# Predicting Programming Community Popularity on StackOverflow from Initial Affiliation Networks

CS224W, November 14, 2013. (Group 18)

Sophia Westwood (sophiasw) Melissa Johnson (melj) Brie Bunge (bbunge)

## 1. INTRODUCTION

StackOverflow has become a popular question and answer site for programmers since its launch in 2008. As new programming frameworks and languages emerge, StackOverflow communities form around the new tags to ask and answer questions. Our analysis investigated both the affiliation network between tags across the lifetime of StackOverflow, as well as the relationship between the initial affiliation networks for these programming tags during the first 4 weeks of their existence and how these initial structures predict future tag popularity in November 2013.

Thus far, much of the literature on groups within social networks has focused on static properties such as community detection and clustering, rather than predicting properties over time. Even recent work in this area primarily considers the relations between group members and how these properties relate to the group's evolution. For example, Backstrom et al found that groups in social graphs with a large number of triangles grow much less quickly than groups with a small number [4] while Kairam et al discovered that the class of growth a group experiences can predict its longevity [11].

Meanwhile, Zheleva et al. and Geard et al. studied the relationship over time between affiliation networks and their underlying social networks, where the affiliation networks form via links between users and subgroups while the social networks involve direct links between users. In general, however, research on the evolution of affiliation networks themselves – particularly when they exist independently of an underlying social network – has been sparse.

We considered the affiliation network over time between tags, looking at the relationships between groups rather than group members. Because StackOverflow lacks formalized "friendships" between users, the affiliation network forms without the influence of an explicit social network. StackOverflow has over 2 million users and 6 million questions as of November 2013 [2], and past work shows that roughly three-fourths of them have contributed to at least one post [12]. Yet, unlike traditional social networks where groups form directly from lists of members, StackOverflow communities arise from the tags that programmers use to label questions. These tags center around specific programming languages, frameworks, and platforms. We leveraged this tag data both to form links in the affiliation network via co-occurring tags, as well as to form links based on user participation in posts across different tags.

We primarily modeled these tag communities as a modified folded bipartite affiliation network, as further discussed in Section 3. Communities represented by tags form the nodes, while the number of engaged members shared between two StackOverflow tags determines the graph's weighted edges. We analyze this affiliation network for the all-time activity on StackOverflow.

Next, for each of the top 1000 tags, we consider the tag's activity and relationships in the affiliation network from the first 28 days of the tag's life on StackOverflow, measured from when the first post with the tag appears. We classify each tag as "more popular" (top 500 tags in November 2013) or "less popular" (top 500 to 1000 tags) where the rank is based on the cumulative number of questions relative to other tags. We compute features including degree centrality, closeness centrality, average shortest path, clustering coefficient, and page rank based on the affiliation network generated from user activity, as well as a simpler affiliation network from co-occurring tags. We apply Random Forest Classifiers, Linear SVCs, Logistic Regression, and AdaBoost Classifiers to predict tag success, showing the importance of a tag's place in the initial tag affiliation network. In all, our results suggest that the initial affiliation network around a tag is more indicative of later success than metrics on initial activity.

We begin with a survey of prior work surrounding groups and evolution of networks in Section 2, followed by a discussion of our data collection process, network modeling, and features in Section 3. Next we share the analysis and predictive results in Section 4, before concluding and discussing future work in Section 5.

## 2. PRIOR WORK

Most literature that analyzes the static properties of groups or communities within networks has focused on traditional social networks, in which people join the network and "friend" each other to form a relatively stable undirected link. Here, we look at recent work on how these groups evolve over time and relate it to our problem of predicting group success based on early affiliation networks.

In "Group formation in large social networks: membership, growth, and evolution," Backstrom et al. [4] predicted which users join communities based on the structure of the community and the pre-existing links within it. The researchers found that groups with a large number of triangles grow significantly slower than groups with a small number of triangles. In addition, the paper drew upon data on papers published at conferences to analyze movements of people between communities in time-slices. The project is similar to our work in that it evaluated a network over time while measuring growth, though it analyzed the internal structure of a

group rather than the external connections between groups as we will.

Next, in "Co-evolution of social and affiliation networks", Zheleva, Sharara and Getoor [16] analyzed how social networks and their affiliation networks evolve and affect each other's evolution. This paper developed a generative model for this co-evolution, discovering various properties of social networks and their affiliation groups. They showed that affiliation group sizes follow a power-law distribution with a heavy tail. The paper concluded that members often join groups for a collection of reasons beyond simple friendships. We attempted to capture these underlying reasons by analyzing the affiliation network structure, for example by taking clustering coefficient and page rank as features. Finally, as discussed in Section 4.1.1, power law distributions unsurprisingly also dominated in our dataset.

Very few papers have considered the independent evolution of the affiliation network itself – each of the papers so far primarily explored individual groups or a relationship between the affiliation network and a social network. It remains an open problem to analyze the properties of affiliation groups themselves, especially independent of their underlying social network. Very recently, Ghosh et al. in "Understanding evolution of inter-group relationships using bipartite networks" [10] created a generative model to predict the evolution of an affiliate network. This paper used a bipartite graph with the two subsets $U$ and $V$ representing the affiliation groups and members. Similar to our work, this paper evaluated the strength of the connection between groups by estimating the number of common users and thresholding edges, providing precedent for our formation of edges discussed in Section 3.

In addition, past research on StackOverflow has particular relevance to our research problem given the domain, although many of these papers focus on textual analysis and individual user prediction rather than graph structure. For example, in "Evolution of experts in question answering communities" [14], Pal, Chang, and Konstan presented a temporal study of the behavioral patterns of experts on StackOverflow. The authors separated activity data into biweekly time buckets to construct a temporal series of activity over 26 time periods for each user. A Gaussian mixture model clustering algorithm combined with the Bayesian Information Criteria resulted in six clusters of equal numbers of experts, exposing three dominant patterns overall. The paper labeled these categories as experts with high initial activity that falls off, low initial activity that ramps up, and consistent activity. While their paper, like ours, predicted activity over time, we focused on how tag communities fit into the larger affiliation network structure, rather than emphasizing the specific temporal activity of individuals.

Next, Anderson et al. in "Discovering value from community activity on focused question answering sites: A case study of StackOverflow" [3] analyzed the development of an accepted answer to a StackOverflow question, considering factors such as the arrival time of each answer, the voters, and the reputation of each answerer. The researchers developed a bipartite graph of questions and answerers and created a randomized baseline pattern of coanswering. The paper examined temporal effects as features on a small scale, using data from an hour after a question is posted to predict page views a year later. The researchers used a logistic regression classifier and performed 10-fold cross validation, finding that the highest importance coefficient associates with the number of answers received. The paper's success in using early information from an answer's lifetime to predict behavior much later on provides proof of concept for our machine learning efforts to predict later community behavior based on early affiliation networks.

# 3. DATA COLLECTION AND MODELS

Now we discuss the key elements of our data collection and data modeling process to transform the raw StackOverflow data into affiliation networks and features for analysis and classification. First, we describe the raw data in Section 3.1. Then, Section 3.2 details the models and features for the analysis in Section 4.

## 3.1 Data collection

Our raw dataset was the StackOverflow anonymized data dump of questions and answers from Aug 2008 to Sept 2013, including posts, comments, tags, and user badges [9]. This particular data set spans a sufficiently broad and recent date range to allow us to track the evolution and emergence of communities over time. The most popular tags (Javascript, Algorithms, Ruby, and so on) indicate particularly active communities. We focused on understanding these tags and the affiliation network between them based on co-occurring tags and user participation in questions with multiple tags.

```
c#, java, javascript, php, android, jquery, c++, python,
html, mysql, asp.net, ios, iphone, css, .net, sql,
objective-c, ruby-on-rails, c, ruby, sql-server, ajax,
xml, wpf, regex, asp.net-mvc, arrays, database, linux,
django, json, xcode, vb.net, windows, eclipse, facebook,
ruby-on-rails-3, string, multithreading, html5, win-
forms, r, wordpress, image, visual-studio-2010, forms,
performance, osx, asp.net-mvc-3, spring, algorithm,
oracle, linq, swing, git, excel, wcf, web-services,
perl, node.js, sql-server-2008, visual-studio, apache,
entity-framework, actionscript-3, hibernate, ipad, bash,
cocoa-touch, cocoa, flash, api, qt, silverlight, file,
jquery-ui, sqlite, list, matlab, .htaccess, tsql, del-
phi, codeigniter, security, function, class, internet-
explorer, mongodb, google-app-engine, oop, flex, jsp,
css3, google-maps, unit-testing, postgresql, validation,
shell, sockets, parsing
```

**Figure 1: 100 Most Popular StackOverflow tags (Nov 2013)**

## 3.2 Data models and features

First, Section 3.2.1 describes our model for the tag affiliation network across the lifetime of the site, which we proceed to analyze in Section 4.1. The following subsections explain feature generation for Section 4.2's machine learning analysis and results. In detail, Section 3.2.2 explains basic features from initial tag activity that we use in Section 4.2.1; Section 3.2.3 considers features used in Sections 4.2.2 and 4.2.3; Section 3.2.4 and Section 3.2.5 explain our generation of two types of weighted affiliation networks; and Section 3.2.6 explains the features we generated from these affiliation networks for use in Sections 4.2.4 and 4.2.5.

### *3.2.1 All-time tag community*

We transformed the raw dataset into a modified folded bipartite network as follows to capture the all-time tag affiliation network. First, we scraped the 100 most popular

tags (as of November 2013) from `http://stackoverflow.com/tags` [15] (see Figure 1 for full list). Then, we associated user activity with tags by mapping user IDs to number of actions for each tag. For a given tag $T$, a user received a point for every post they created with tag $T$. In addition, a user received a point for every answer or comment on a post with tag $T$. Note that a question may have multiple tags.

Next, we defined an activity threshold $H$ where a user needed to have at least $H$ points in order to belong to a tag community $T$. We chose activity threshold $H = 10$ (see Section 4 for justification).

To form the network, each tag $T$ became a node. For every pair of tags, we counted the number of users that belong to both tag communities – if this number exceeded a certain overlap threshold $D$, we drew an undirected edge between the tags. We also experimented with weighted edges, where the weight of the edge corresponded to the strength of the overlap. As discussed in Section 4, we chose overlap threshold $D = 100$.

#### 3.2.2 Tag's activity in first 28 days

We next modified this procedure to predict future tag popularity based on the first 28 days of activity. To do so, we scraped the 1000 most popular tags (as of November 2013) from `http://stackoverflow.com/tags` [15]. Tags ranking in the top 500 were classified as "more popular," while tags ranking from 501 to 1000 were classified as "less popular."

For each tag $T$, we found Day 0 as the day of the first post tagged with $T$ in the lifetime of the site. Then, the first bucket $B$ contains only the first 28 days from Day 0. All activity described as "initial tag activity" for the remainder of the paper refers to activity on the StackOverflow graph during these first 28 days for the tag.

As a starting point, we computed basic features of this initial tag activity, looking only at the tag itself. Note that this approach follows the emphasis on individual groups described in Section 2 Prior Work, and ignores the tag's position within the larger affiliation network of tags.

1. Number of posts tagged with $T$.
2. Number of questions tagged with $T$.
3. Number of comments for items tagged with $T$.
4. Number of unique users participating in items tagged with $T$.

These features capture the hypothesis that a tag's final level of activity is evident in its initial level of activity.

#### 3.2.3 Tag's initial relationship to other tags

Next, we computed indicators of $T$'s relationship to other tags, transitioning towards a model that includes $T$'s place in the affiliation network. We began with four simple metrics:

1. Number of posts tagged with $T$ with multiple tags.
2. Percentage of posts tagged with $T$ with multiple tags.
3. Number of posts tagged with $T$ also tagged with one of the top ten tags (by # of posts during the 28 days).
4. Percentage of posts tagged with $T$ also tagged with one of the top ten tags (by # of posts during the 28 days).

Here, items 1 and 2 indicate the amount of overlap that $T$ has with all other tags on the site, while items 3 and 4 consider which tags $T$ overlaps with, and only count the ten most popular tags. The ten most popular tags are defined as the ten tags during $T$'s initial 28-day period that appeared in the most posts, and may be different for different time periods.

#### 3.2.4 Tag's co-occurring tag affiliation network

To further investigate a tag $T$'s position in the initial affiliation network, we created a tag affiliation network for the first 28 days of each tag based on co-occurring tags. In this network, each node is a tag community. An undirected edge between two tag nodes $T_i$ and $T_j$ has weight $W$, where $W$ is the number of posts in which $T_i$ and $T_j$ both occur. So, pairs of tag communities with more posts tagged with both tags will have stronger connections in the graph. In these affiliation networks, we only considered activity within the first 28 days of tag $T$ when forming edges. We form 1000 of these networks, one for the first 28 days of activity for each tag.

#### 3.2.5 Tag's user-activity tag affiliation network

Similar to our all-time model in Section 3.2.1, we also created 28-day affiliation networks based on user activity between tags. In this network, each node is a tag community as in Section 3.2.4 above. Edges, however, form from user activity that links the two tag nodes. Specifically, an undirected edge between two tag nodes $T_i$ and $T_j$ has weight $W$, where $W$ is the number of users that have made at least five contributions to both $T_i$ and $T_j$ (see Section 4 for discussion of the cutoff; we chose 5 rather than 10 here due to the shorter timespan of activity). Again, we form 1000 of these affiliation networks, one for the activity during the first 28 days of each tag.

#### 3.2.6 Affiliation network features

For each tag, we calculated features based on the two types of affiliation network described above in Sections 3.2.4 and 3.2.5. These features capture the importance of each tag at the time that it first appeared on StackOverflow, and its relationship to other tags, with the goal of predicting community success from the relationships of the communities to each other:

1. Degree centrality: Basic measure of how important a node is based on how strongly connected it is to other nodes.
2. Average shortest path: Measure of how close a node is to all the other nodes in a graph
3. Closeness centrality: Measure of closeness, often used when analyzing information flow.
4. PageRank: Standard measure of importance as discussed in Section 4.1.4.
5. Clustering coefficient: Measure of how embedded a tag is in an already closely-connected community.

Because we used weighted edges, we implemented algorithms ourselves to compute each of these features. For features such as degree centrality where the degree of a node plays a central role in the measure, we substituted degree

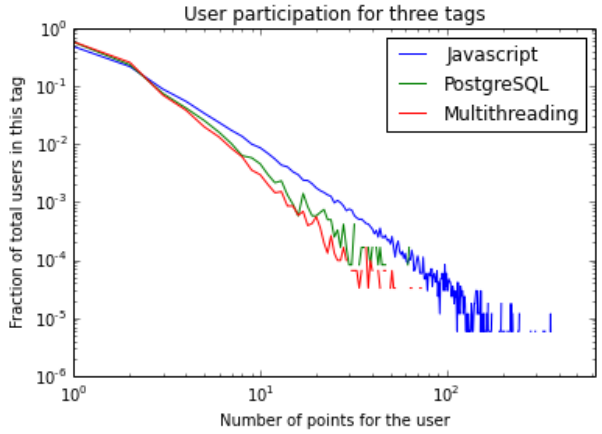


**Figure 2: The user activity for Javascript, PostgreSQL, and multithreading tags. Points correspond to activity in posts and comments with this tag, as described in Section 3.2.1.**

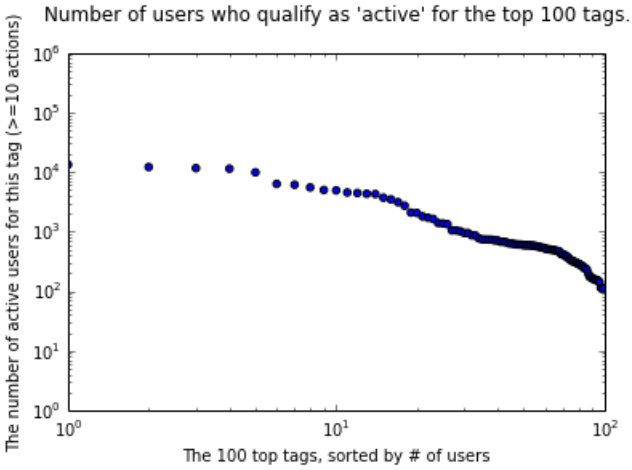


**Figure 3: The number of active users per tag follows a power-law distribution.**

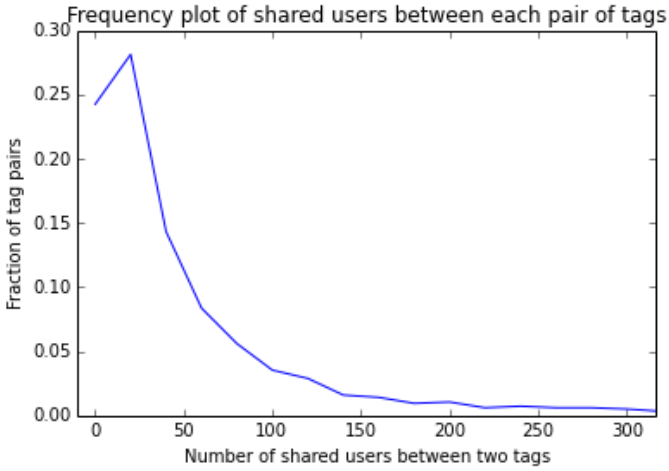


**Figure 4: The number of shared users between two tags follows a power-law distribution.**

with the sum of the node's edge weights, in order to account for some relationships being stronger than others [6]. For closeness centrality and average shortest path, which rely on calculating path lengths, we replaced a measure of the number of hops with a modified version of Dijkstra's algorithm. While Dijkstra's algorithm is meant for graphs in which edges are costs, and thus the shortest path minimizes the edge values traversed, we implemented a version used by Newman [13] which first transforms the weights of each edge into a cost by dividing 1 by the weight, and then running Dijkstra's algorithm, which gives a measure of farness. Inverting the farness then gives us a measure of closeness. To calculate the clustering coefficient, which relies on the number of triangles formed by each node, we represent each triangle as the mean of the two edge weights coming from the node, instead of just counting it as one as we would for an unweighted graph [6].

# 4. ANALYSIS AND RESULTS

We began our analysis in Section 4.1 on the full affiliation network to better understand the overall structure of the graph and the interaction patterns between tag communities. We use these insights in Section 4.2 as we predict community success based on features from the first 28 days of a tag's activity.

## 4.1 All-time network analysis

We start in 4.1.1 by discussing the high-level analysis leading to the chosen thresholds for network formation in the all-time community. Next we provide a simple summary of the full affiliation network in 4.1.2. Section 4.1.3 discusses a community detection algorithm and Section 4.1.4 discusses PageRank to prepare for 4.1.5, where we run community detection and PageRank on the all-time network to further uncover the network structure. This section validates common intuitions about programming communities and leads to Section 4.1.6 where we investigate different underlying types of user behavior that contribute to the links formed in the user-based affiliation networks.

### *4.1.1 Thresholds for power-law structures*

First, we analyzed the generation of all-time affiliation network model from Section 3.2.1 to understand the network structure with regard to tag activity. We started by plotting the activity levels of users in Figure 2 for three tags of varying levels of popularity: Javascript, the third most popular tag by number of questions asked; multithreading, the 38th most popular tag, and postgresql, the 95th most popular tag. For each tag, user engagement follows a power-law distribution, as we would expect for user behavior in a network. The most engaged users score above 100 (using the point scale described in 3.2.1) while most score just 1.

From Figure 2, we chose a user engagement cutoff of 10 to select roughly the most engaged 3% to 7% of users for each tag as "active members" in the community. Note that the informal but popular analysis of user communities in [5] chose this overlap number as 3 to involve the top 20% to 30% of users. We experimented initially with a lower threshold of 3, and then increased it to 5, and then finally increased

it to 10. Each choice resulted in a power law distribution of the number of active users, and we settled on 10 as a heuristic that was both strict enough to require some level of commitment, yet loose enough to let in more than the experts. Figure 3 shows the resulting plot of the raw number of active users for each of the top 100 tags – the linear trend on the log-log plot confirms the power law distribution.

Next we investigated the number of shared users between each pair of tags. Figure 4 buckets the number of shared users and plots the fraction of tag pairs that fall into each bucket. Again, the number of shared users follows a power law distribution, and the plot suggests a sensible cutoff of 100 to 200 shared users for communities to create an edge. We chose the cutoff as 100, which translates to 19% of all tag pairs forming edges. As a result, two nodes have an edge if their tags share 100 users who have contributed 10 posts, answers, or comments in each community.

#### 4.1.2 All-time network summary

After applying the thresholds, the all-time tag affiliation network described in Section 3.2.1 contains 94 nodes, 958 edges, and one strongly connected component. Note that six of the nodes have disappeared because they do not share 100 engaged users with any other tag. The full diameter of the graph is approximately 4 hops, and the clustering coefficient is 80%. Together, these numbers reflect the intuition that if we have two related tags such as JQuery and Javascript, a third tag Ajax will also likely be connected to JQuery if it is connected to Javascript. It also demonstrates that the small-world phenomenon applies to this network of communities.

#### 4.1.3 Background: Community detection

To prepare for the network analysis on the all-time tag affiliation network in Section 4.1.5, we discuss the details of two algorithms: community detection and PageRank.

The community detection algorithm takes in a graph G where each node belongs to some community. We initialize these communities by adding each node to its own new community. Note that getCommunity(node) returns the community that the given node is a part of, while getNodeSwitchedModularity() returns the modularity of the graph formed by switching the node into the given community. We then run the following algorithm on the graph to partition it into communities:

```
function PartitionGraph(G)
    stable ← False
    while stable ≠ True do
        stable ← True
        for node ∈ G do
            community = GetNewCommunity(G, node)
            if community ≠ getCommunity(node) then
                stable ← False
            end if
            switchNodeToCommunity(community, node)
        end for
    end while
end function
```

where GetNewCommunity is defined as follows:

```
function GetNewCommunity(G, n)
    maxModularity ← modularity(G)
    community ← getCommunity(n)
    for neighbor ∈ getNeighbors(n) do
        neighborComm ← getCommunity(neighbor)
        switchMod ← switchMod(G, neighborComm, n)
        if switchMod > maxModularity then
            maxModularity ← modularity(G)
            community ← neighborComm
        end if
    end for
return community
end function
```

where $switchMod()$ returns the modularity of the graph if the node were to be switched into its neighbor node's community.

We next use the generated communities to create a new graph where each node represents a community. Edges are drawn between these nodes with each edge weight given by the sum of the weights of the edges between those nodes in those two communities. We then iteratively run PartitionGraph on this new graph of communities to merge more communities, continuing until the algorithm is stable and we cannot increase the modularity of the graph by switching any one node into another community.

#### 4.1.4 Background: PageRank

The intuition behind PageRank is that there is a web surfer who starts on a random page, then keeps clicking on web links until getting bored and starting on a new random page [8]. The probability that this surfer visits a given page is that page's PageRank. In our problem, the nodes are tags and the edges are users who have contributed to both endpoints. Imagine a StackOverflow tag surfer, who traverses tags by means of having a contributor to the current tag transport the surfer to another tag to which they contribute. PageRank is high for tags that have many users who contribute to other tags.

Note that PageRank was designed for directed graphs, but it can operate on undirected graphs by allowing undirected edges to be traversed in both directions. The directed PageRank formula is given by

$$PR(A) = (1-d) + d(\frac{PR(T1)}{C(T1)} + ... + \frac{PR(Tn)}{C(Tn)})$$

$$where\ C(A)\ is\ the\ out\ degree\ of\ node\ A$$
$$and\ d\ is\ a\ dampening\ factor\ \in [0,1]$$

[8]. The dampening factor represents the probability that the random surfer will request a new random page. Extending this to an undirected graph, we have

$$PR(A) = (1-d) + d(\frac{PR(T1)}{D(T1)} + ... + \frac{PR(Tn)}{D(Tn)})$$

$$where\ D(A)\ is\ the\ degree\ of\ node\ A$$

#### 4.1.5 Clustering and PageRank: all-time network

We next applied the undirected PageRank and community detection algorithms to the all-time tag affiliation network from Section 3.2.1. We modified this network to weight edges linearly by the number of shared users between the endpoints. As discussed previously, only users that made at least 10 contributions to each tag were considered as engaged users, and edges only counted if the two tag endpoints had at least 100 common engaged users. The PageRank algorithm (see Section 4.1.4) considered edge weight and used

$d = 0.85$, as suggested by Brin and Page [8], and stopping criterion $\epsilon = 0.001$. In addition, we ran the community detection algorithm [7] (see Section 4.1.3) to connect the coding communities into larger groups. See Figure 5 for results.

In the Figure 5 visualization, nodes closer to the center represent tags with a higher page rank, while nodes at the edge of the figure have the lowest page ranks. Note that these results roughly align with the importance of the tags as reported on StackOverflow. Meanwhile, the meta-community clusters are colored similarly and grouped physically into each "arm" of the visual.

Figure 6 shows the results of these two algorithms on the graphs, using a higher threshold of 1000. Each color represents a meta-community identified by running our community detection algorithm on the graph. The modularity scores are 0.25 and 0.37 for figures Figure 5 and Figure 6. We can again connect these results with common intuitions on the technologies, where the purple group with Objective-C, iPhone, iOS, and Xcode can be labeled "the Mac development group", the blue group with Javascript, JQuery, PHP, HTML, MySQL, and CSS is "the Web development group", and so on.

Together, these visualizations provide common-sense validation on the notion of tags as meaningful affiliation networks.

### 4.1.6 *User behavior across tags*

We next considered user behavior across tags to better understand the formation of affiliation networks with user-based links, including the networks used in Sections 3.2.1, 3.2.5, 4.1, and 4.2.5. The users' activity distributions quantified how much the user linked tag communities together. We determined this activity distribution as follows. First, we fetched all of the questions, answers, and comments that a user authored and fetched the tags corresponding to each of these contributions. We then aggregated the count of contributions for each tag to acquire the number of contributions made to each tag for each user, which we converted to proportions.

Then, we applied k-means [1] to cluster the activity distributions of 1000 randomly sampled users (among users that have made at least 10 posts to filter out passive users, as was done in by [14]). As shown in Figure 7, some types of users place most of their activity into one tag (such as the red type), while other types of users more evenly distribute their activity (such as the blue and aqua activity classes) while still focusing on a few tags, while other types of users participate in many tags without a strong focus. 43% of users fall into the blue and aqua activity classes. We consider these users to be "linker" users as they are likely to exceed the required number of contributions in their top two tags, creating edges in the affiliation network. In contrast, users who focus most of their activity into one tag act as "pillars" for the tag activity without creating many links, while many "dabblers" may not participate sufficiently in any two tags to count as active under both.

We now move to the prediction problem of understanding tag success based on the affiliation network during a tag's first 28 days.

## 4.2 Results Predicting Future Success

We predicted the success of individual tags based on their initial activity in the first 28 days, using features discussed

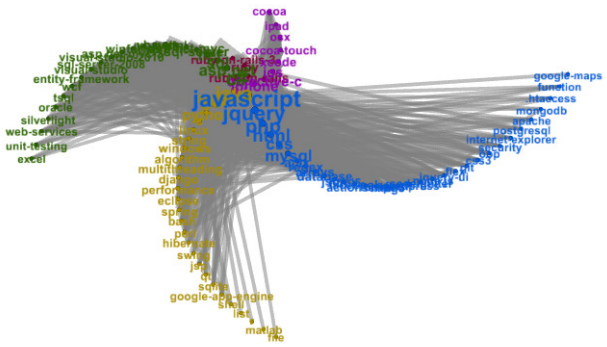


**Figure 5: The clusters identified in the network, together with nodes organized by PageRank. High PageRank nodes are closer to the center. Each colored arm is a cluster.**

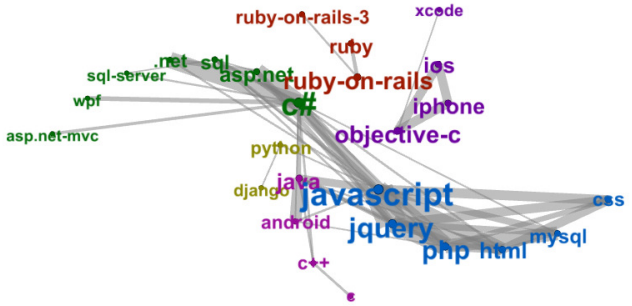


**Figure 6: Sparser version of the clusters identified in the network, together with nodes organized by PageRank. The cutoff is at least 1000 users in common with at least 10 contributions.**

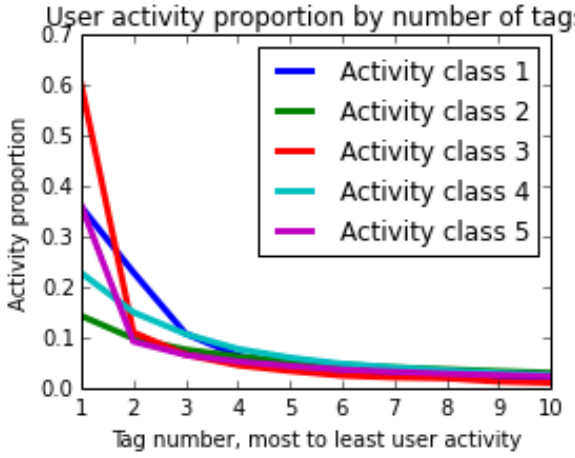


**Figure 7: User types regarding activity distributed across number of tags all-time.**

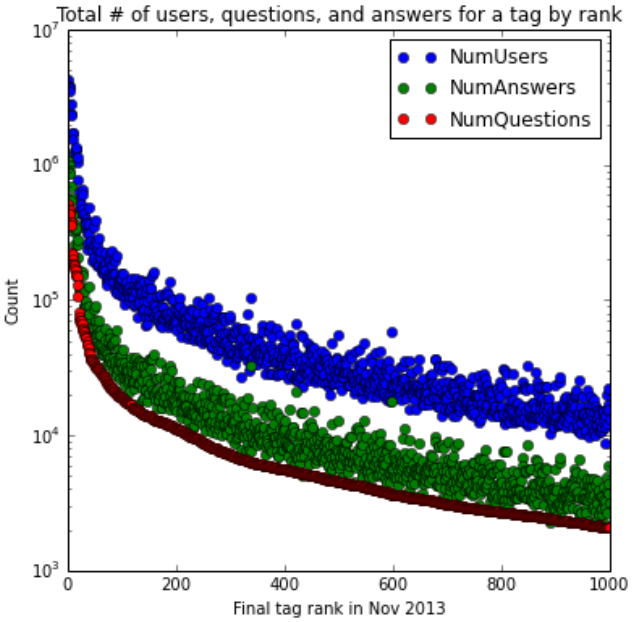


**Figure 8: The total number of users, answers, and questions for each tag given the tag's rank in Nov 2013. Note that the number of questions determines the tag rank, and the total numbers of users and answers are tightly correlated with the rank.**

in Sections 3.2.3, 3.2.4, 3.2.5, and 3.2.6. We split the data set of 1000 tags into two parts for hold-out cross validation, placing 70% of the 1000 samples into the training set and placing the remaining 30% into the testing set. For each tag, we predicted whether the tag would be "more popular" or "less popular" based on the first 28 days of activity for the tag, where popularity is determined by the tag's rank in November 2013. Rank 1 to 500 counts as "more popular," while a rank between 501 and 1000 counts as "less popular."

We applied the Python package scikit-learn together with SNAP, Gephi, and a MySQL database, writing data processing code and weighted graph algorithms to set up a code pipeline for feeding in features and evaluating performance. For each feature set, we applied a Random Forest model, a Linear SVC, Logistic Regression, and an Ada Boost Classifier. We evaluated testing accuracy, training accuracy, test precision, and test recall to measure success.

We also investigated predicting the rank or number of posts directly as a regression problem. For most tags, however, there is only a small amount of data contained in the first 28 days of activity – as seen in Figure 9, most tags have between 10 and 100 posts during this period. So, as detailed in sections below, even the simpler binary classification of more popular versus less popular is challenging. The power law distributions discussed in Section 4.1.1 further complicate the picture. The non-uniform distribution in number of posts makes it difficult to obtain a meaningful number of samples for tags with low post counts, and a prediction that is 300 posts too high has excellent accuracy for very popular tags, but very poor accuracy for less popular tags.

### 4.2.1 *Initial tag activity*

**Table 1: Classifying future popularity only from # Answers, # Questions, # Comments, and # Users during first 28 days of tag.** Columns: test/train accuracy, test precision/recall for Less Popular, test precision/recall for More Popular.

| | test | train | prL | recL | prM | recM |
|---|---|---|---|---|---|---|
| Random Forest | .55 | .93 | .51 | .66 | .61 | .46 |
| Linear SVC | .60 | .60 | .55 | .80 | .72 | .44 |
| Logistic Regression | .58 | .59 | .53 | .80 | .70 | .40 |
| AdaBoost | .62 | .65 | .56 | .77 | .71 | .49 |

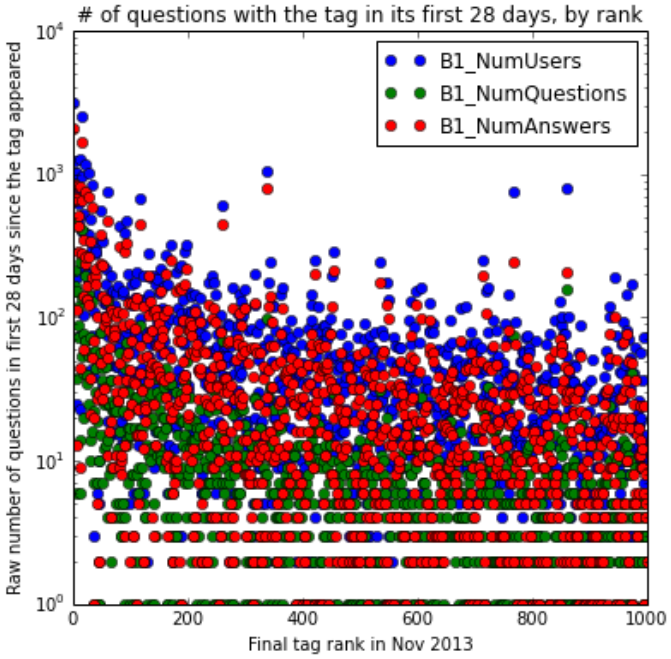


**Figure 9: The number of users, answers, and questions for each tag during the first 28 days after the first question with the tag appears, given the tag's rank in Nov 2013. Note that the activity during the first 28 days is not strongly correlated with the tag rank.**

To begin, we trained using only the four basic features from Section 3.2.2: Number of Comments, Number of Users, Number of Answers, and Number of Questions in the first 28 days of each tag. Training only on these features captured the hypothesis that the raw amount of initial activity is predictive of later activity. As shown in the results in Table 1, this hypothesis does not do very well, with around 57% test accuracy.

Figures 8 and 9 provide insight into why this classification problem is challenging. While the activity over the lifetime of the site with regard to number of users, answers, and questions strongly correlates with tag rank in November 2013, the activity in the first month of a tag is extremely noisy, with no clear correlation patterns to distinguish the more popular tags from the less popular tags. So, rather than focus only on the tag activity itself, we move on to look at the broader tag affiliation network.

### 4.2.2 *Tag co-occurrence counts with all tags*

As shown in Section 4.1, analysis on the tag affiliation

**Table 2: Classifying future popularity only from % and # of tag posts also with another tag during first 28 days of the tag.** Columns: test/train accuracy, test precision/recall for Less Popular, test precision/recall for More Popular.

| | test | train | prL | recL | prM | recM |
|---|---|---|---|---|---|---|
| Random Forest | .59 | .70 | .56 | .42 | .60 | .72 |
| Linear SVC | .58 | .58 | .53 | .82 | .72 | .37 |
| Logistic Regression | .58 | .58 | .53 | .83 | .72 | .37 |
| AdaBoost | .65 | .66 | .67 | .45 | .64 | .82 |

**Table 3: Classifying future popularity only from % and # of tag posts also with a top ten tag during first 28 days of the tag.** Columns: test/train accuracy, test precision/recall for Less Popular, test precision/recall for More Popular.

| | test | train | prL | recL | prM | recM |
|---|---|---|---|---|---|---|
| Random Forest | .65 | .65 | .65 | .50 | .65 | .77 |
| Linear SVC | .63 | .62 | .58 | .66 | .68 | .60 |
| Logistic Regression | .63 | .62 | .58 | .66 | .68 | .60 |
| AdaBoost | .64 | .63 | .75 | .31 | .61 | .91 |

network has the potential to produce meaningful results. So, as a coarse metric of a tag's relationship with other tags, we first considered the feature set consisting only of the number of posts tagged with both $T$ and some other tag during the initial 28 days of activity, and the percent of $T$'s posts where this happens (further detailed in Section 3.2.3).

The results in Table 2 show that learning from just these two features gives us an accuracy around 60%, marginally better than predicting based on the tag activity itself as in Section 4.2.1. The fact that results are similar and slightly better using only these two features shows that a tag's place in the affiliation network can be just as predictive, if not more so, than the activity within the tag itself. Intuitively, this hypothesis is that the success of a tag stems from attracting participation from other tags and gaining publicity via your links with other tags.

### 4.2.3 Tag co-occurrence counts with top tags

Continuing off of the results in Section 4.2.2, we hypothesized that it is even better to have overlap with the most popular tags during a tag's initial 28 days, as these popular tags will produce the most publicity and interest for your tag. Based on this hypothesis, we trained our model on the number of posts that our tag shares with one of the time period's top ten tags, and the percent of our tag's posts that these shared posts make up (see Section 3.2.3 for more details).

**Table 4: Classifying future popularity based on degree center, closeness center, average shortest path, PageRank, and the clustering coefficient of the tag $T$ in the co-occurring tag network (nodes are tags, edges are weighted by the number of posts tagged with both tags during the first 28 days of tag $T$.** Columns: test/train accuracy, test precision/recall for Less Popular, test precision/recall for More Popular.

| | test | train | prL | recL | prM | recM |
|---|---|---|---|---|---|---|
| Random Forest | .63 | .98 | .57 | .70 | .69 | .56 |
| Linear SVC | .65 | .66 | .60 | .70 | .71 | .61 |
| Logistic Regression | .65 | .65 | .60 | .72 | .71 | .60 |
| AdaBoost | .68 | .70 | .69 | .53 | .67 | .80 |

The results in Table 3 show the success of this hypothesis, as we increase our accuracy to around 63%. While this is only a modest increase, it is telling that knowing only our tag's initial relationship with the most active 10 tags allows us to make better predictions about later activity in the tag than we could when we knew about the raw quantity of activity within our tag.

In fact, when we train our Random Forest Classifier using all of the features considered so far (that is, the features from Sections 4.2.1 and 3.2.3), the computed feature importances consider the Percent Posts Tagged with Tag and Top Tag as the top feature with 0.33 importance, versus the runner-up with 0.23 importance, involving the raw number of posts with co-occurring tags.

These results provide further support for the hypothesis that predicting a tag's success must look beyond what is happening within the tag itself, and consider the tag's place in the wider affiliation network to understand its relationship with other tags – especially the popular tags.

### 4.2.4 Tag affiliation networks from co-occurring tags

As the next step in our analysis, we ran the classifiers on features from the tag co-occurrence affiliation network described in Section 3.2.4. This network considers tags as nodes, and forms undirected edges between tags weighted by the number of posts where the two tags co-occur. For example, a post with four tags would contribute to six edge weights. This network provides a more nuanced picture of tag overlap to continue the analysis from Sections 4.2.2 and 4.2.4 above. We only considered site-wide activity from the first 28 days of the target tag's existence when weighting the edges, and all features are computed with respect to the target tag for the prediction problem on all 1000 tags, resulting in 1000 networks. Details on how we computed these graph features can be found in Section 3.2.6.

The results in Table 4 show another modest improvement from training on these features, with an average accuracy of around 66%, and generally strong precision recall numbers. The feature importances as computed by the Random Forest Classifier and the AdaBoost Classifier both identified clustering coefficient and average shortest path as particularly important.

Figure 10 presents a plot of the clustering coefficients by rank. There appears to be a loose correlation where tags with lower clustering coefficients appear to become slightly more popular. One possible explanation is that very highly clustered tags contain too much overlap with existing tags which makes it difficult to break out of the tightly-knit meta-community to become individually very successful, because questions tend to be directed to the wider community instead. On the other hand, tags that are moderately well-clustered might attract activity and users from their similar tags, but are still sufficiently standalone to attract a large number of questions directed only at the tag. Regardless, the correlation is weak enough that the prediction problem is still quite challenging.

### 4.2.5 Tag affiliation networks from user activity

Similar to the all-time affiliation network analyzed in Section 4.1, we can also form an affiliation network between tags based on user activity. While the affiliation network in Section 4.2.4 can be formed simply from considering co-occurring tags on posts, an affiliation network based on

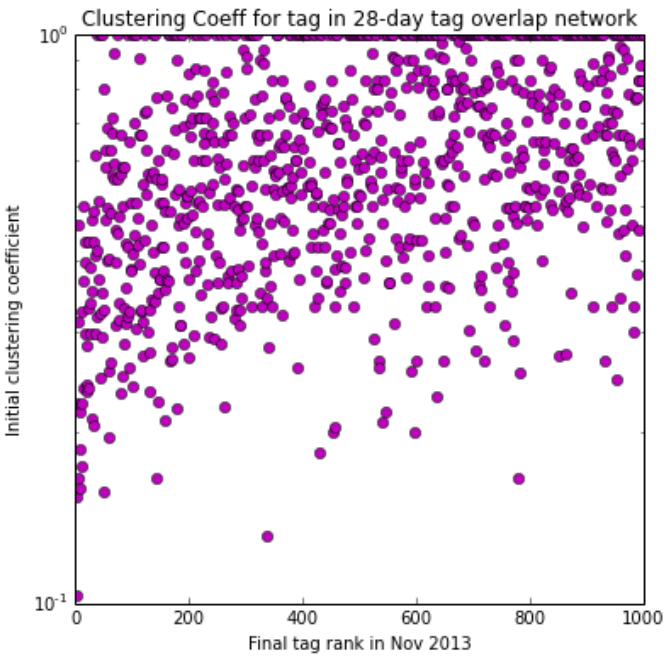


**Figure 10: Clustering coefficient for the tag in the first 28-day affiliation network where edges between tag nodes are weighted by # of posts where the tags co-occur.**

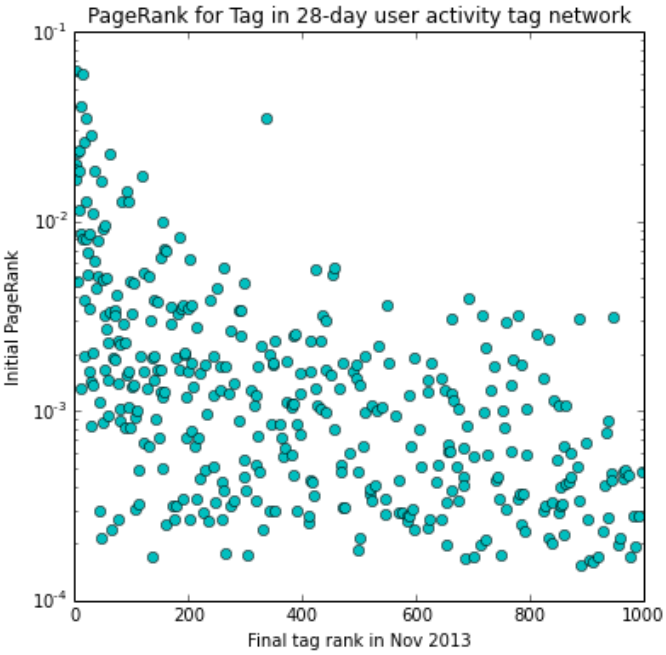


**Figure 11: PageRank for the tag in the first 28-day affiliation network where edges between tag nodes are weighted by user activity across both tags. Note that this plot is sparser, as tags that did not meet the user activity thresholds for edges were given a PageRank of 0.**

**Table 5: Classifying future popularity based on Avg Shortest Path, Page Rank, and Clustering Coeff from the user activity tag affiliation network during first 28 days of tag.** Columns: test/train accuracy, test precision/recall for Less Popular, test precision/recall for More Popular.

| | test | train | prL | recL | prM | recM |
|---|---|---|---|---|---|---|
| Random Forest | .64 | .72 | .71 | .37 | .62 | .87 |
| Linear SVC | .65 | .59 | .65 | .52 | .65 | .77 |
| Logistic Regression | .65 | .59 | .65 | .52 | .65 | .77 |
| AdaBoost | .68 | .63 | .82 | .39 | .64 | .93 |

**Table 6: Classifying future popularity based on all features during the first 28 days of the tag: # questions, answers, comments, users, # and % of overlap with all other tags as well as the top tags, and degree centrality, closeness centrality, average shortest path, PageRank, and clustering coefficient from both overlapping tag networks.** Columns: test/train accuracy, test precision/recall for Less Popular, test precision/recall for More Popular.

| | test | train | prL | recL | prM | recM |
|---|---|---|---|---|---|---|
| Random Forest | .67 | .99 | .61 | .75 | .74 | .60 |
| Linear SVC | .63 | .61 | .57 | .76 | .72 | .52 |
| Logistic Regression | .59 | .62 | .54 | .74 | .68 | .47 |
| AdaBoost | .64 | .78 | .61 | .63 | .68 | .66 |

user activity is more nuanced. As further discussed in Section 3.2.5, we generate this affiliation network where edges between tag nodes are weighted by the number of active users who contribute to both tags.

The results in Table 5 show an average accuracy around 66%, similar to the results from the co-occurring tag network in Section 4.2.4. While there is little improvement, the similarly successful results provide further validation that our affiliation network model based on user activity captures important features of the StackOverflow tag relationships.

We can also consider Figure 11 to understand how a tag's position in the affiliation network predicts later success. The plot suggests that a higher PageRank is slightly more correlated with more popular nodes, a reasonable result as PageRank quantifies the importance of a node in the graph, and nodes in a position of importance during their first 28 days are intuitively seem likely to continue to attract activity and be successful (see Section 4.1.4 for details on the algorithm).

## 5. CONCLUSIONS AND FUTURE WORK

While our best predictions are just around 66% accurate, they arise from features of a tag's connections with other tags in the network. As seen in Table 6, taking all features together does not improve the prediction, suggesting that there may not be enough information in the first 28 days to do better than around 66%. Indeed, it is impressive that it is possible to predict success several years later from the first 10 to 100 posts for a tag during its first 28 days on the site, and that the best predictions arise not from

activity within the tag itself, but from the tag's place in the 28-day affiliation network between tags. To summarize, our analysis of the all-time StackOverflow affiliation network modeled in Section 3.2.1 and evaluated in Section 4.1, combined with the affiliation-based prediction problem modeled in Section 3.2 and analyzed in Section 4.2, provide strong support for the importance of affiliation graphs in social network analysis.

Future work would investigate the length of the initial time window for better predictions, attempting to predict success after 8 weeks or 12 weeks rather than just the first 4 weeks, or basing the initial window on some metric of when the tag has reached a critical mass of users. Part of this work could consider features across the first time chunks, developing feature vectors for each chunk to consider time-based feature vectors over the first 16 weeks. This additional data would lend greater predictive power, allowing for a more precise prediction problem. Beyond level of success based on rank, other prediction problems might be to predict the tag's PageRank in the overall affiliation network, the number of unique users in the tag, and number of page views. Beyond StackOverflow, future work would apply our analysis to other domains with affiliation networks apart from explicit social networks, such as categories of restaurants on Yelp, or affiliation networks that do have attached social networks, such as jobs on LinkedIn.

# 6. REFERENCES


[1] Kmeans, scikit learn. `http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html`.

[2] Stackexchange data explorer, 2013.

[3] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. Discovering value from community activity on focused question answering sites: A case study of StackOverflow. 2012.

[4] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 44–54. ACM, 2006.

[5] balpha. Sizes and overlaps of communities on the StackOverflow site. `http://meta.stackoverflow.com/questions/36590/sizes-and-overlaps-of-communities-on-the-sites`, 2010.

[6] A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11):3747–3752, 2004.

[7] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[8] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117, 1998.

[9] ClearBits. Stack exchange data dump. 2013.

[10] S. Ghosh, S. Saha, A. Srivastava, T. Krueger, N. Ganguly, and A. Mukherjee. Understanding evolution of inter-group relationships using bipartite networks. 2013.

[11] S. R. Kairam, D. J. Wang, and J. Leskovec. The life and death of online groups: Predicting group growth and longevity. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 673–682. ACM, 2012.

[12] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann. Design lessons from the fastest q&a site in the west. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 2857–2866. ACM, 2011.

[13] M. E. Newman. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Physical review E*, 64(1):016132, 2001.

[14] A. Pal, S. Chang, and J. Konstan. Evolution of experts in question answering communities. 2012.

[15] StackOverflow. StackOverflow list of tags by popularity. `http://stackoverflow.com/tags`, November 2013.

[16] E. Zheleva, H. Sharara, and L. Getoor. Co-evolution of social and affiliation networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1007–1016. ACM, 2009.