CS 224W Final Project: Transfer of Status Between Sub-Communities in Social Media

Group 5: Catherine Chen, Rohan Puttagunta, and Nick Wu

December 10, 2013

Introduction

The use of the Internet around the world has skyrocketed in recent years, and online forums have developed to accommodate interests, both broad and specialized, as people have begun to seek advice and knowledge from the vast resource of Internet users rather than counting on the relatively small number of people they personally know. Question-and-answer (Q&A) platforms such as Yahoo! Answers and Quora allow users to ask questions regarding any topic, while Stack Overflow is dedicated to the specific field of computer science.

These websites are based on two simple ideas:
(1) If I have a question, someone else in the world probably knows the answer,
and
(2) I am probably not the only person with this question, so if I receive a good answer, I should allow other people to access that answer, as well.

Unlike the more general Q&A sites that include many questions asking about personal opinions, Stack Overflow is designed for factual questions and answers, so users responses are assumed to be objective and verifiable. However, there is no administrator or other authority that endorses answers that are correct: all judgments are made by the community at large.

One potential issue with completely open forums like Stack Overflow is that anyone can answer a question, so there is no guarantee about the quality of any given response. In order to distinguish among different answers, most Q&A websites implement some form of voting or points system through which people can recognize the utility of a given answer and reward users who provided good advice. Stack Overflow allows users to give a post an "upvote" or a "downvote" depending on how helpful they find it to be. In terms of questions, the upvote is generally used to indicate that someone thinks the question is insightful, while the downvote usually means that the question is either a repeat of an earlier post or somehow ambiguous. For responses to questions, upvotes are used to express agreement, while the downvote indicates that something about the response was not satisfactory. The original asker of the question can accept one answer as "correct," which will be the first answer to show up whenever someone views the page for that question; all other answers are sorted in descending order by upvotes. In

addition to quantifying the helpfulness of individual answers to questions through votes, Stack Overflow awards "reputation points" and "badges" to users based on their accepted answers and the number of upvotes they've accrued. Since this value system is entirely determined by interactions among users, it presents interesting questions about the network.

This paper investigates the concept of status on Stack Overflow as it relates to the subcommunities defined by "tags," which are the topics relevant to a particular question, as decided by the user who asks it. The first aim of this study is to determine the existence of correlation between a users status within a subcommunity and the positive or negative nature of the evaluations of his or her answers therein. Naturally, a user with high-status will likely be an experienced contributor who will have garnered many upvotes over time, but we are interested in the perceived quality of each of their answers rather than the total volume of approval. Additionally, we want to note any changes in this perception of quality as the user's status increases, in order to find out if the responses posted by a user with high status in a given topic are considered better, on average. The next question that we pose is: can a user's high status in one subcommunity transfer into other subcommunities? If so, can the extent of status transfer between the subcommunities defined by two tags be used as a reliable measure of the similarity between the two topics?

Prior Work

The basis of this study comes from the work done by Anderson, Huttenlocher, Kleinberg, and Leskovec in their 2012 paper "Discovering Value from Community Activity on Focused Question Answering Sites: A Case Study of Stack Overflow," in which the authors investigate multiple aspects of the dynamics of knowledge creation through the Q&A platform. The paper found that questions with large numbers of answers, many highly voted answers, a particularly long most-upvoted answer, and the reputation of the asker were the most essential features for predicting long-lasting value of a question. In terms of user reputation, Anderson et al. find a direct relationship between the speed of question-answering and the level of reputation, leading to a reputation pyramid that clearly separates high-reputation users from low-reputation ones. Additionally, they find that most high-reputation users gained that status from answering questions rather than from asking them, and that the elite users with the highest reputation have more gain from having their questions accepted as correct than users at other reputation levels. Although the authors concede that this may largely be attributable to an idiosyncrasy of the reputation point system, it does still suggest a connection between high-quality responses and the status of the responder. While acceptance by the question asker is not an issue that we will be addressing directly, we are trying to find a similar link between status and positively-received answers, using status as the predictor.

Further inspiration was drawn from "Effects of User Similarity in Social Media," another 2012 paper from the same aforementioned four authors, in which they find that user similarity has a strong positive effect on positive responses in social media environments and that user status has a strong positive effect in the absence of similarity. On a website like Stack Overflow, user similarity can easily be defined by the tags under which they are most active, but if we are comparing status and upvotes within a single tag, it is perhaps irrelevant to consider the users activities in other tags, regardless of how similar their external activities are to those of other users. Furthermore, as the paper notes, so-called "tag-similarity"

is less indicative of positive responses than user-similarity, which is defined by evaluative relationships with common users. The paper proceeds to focus on user-similarity over tag-similarity, and manages to extract considerable structural properties based on this form of similarity, but we believe that tags hold a significant amount of evaluation-related structure as well. We would therefore like to explore the effect of tags on approval more thoroughly.

Method

Our ultimate goal is to evaluate the effect of a user's status in one tag on the score received by that user's posts in another tag. However, we know from prior work that higher status overall generally results in more positive evaluation, so we need to separate the effects of the user's statuses in various different tags on the evaluation of his post. To do so, we first need two things: a metric for a post's score and a baseline expectation for the post's score based on some aspect of the user's status. We will then be able to determine if a certain tag's status explains some deviation from our baseline expectation.

To start off, we need a formal definition of a user's status. A seemingly natural choice would be to use reputation as calculated by Stack Overflow, which is calculated based on upvoted, downvoted, and accepted answers, and a few other site-specific metrics. Such a choice would be relatively uninteresting, though, because if our sense of status is defined by upvotes, finding a relationship between status and upvotes seems meaningless. Instead, we define status simply to be a raw count of a user's activity - answers and questions posted; this definition was successfully utilized by Anderson et al. in both papers cited above. Extending the definition slightly, the status within a tag shall simply be a count of posts within the tag. As Stack Overflow answers are supposedly judged objectively, there is no inherent reason for more experienced users answers to be consistently more highly upvoted. The existence of such a relationship would suggest a more interesting underlying mechanism - perhaps knowledgeable users are more incentivized to stick around and accrue status; perhaps status biases voters, or perhaps users experience on Stack Overflow is educative.

Our data collection process initially involved downloading all Stack Exchange data from September 2011. We used an XML parser to extract all post data made on Stack Exchange and then stripped out the ID of the owner of each post, along with the number of upvotes received as a result of that post, the number of views the post's question recieved, and the associated tags. After obtaining these raw data, we aggregated the number of posts the user made on each tag as well as the number of upvotes received, for all valid pairs of (user, tag).

As a first analysis, we wanted to evaluate the relationship between status and upvotes within a tag to establish the baseline expectation for upvotes based on status. We decided to see how the number of upvotes changed as a user continued to post content for a given tag over time. We focused our analysis on the five largest tags on the Stack Overflow website by volume: C#, C++, Java, JavaScript, and PHP. After sorting all posts by creation time, we made a single pass through all the data, tracking pairs of (number of posts within a tag, number of upvotes within the tag). In other words, at the time of a given post x, if the author had p existing posts within the posts tag, and the post x earned u upvotes, we stored the pair (p, u). Finally, for each distinct post count (each of which corresponds to a status

level), we found the average number of upvotes.

We found that plotting the average number of upvotes versus status level in a tag was extremely noisy. We realized that the number of upvotes was too affected by the popularity of the question itself, rather than the quality of answers. Rather than measuring the absolute magnitude of the upvotes, we realized that we actually wanted a measure representing the likelihood of a post getting an upvote independent of the question's popularity. As such, we decided to normalize our upvotes by the number of views the page received. Though the plots looked cleaner, we had a new problem - a large number of posts had near 0 ratings and were dragging down the average. This makes sense as the more popular questions generally have a multitude of answers, most of which are buried and ignored. In order to minimize the influence of these posts, we decided to use the median normalized upvote count instead of the mean. With this change, the plots showed a very clear and consistent positive correlation between status within a tag and the expected score of the user's post.

After performing the analysis described above, we analyzed the effect of status on upvotes between pairs of the most popular Stack Overflow tags. Given two tags A and B, we wanted to investigate whether a high status in A leads to higher scores in B. For this analysis, we used the the first analysis as an *a priori* predictor for the posts' score in B, based off of the user's status in B. We then subtracted this from the actual normalized upvote score to get the error in our predictor. We plotted this error term against the user's status in A, once again taking the median at each status level. Under this measure, a positive correlation would suggest the existence of a cross-tag reputation premium that gives a user with high status in A a score boost even when posting in B.

Results and Findings

Our first analysis explored the possibility of a relationship between status and approval within a tag by studying the number of upvotes on a user's post and the user's status at the time of the post under the five largest tags in terms of number of posts: C#, Java, PHP, JavaScript, and C++. As expected, the raw data were noisy, due in large part to the many users who have posted a small number of times, for whom there is greater variance in number of upvotes per post. We tried to reduce the noise by averaging the number of upvotes for each distinct status value, but there was still too much noise. The results of this initial analysis appear in Figure 1:

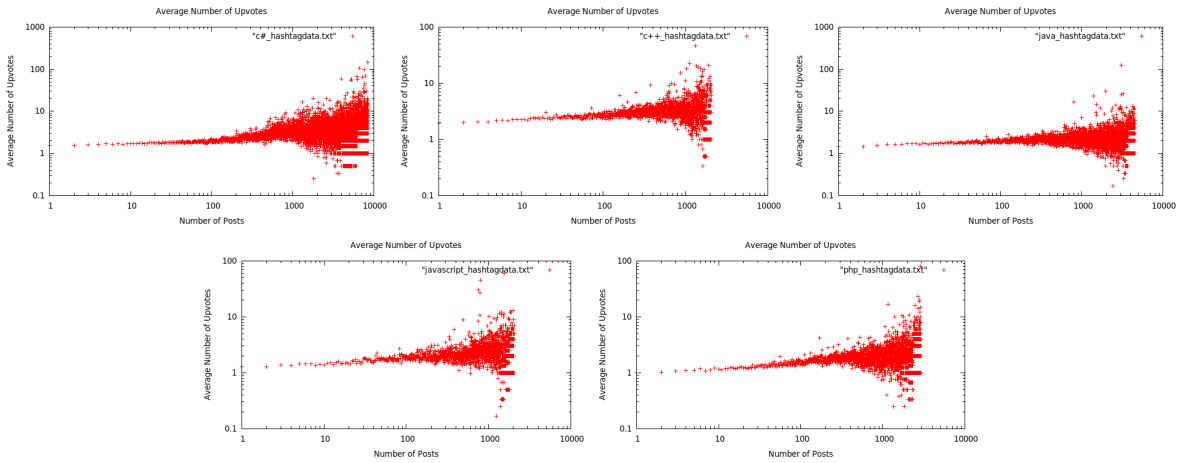

Figure 1: Average number of upvotes versus author's existing status (number of posts) within the tag, on a log − log scale. From left to right and top to bottom, the graphs correspond to the tags C#, C++, Java, JavaScript, and PHP.

As Figure 1 demonstrates, there appears to be a positive correlation between the average number of upvotes and the author's status within a given tag. However, the trends are still noisy, and we attribute a portion of the noise to the fact that the number of upvotes is heavily influenced by the popularity of the question. Thus, we normalized the upvotes for each post to account for pageviews of the question that the post was answering. After performing this normalization, we were able to see slightly cleaner upward trends, as illustrated in Figure 2:

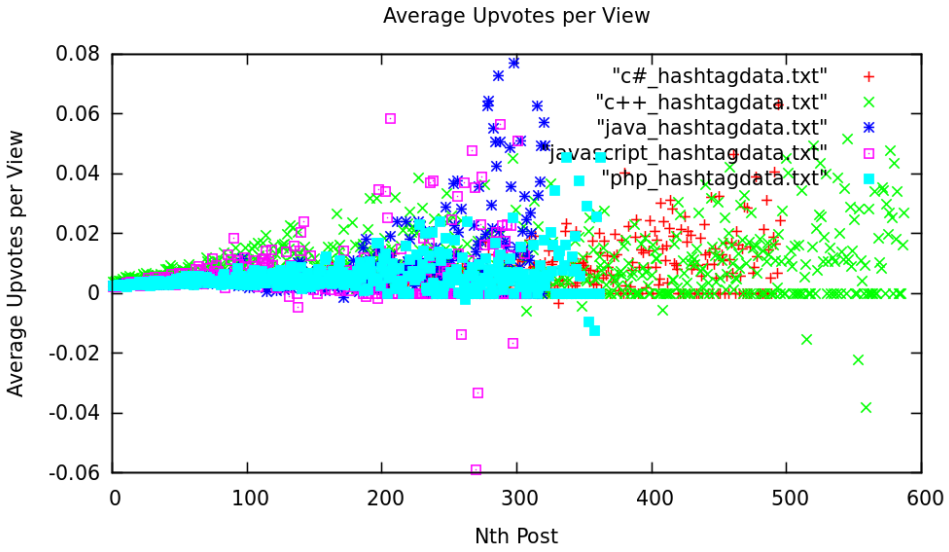

Figure 2: Average number of upvotes versus number of the author's existing posts within the tag, normalized for pageviews of the relevant question

Though we saw improvement after normalizing the upvotes, the graphs were still unsatisfactory. Next, we performed the analysis of normalized upvotes versus status again, but with the median instead of the mean. The median was chosen as a measure for post popularity that was more robust to outliers. Using the median instead of the mean to combine the scoring for each status level created the much clearer and more consistent lines seen in Figure 3. Note that each language tends to be very jumpy at higher status values; though we still see improvement within each section, there are disconnects between sections. The disconnects occur because of the small number of users at those high status levels - when the number of users decreases by one, a disconnect occurs.

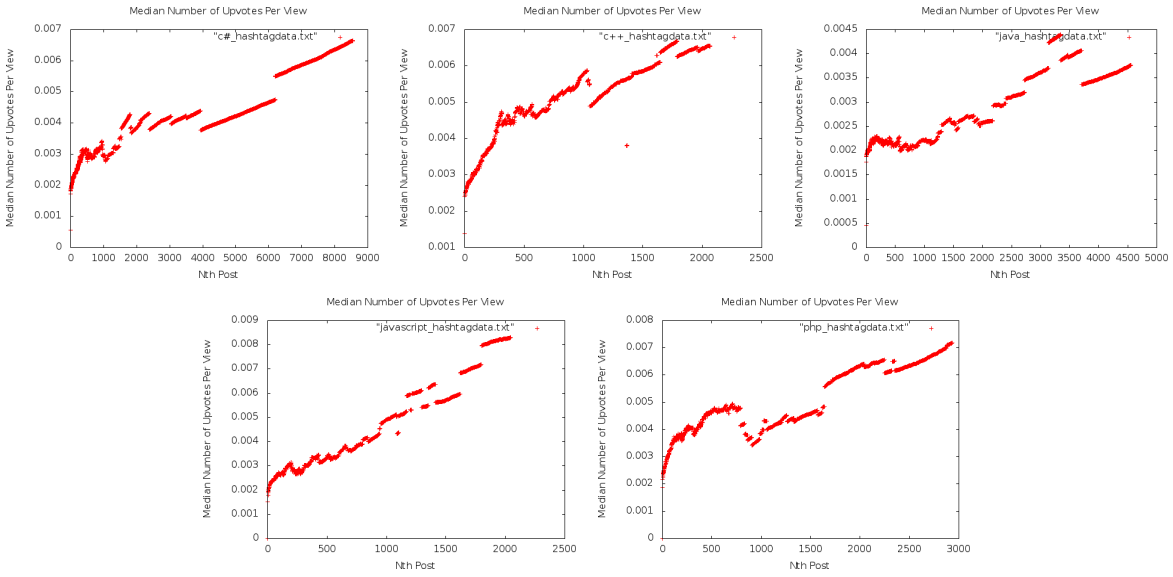

Figure 3: Plots for each of the five most popular tags of the number of the author's existing posts (i.e. the author's status) within the tag versus the median number of upvotes garnered.

Finally, we turned to the cross-tag analysis. Since we focused our investigation on the five tags C#, C++, Java, JavaScript, and PHP, there were twenty possible pairwise causal relationships. The plots below in Figure 4 illustrate the relationship between status in one subcommunity and upvotes in the other for each pair.

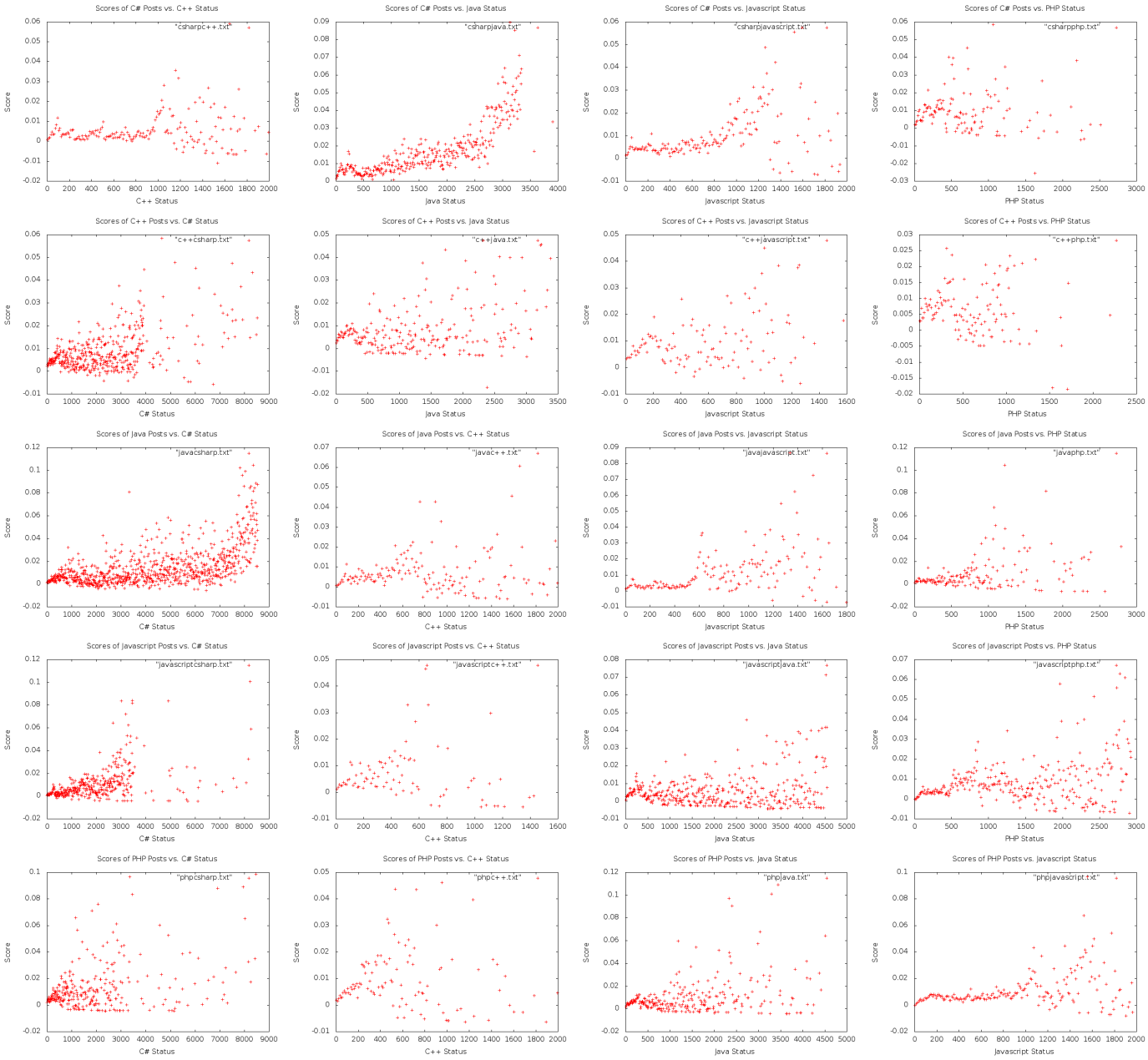

Figure 4: Plots for each of the five most popular tags of the number of the author's existing posts (i.e. the author's status) within the tag versus the median number of upvotes garnered.

Interestingly, most of these plots are relatively uninteresting. They start out around zero, but as the status increases, they get noisier as the number of relevant posts decreases. Strangely, a few of the graphs, such as the two relating C++ and Java, contain a noticeable presence of sub-zero score posts, suggesting that higher statuses in one tag should cause us to be more wary of posts in the other. This is especially strange given the relatively close nature of Java and C++; we actually expected those two tags to be the most related. There are, however, two relationships that seem to to be particularly

notable: Java and C#, as seen in the second graph in the first row and the first one in the third row of Figure 4, and JavaScript and C#, as seen in the third graph in the first row and the first one in the fourth row.

The Java and C# relationship is characterized by a somewhat weak positive correlation to start, with a strong and noticeable uptick at the highest status levels. Upon further investigation, we realized that this uptick can be largely attributed to one particular power C#/Java user. In fact, the highest points in the Java graph correspond to this user; the subsequent drop in score is a result of reaching his final status level. Looking at the user's graph of post scores versus his overall status in Figure 5 below, the user has gotten markedly better scores for his posts at his highest status levels, and this increase singlehandedly explains the Java/C# relationship.

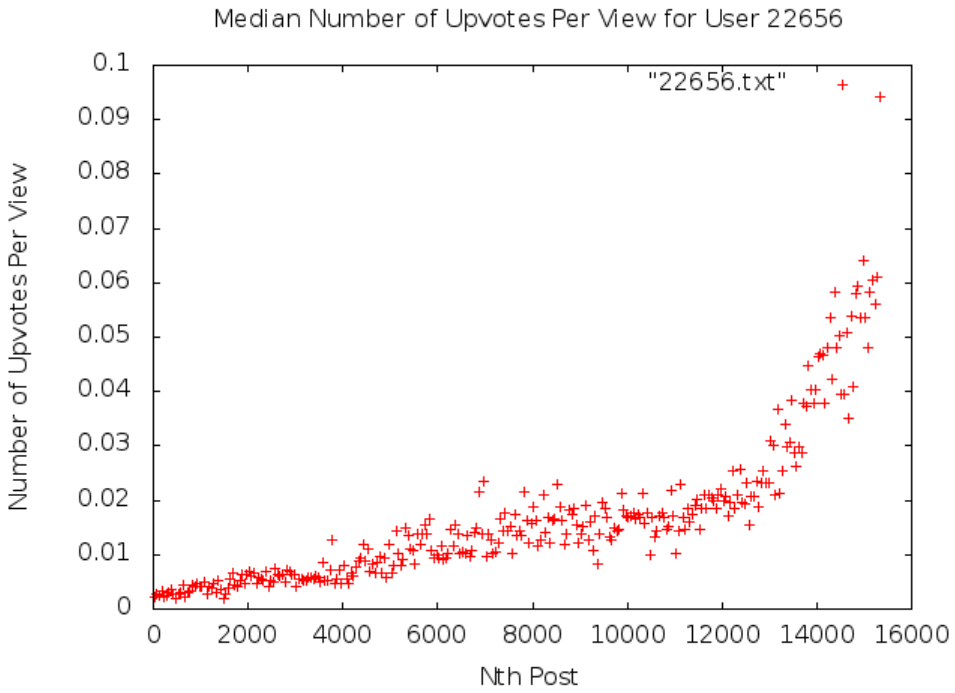

Figure 5: Upvotes versus status for the most prolific Java and C# user.

Similarly, the interesting aspect of the JavaScript and C# relationship falls largely on a single user. Both graphs in this relationship see a large uptick in scores at moderate status levels before degenerating into uninteresting noise. The status levels at which these relationships degenerate actually occur at the final status levels of one user. If we look at this user's graph of post scores versus overall status below in Figure 6, this user has also gotten markedly better scores for his posts at his highest status levels. Once again, the single user singlehandedly explains the JavaScript/C# relationship.

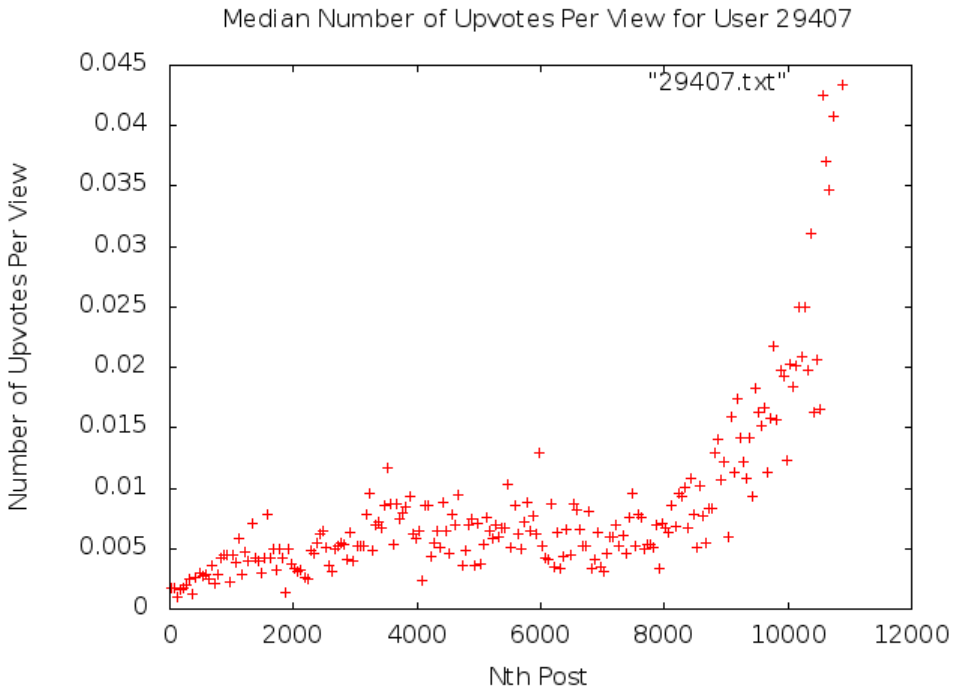

Figure 6: Upvotes versus status for the most prolific JavaScript and C# user.

Conclusion/Further Work

We have found that there little to no cross-tag effects of status on evaluation. That is, a higher status in one tag A does not serve as a predictor of more positive evaluation in a different tag B beyond the predictive power of the user's status in tag B. Indeed, the only interesting positive correlations were caused by individual power users and were not indicative of any larger trends. Fascinatingly, our work suggests that the previously found conclusion that a user's status affects their evaluation is largely explained by only observing activity within a single tag. That is, the evaluation of a post can be predicted reasonably well only by looking at the user's history in that post's tag; the rest of the Stack Overflow activity is largely irrelevant. It would be interesting to pursue this implication further, as a more focused investigation could be fruitful.

References

A. Anderson, D. Huttenlocher, J. Kleinberg, J. Leskovec. "Discovering Value from Community Activity on Focused Question Answering Sites: A Case Study of Stack Overflow." *In Proc. KDD, 2012.*

A. Anderson, D. Huttenlocher, J. Kleinberg, J. Leskovec. "Effects of User Similarity in Social Media." *ACM International Conference on Web Search and Data Mining (WSDM)*