The least reported user for AverageHub was still reported 8 times, and those who reported that user were generally successful at identifying other highly reported users. The user with the highest reported ratio below this cutoff actually only had 3 reports (out of 11 conversations). All 3 reports came from toxic users (who got reported a lot). When we manually investigated these highly ranked users, we determined them to be spam bots and a dirty user. Thus, setting a rank threshold makes sense for determining which users to remove from the system.

AverageHub's rank provides an intuitive and fair way to judge reported users. To rank highly in AverageHub, a user must be reported by many users that rarely report anyone else (or on average, are very good at identifying bad users). In other words, a user is so offensive that he/she motivates users, which generally do not report people, to report him/her.

We expected that Pagerank would not perform well because it is counterintuitive to assign more hubbiness/report weight to users who have high rank. Thus, we attempted to model Inverse Pagerank. In this scenario, having lower rank was better. Reports were treated as removed edges from a fully-connected endorsement graph. However, because the graph was too dense, the results indicated that rank was linearly correlated with degree. While sparsity could be recovered by removing all the bidirectional edges, it would not make sense to do so. The remaining graph would have edges going from reportee to reporter. This gives rank the wrong meaning. Every time a reporter reports, their rank rises (although it should not change), and the reportee's rank would not change (when it should actually decrease).

As Figure 5 demonstrates below, link-based ranking algorithms are insufficient to determine all good and bad users in the Chatous network. First, many users are not reported, which is clear because they received the baseline pagerank. Further, the green (clean) and red (dirty) nodes receive similar rank when they are reported, making it difficult to distinguish between them with a simple rank threshold. We expect that other Chatous resources like chat content and user profile information would be necessary to detect such malicious users.

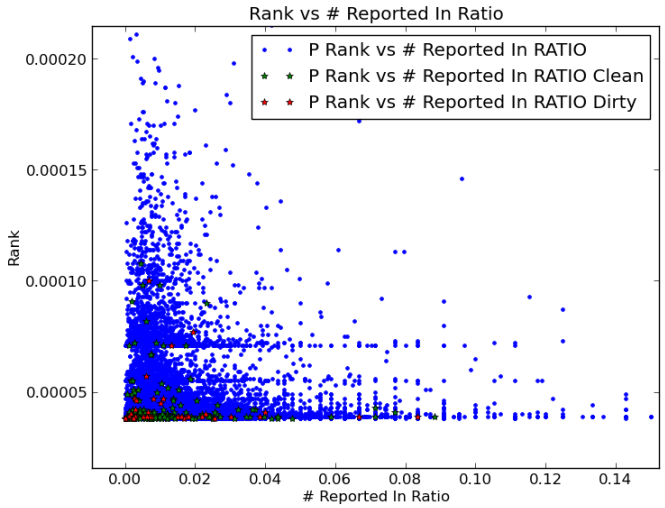



Figure 5. Pagerank vs # Reported In Ratio for all users in the report network. Manual user evaluation was performed to determine the clean users (green) and dirty users (red).

## 6. Conclusion

We found that it is not easy to reliably predict whether a given conversation is positive or negative with only the word-ids, given the frequent misspellings. If we had access to the actual words we might have tried sentiment analysis methods and could also define similarity between words that are misspelled/synonymous.

Using the noisy baseline of conversation length, our triad prediction method predicts friendships with a precision of 84%, using conversation length to sign the edges and degree polarity method for the feature set, so at least 8 out of 10 users matched with this algorithm are likely to enjoy chatting.

For reports, while there is no strong correlation between rank and average conversation length, we can safely ignore reports for users whose average conversation length exceeds a certain threshold because their ranks are always low. When investigating outlier nodes with extremely high rank, Pagerank and AverageHub seem to have obvious threshold cutoffs above which reported users are clearly toxic to the system. Indeed, upon further investigation, all such users were found to be malicious. Unfortunately, it is not possible to identify all dirty users with rank alone. We conclude that rank is a useful feature for identifying bad users but that it should be taken in combination with other data.

## References


1. Turney, P. D. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews <http://acl.ldc.upenn.edu/P/P02/P02-1053.pdf>
2. Leskovec, J., Huttenlocher D., Kleinberg J. Predicting Positive and Negative Edge Links in Online Social Networks <http://snap.stanford.edu/class/cs224w-readings/leskovec10positivenegative.pdf>
3. Najork, M. Comparing the effectiveness of HITS and SALSA <http://snap.stanford.edu/class/cs224w-readings/najork05salsa.pdf>
4. Borodin A., Roberts G. O., Rosenthal J. S., Tsaparas P. Link Analysis Ranking Algorithms, Theory, and Experiments <http://cs.wellesley.edu/~cs315/Papers/Borodin_LinkAnalysisRanking.pdf>
5. Backstrom L., Leskovec J. Supervised Random Walks: Predicting and Recommending Links in Social Networks <http://snap.stanford.edu/class/cs224w-readings/backstrom11randomwalk.pdf>
6. Guo K., Bhakta P., Narayen S., Loke Z. K. Predicting Human Compatibility in Online Chat Networks
7. Chatous - Random Chat <https://chatous.com>