– Independent Work Report Fall, 2015 –

Finding Experts in Unstructured Communities through Relationships and Topics

Michael Zhang
Adviser: Andrea LaPaugh

Abstract

The problem of expert finding has been explored for many structured question and answer-based social networks. Less explored is the problem of finding trusted individuals in an unstructured social setting. We explore the issue of finding experts in Reddit's /r/programming, /r/CFB and /r/cars Subreddit – each of which is either unstructured or semi-structured – using a machine learning approach based on prior research in classifying user roles. To target the problem, we assumed that "experts" is a role similar to "answer-person" in previous research in that they have a common pattern of communication quantitatively recognizable through analysis of social network features. Using a combination of inferred topic distributions and social network structure, we trained a supervised classifer to find the topical-network "fingerprint" of experts. Among features examined for a given user include neighbor topical similarity, topical entropy, and depth 2-egocentric network properties. We compare our results with previously considered techniques for evaluating expertise: namely, ExpertiseRank (Social Network Page Rank), HITS and a simple z-score metric introduced by Zhang, Ackerman and Adamic (2007). Using our role-based algorithm, average f1 and recall scores during validation demonstrate comparable or better performance than prior social network analysis-based algorithms. We conclude by describing future metrics that can be explored in the problem of expert classifying in pure discussion communities.

1. Introduction

Reddit is a social media website based around link sharing. Users can link to any other website they find interesting and the community as a whole can vote on links, and possibly leave a comment. Comments themselves have their own voting system, and other users reading them can vote them

Figure 1: Example comment reply thread with associated score. Each indent indicates a reply.

up or down depending on the perceived quality of the link reply, and leave a reply to that particular comment as well. Each vote and each comment is assigned a score, based on the number of upvotes (likes) and the number of downvotes (dislikes). Reddit has become a major source of news and opinion sharing for many of its visitors, and its comment section is known to be a haven for similar-minded individuals to discuss and express discontent on a wide range of political, social and cultural issues from presidential elections to police violence to new movies. Reddit typically receives millions of unique visitors per month leaving tens of millions of comments per month. The Reddit community prides itself in its relative free speech and is fiercely defensive against perceived encroachment of community freedoms to discuss. This type of environment paves the way for lively yet unstructured discussions across the board.

As a website, Reddit is divided into many subcommunities called "Subreddits". These Subreddits are designed to narrow the focus of links to a specific topic, community or structure. For instance, the Subreddit "/r/gaming" has posts about new video games, "/r/funny" contains humorous links and pictures, "/r/programming" contains links and discussions about general coding news, "/r/omaha" contains discussions related to the city of Omaha, Nebraska, and so on. Compared to many similar sites, the comments section of Reddit is notable in its diversity of user types stratified based on the borders defined by the Subreddit. Insider lingo, references and jokes are common. Many Subreddits have posters that are trusted to have particular interesting, funny or helpful posts. For the purpose of this work, we consider these users to be the "experts" of a given Subreddit. In this work, we seek to

find common trends of posting activity for experts in Subreddits. In doing so, we hope to measure the performance of a fingerprinting approach to finding experts in communities that do not have a strongly imposed structure. In other words, we hope to find experts outside of a traditional question and answer-based forum domain. First, we discuss some related prior research.

2. Related Work

2.1. Expert finding in Question and Answer Networks

The problem of finding experts in a question and answer based social network has been explored by Zhang et al. on the Java Forum – an online help seeking community of programmers – and Adamic et al. on Yahoo Answers – a general question and answer based platform. [21, 1]

Zhang et al.'s research explored the efficacy of two social network-based techniques for ranking experts on social networks – the ExpertRank and HITS algorithms – relative to a simple z-score based measure. Their work utilized the assumption that in the Java Forum, a asker-replier relationship indicates greater expertise in a specific area on the part of the replier. This assumption comes from the structure of the forum: when a post is created, it is most likely a user asking a question about Java Programming. Zhang and team constructed a graph of users based on question asker and responder relationships. Each vertex is a user and if A replies to B, then B has a directed edge to A. From this graph, the team ran ExpertRank – a modified PageRank algorithm with users instead of webpages and reply-response relationships instead of links. Recall that the PageRank algorithm gives high weight to pages that many links refer to, but even more so to pages referred to by high weight pages themselves. In terms of Zhang et al.'s network, this translates to the idea that if a user replies often, he will have a greater chance of being classified as an expert, but this is even more true if the user replies to an expert himself. The ExpertRank of a user is thus the expertise level of the user. The HITS (Hyperlink-Induced Topic Search) algorithm takes a slightly different link analysis approach than PageRank (or Expertise Rank). The algorithm iteratively assigns each node in the graph a hub and authority score. The hub score of a node is dependent on the authority score of its neighbors, while the authority score is dependent on the hub score of its neighbors. Intuitivity

the algorithm attempts to divide nodes into hubs – users helped by many different experts – and authorities – users who "help" many different users in a specific problem domain. The authority score of a user node was thus interpreted as the expertise level of the user. Zhang et al. compared these two techniques with a simpler approach using z-score. We describe this approach later when we benchmark our algorithm. From an analysis of two statistical measures – Spearman's Rho and Kendall's Tau – in comparison to manually ranking experts by hand, Zhang et al. determined that the z-score approach performs slightly better than the Expert Rank and HITS algorithm in the Java Forum dataset. [21, 16, 12]

Adamic et al. performed an in-depth analysis of the Yahoo Answer's forum. Part of their work included finding characteristics that correlate with an answer response being the best one in a question thread. Adamic et al. found variation in metric-answer success correlation depending on topic. One such metric the team explored was comment entropy. At the high level, comment entropy is a measure of the divergence a user demonstrates in his post choices. A user with a tendency to post in only a few select topics demonstrates low entropy, but a user who answers a very diverse range of questions in terms of topic has higher entropy. The team found that in technical subcategories of Yahoo Answers like "math", "physics" and "biology", users with lower comment entropy tended to be more highly correlated with writing the optimal answer choice. Other user-specific variables that correlated with high success probability included reply length and number of prior best answers. [1]

2.2. Role finding using Social Networks

On the other side of the analysis spectrum, the task of finding users fulfilling specific roles has been evaluated through sociological study and a variety of network techniques. We discuss some relevant analysis below.

Descriptions of user roles in social networks were formalized in the non-computation sociological work of Golder and Donath. Their work analyzes the type of individuals in Usenet communities including the "celebrity", the "troll", and the "newbie". Golder and Donath's work is referenced in

various other works in the social network analysis of Usenet communities. Welser et al. for instance, characterized the features that describe a "fingerprint" for the answer-person user role on Usenet. The answer-person is a user who is known for answering the questions of newer users. Welser identifies this role as significant because it is typically characterized by noticeable social network trends. Among these trends include disproportionate tendency to reply to isolated users, low triangle count (individuals they reply to tend not to know each other), and tendency not to send multiple messages to the same recipient. In his results, Welser found a strong correlation between measure of sporadic thread activity and degree of answer-person-like behavior. So an individual who does not start many threads but responds often by making few posts to many threads is more likely an answer-person. Fisher et al. expanded on Welser's work by placing more focus on networks containing a query user in the center, all of his directly adjacent neighbors, and all users adjacent to those neighbors (degree-2 egocentric networks). To generate the egocentric network, Fisher et al. described the following type of graph: each vertex is a user, with a directed edge existing from A to B if A had replied to B in some thread (note that this is the opposite as described in Zhang et al.'s work). These edges are weighted by the number of replies. So an edge from A to B will have weight n if A had replied to B n times. From an analysis of common properties like interconnectivity of members and reciprocity of ties, Fisher et al. confirmed that answer people tend to have lower tie reciprocity in technical threads like those about servers, along with low interconnectivity. [7, 9, 20]

Most relevant to the exploration of roles specifically on Reddit come from Buntain and Golbeck. Buntain and Golbeck tested the effectiveness of egocentric graphs in "fingerprinting" the answer-person role on Reddit for the sake of building an automatic classifier. They assumed that user fingerprints for the answer-person role can be determined by analyzing features of the 1.5-depth egocentric network surrounding a user. Using a training set of over 200 users, the pipeline they developed trained multiple decision tree models based on calculated metrics on a training user's egocentric network. To construct the egocentric network, Buntain and Golbeck took inspiration from the graph building strategy of Fisher et al. Among metrics tested included a clustering coefficient measure, a measure of density, of neighboring nodes with low degree, number of direct neighbors

with low degree, and proportion of neighbors with degree greater than 1. Their work achieved an accuracy rate ranging from .66 to .92 in predicting whether a given user is an answer-person. [4].

2.3. Topical modeling in Social Networks

To model topic distributions for a given user, we looked into using the LDA (Latent Dirichlet Allocation) model. The model assumes that each document in a corpus is created through a generative process of selecting a length $N \sim Poisson(\xi)$, a topic distribution $\theta \sim Dir(\alpha)$, and for values 1 to N, selecting a topic $z_n \sim Multinomial(\theta)$ and a word w_n with multinomial probability $p(w_n|z_n, \beta)$ conditioned on the topic z_n and some latent parameter β. β encodes the probability a given word w_i appears given that we have chosen topic z_j. [2]

Prior work in using topical models to analyze social networks has been done by Pennacchiotti and Popescu. In their analysis of Twitter, the group built an LDA topic model with users as documents and the full content of a user's Tweets as the body of a document. A user could thus be inferred to have a multinomial distribution over K topics. Pennacchiotti and Popescu assumed a fixed topic count of K = 100 and trained their model over the full corpus of 2 million users. From the model, Pennacchiotti and Popescu inferred topical distributions for each user. Using information from these distributions and some other Twitter metadata, the team built a gradient boosted decision tree classifier to label users by their political affiliation, their race and their affinity for Starbucks. Pennacchiotti and Popescu managed to achieve relatively high precision, recall and f1 scores for classifying users by political affiliation and product affinity but found weaker success in classifying ethnicity. [17, 2]

3. Approach

3.1. Data

On July 3rd 2015, user Stuck_in_the_Matrix posted in Subreddit "/r/datasets" a collection of every publicly available Reddit comment up until that date.[1] We filtered the data for every post in the

[1]The dataset is available for download here (web link)

Level	Comments	Users
Sitewide (1/2015)	53,851,542	2,512,123
/r/programming	46,553	11,365
/r/CFB	400,464	25,410
/r/cars	95,840	13,331

Table 1: Comments and users in at various levels in the dataset

Subreddit	Post Cutoff	Score Cutoff	Active Users	Experts
College Football (CFB)	130	11.3	500	50
Cars	33	7.2	502	49
Programming	15	10.2	499	50

Table 2: Cutoff and user counts for each Subreddit under consideration

month of January 2015. We further filtered for only comments in the "/r/programming"[2], "/r/CFB"[3] (college football) and "/r/cars"[4] Subreddits. The number of comments and users are shown in table 1 above.

For training, we focused on users who have a significant number of posts. As an upper limit of cases to consider, we only trained and tested our algorithm on the top 500 or so most active users in a given Subreddit. To do this, we defined a lower bound for the number of posts a user needs to have made to be considered in our study. For "/r/programming" this threshold was 15 comments per a user, for cars 33, and for CFB a substantial 130 (college football is a much more controversial topic than programming!). To obtain ground truth expertise, we filtered our active set for the top 50 or so users in terms of average comment score. To do so, we had to define different score cutoffs for each Subreddit. For programming this cutoff was 10.2, for cars 7.2 and for CFB 11.2. This information is listed in table 2 above.

We chose the Subreddits under consideration based on size, topic and structure. Programming is by far the smallest Subreddit, but it presents a more technical topic. CFB is in many ways very similar to Programming in terms of structure and community, but it is much larger and presents a less technical topic. Cars is the outlier community among the three in that, though it has many discussions similar to the ones in /r/CFB and /r/programming, many of its posts tend to be more

[2]https://www.reddit.com/r/programming/
[3]https://www.reddit.com/r/CFB/
[4]https://www.reddit.com/r/cars/

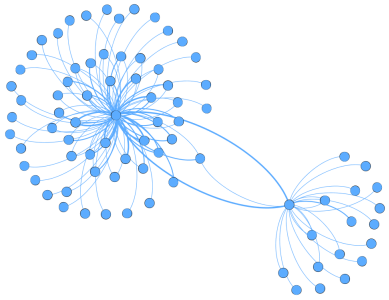

Figure 2: Example expert user egocentric network.

structured. The community tends to have users who will ask questions about cars and parts. In addition, manys posts are in fact users presenting pictures of their new vehicles for comment and advice on ownership. In this sense, cars is perhaps the closest to a traditional question and answer forum in that it is semi-structured.

3.2. Packages for implementation

We construct our egocentric networks in the style of Welser: for each node A, A has a directed edge to B if A has replied to B. For a given user, the egocentric network consists of the user, his immediate neighbors and the neighbors of those neighbors. The weight of an edge in this network is equal to the number of times in which A has responded to B. We built our egocentric network using the Python NetworkX package [5]. A sample egocentric network is shown figure 2 of an exemplar expert. The visualizion was made using Gephi.

To model topic distributions, we used an LDA model with users as documents and the full body of all of the user's comments concatenated with newlines as the content of the documents. The full comment corpus thus contained the bodies of all $46,554$ comments split into $11,365$ documents. We fixed the number of topics generated to 50. A sample list of the most common words for 10 of these topics is shown in table 3. For each document, we filtered for common English language stop words found in the Python nltk (Natural Language Toolkit) package and the Python stop-words

[5] https://networkx.github.io/

Topic #	Top 10 word stems
0	test, unit, code, write, chang, tdd, root, email, offici, pass
1	us, ai, programm, human, will, hire, goal, compet, 10x, anyon
2	hash, length, delet, proven, assert, charact, wherea, password, directori, alloc
3	linux, window, kernel, tab, extens, indent, git, packag, ahead, instal
4	javascript, use, c, librari, js, support, languag, recurs, imper, browser
5	program, languag, learn, rust, c, python, teach, book, first, student

Table 3: Top 10 word stems for 6 different topics generated by the LDA model.

package[6] [7]. In addition, we removed common special characters present in Reddit comment syntax (a variant of markdown syntax), urls, numbers and non-ascii characters. Finally, we took the filtered text and stemmed each token using a porter stemmer. The final results were then converted into a bag of words data structure, mapping word stems to integer ids, before being fed into a topic modelling package. The LDA implementation we used can be found in Řehůřek and Sojka's Gensim package for the Python programming language[8]. [18]

3.3. Strategy

Figure 3 shows a high-level view of our classification scheme. We describe some of the assumptions we make and the features we generate below.

Our approach emphasizes properties of the local network surrounding the user. Because we are analyzing a discussion based social network we cannot make assumptions about relative expertise in poster-replier relationships as Zhang et al. and Adamic et al. could make on the Java Forum and Yahoo Answers, respectively. Instead, we hypothesize that Reddit experts will have clear social network properties like the answer-people role described in Welser, Fisher and Buntain. As a summary of the prior work and an explanation of our ideas, we assume that like answer-people, experts will have relatively low degree neighbors, few strong ties and a lighter network density. These assumptions are derived from the idea that when an expert makes a post, many users may be inclined to respond to that post with positive reactions because of the expert's wit or helpfulness. These users may not be frequent posters or connected to each other otherwise.

[6] http://www.nltk.org/
[7] https://pypi.python.org/pypi/stop-words
[8] https://radimrehurek.com/gensim/

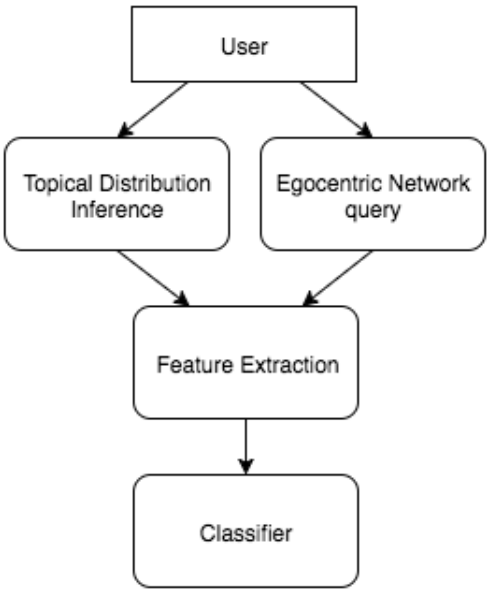

Figure 3: Flowchart representing our expert finding technique. During training and testing, given a query user we generate the egocentric network and infer a topical distribution. Using these, we extract a set of features that we believe are meaningful for determining whether a user is an expert. These features are then fed into a classifier for either training or classification. In our implementation, the classifier is a decision tree.

The assumptions mentioned above lead us to explore six features of a user's egocentric network:

1. Clustering coefficient
2. Density of network
3. Undirected ratio of neighbors with low degree
4. Proportion of replier nodes with low degree
5. Proportion of user replied to nodes with low degree
6. Proportion of neighbors with deep ties to the query user

Note that a majority of these features were inspired by either Buntain and Golbeck, Welser et al. or Fisher et al. [4, 20, 7]

In addition to hypotheses about an expert's network, we also made two assumptions about expert topical distribution. Based on Adamic's research, we assumed that an expert would likely have lower topical entropy than an everyman poster. This is based on the idea that most experts have a limited range of topics in which they are confident in writing about. We also assumed that an expert would be dissimilar to the nodes in which he is connected. This is based on the idea that repliers

may attempt to connect an expert post with other topics that may be tangentially relevant to the original post.

These two assumptions lead us to add three additional features to our model's analysis.

1. Topical entropy
2. Average topical divergence with neighbors who have replied to the user
3. Average topical divergence with neighbors the user has written a response to

After obtaining these features, we will train a decision tree classifier and measure performance against three other algorithms. We will first discuss how we implemented measurements of these features in more detail below.

3.4. Clustering coefficient

There are typically two types of clustering coefficients defined for a graph. The global clustering coefficient is a measure of how many tight knit groups are present in an entire network while the local coefficient measures how close one node in a graph is to forming a clique with its neighbors. For the purposes of our approach, we use the unweighted, undirected local clustering coefficient for each user. The value of this local clustering coefficient at u is

$$c_u = \frac{2T(u)}{deg(u)(deg(u)-1)}$$

Where $T(u)$ is a measure of the amount of triangles through the node u. Intuitively, this measure is equal to the fraction of the number of possible triangles that are present around u. [10]

3.5. Density of network

The density of the network of u is the fraction of the number of possible edges in the undirected egocentric network of u that are present in the actual graph. In the form of an equation,

$$d_u = \frac{2|E|}{|V|(|V|-1)}$$

[4]

3.6. Undirected ratio of neighbors with low degree

This measure is the smoothed ratio of the number of users in the egocentric network with low degree versus the number of users without low degree. Low degree was defined as having an undirected degree of less than 4. For a given node u, this metric is

$$l_u = \frac{|low| + 1}{|V| - |low|}$$

Where $|low|$ is the number of nodes in the egocentric network around u with degree less than 4. [4]

3.7. Proportion of replier nodes with low degree

This measure is the proportion of nodes that have responded to u that have a degree less than 2. In other words, this is a measure of all repliers that have only replied to u in their entire corpus of comments. In other words,

$$r_u = \frac{|low| + 1}{|repliers|}$$

Here $|repliers|$ is the total number of user nodes with a directed edge to u. $|low|$ is the total number of replier nodes that have u as its only directed neighbor. We expect this measure to be fairly high for our hypothetical expert. This could potentially, for instance, indicate that the user has many single time "I agree!" type responders. [4]

3.8. Proportion of response nodes with low degree

This measure is the reverse direction of the measure above. It indicates the proportion of nodes that our target user u has responded to, that have no other neighbors but u. In other words, it is the number of one time users that u has replied to. As an equation,

$$e_u = \frac{|low| + 1}{|responses|}$$

Here $|responses|$ is the total number of user nodes that u has a directed edge to. $|low|$ is the number of these nodes that have u as its only neighbor. This measure is in some ways a sanity check. We do not expect it to be that high for experts, but if it were, it would indicate that our expert users are fairly one to one with the assumptions made about answer people in prior work. This would suggest that experts may in fact be answer-people in a discussion forum. [4]

3.9. Proportion of neighbors with deep ties

This is a measure of the proportion of users with more than one interaction with the egocentric target user u, regardless of whether those interactions were a reply left or a reply received.

$$d_u = \frac{|deep|}{|neighbors|}$$

Where $|deep|$ is the number of neighbors with more than one interation. [4]

3.10. Topical entropy

Using our LDA topical model, we can obtain an inference of the weights applied to each topic for a given user document (the complete corpus of a user's comments, filtered and processed). Intuitively, these weights correspond to the probability that a given user would select some topic, with higher weights suggesting a user would more likely select the topic during document generation. To calculate topical entropy, we used the normalized value of these weights and applied the base e entropy formula. Let w_i be the normalized weight on the ith topic for user u. The entropy for user u is then

$$H(u) = -\sum_{i=1}^{K} w_i log(w_i)$$

Where K is the number of topics (in our case 50) and log refers to the natural logarithm. [1]

3.11. Average topical divergence

We measured two types of average topical divergence for a user u: reply divergence and response divergence. Reply divergence indicates the divergence of topic weights between the user u and all of his outward neighbors (the user he has replied to). Response divergence indicates the divergence in topic weights between the user and all of his inward neighbors (the users who have responded to him).

To measure topical divergence, we used the Kullback-Leibler (KL) divergence measure. KL divergence is a tool to measure the difference between two distributions, P and Q. It is not symmetric, and intuitivity $D_{kl}(P||Q)$ is a measure of the amount of information lost when Q is used as an approximation for P. Symmetric versions of the measure exist, but for our purpose of measuring the divergence of a user u from his neighbors, we use the unsymmetric form with user u's distribution as P and the neighboring distribution as Q. Hence

$$div(u,v) = D_{kl}(T(u)||T(v)) = \sum_{K} w_i log(\frac{w_i}{x_i})$$

Here, u is the target user and v is some neighbor of u. $T(u)$ is a hypothetical function returning the topical distribution of u. w_i is the normalized weight on topic i from $T(u)$, x_i is the normalized weight on topic i from $T(v)$, K is the number of topics and log indicates the natural logarithm.

Computing the average divergence is thus

$$\frac{1}{|neighbors(u)|} \sum_{v \in neighbors(u)} div(u,v)$$

This formula is applied twice to obtain both reply neighbors (outwards edges) divergence and response neighbors (inwards edges) divergence. In the first case neighbors refers to outward edges, in the second case, inward edges. [5, 11]

4. Experiments, Comparisons and Results

4.1. Training a classifier

Using our 9 parameters, we trained a decision tree classifier using the Python scikit-learn package[9]. To reduce overfitting, we experimented with different max_tree_depth values. The final tree depth value we settled for was $d = 10$.

Because our datasets are very unbalanced (the ratio of expert to non-experts is one to 10 by our splitting process), we weighted the expert class by a value 10 times higher than the non-expert weight to prevent excessive bias towards the non-expert class. To train our classifier, we used a k-fold cross-validation procedure with k = 5. For each Subreddit, We split our example sets of around 500 active users into 5 folds of approximately 100 users each. For $i = 1...5$, we trained a model using every fold except for the ith one. Finally, we tested our model using fold i. To evaluate our model, at each step we calculated the precision, recall, f1 scores and the Matthews correlation coefficient. After all scores have been generated, we averaged our 5 values.

4.2. Benchmarks and implementation

We benchmark our algorithm against purely global link-analysis based classification techniques. First, we discuss the ExpertRank, HITS and z-score method described by Zhang et al.

As briefly alluded to in prior work, the ExpertRank algorithm assigns a ranking score to each node in the complete social network based on reply relationships. In Zhang et al.'s work, the relationship between original poster and replier is described as the original poster having a directed edge to all users that have replied to him. In our benchmarks, we will explore using the algorithm in both the case where the replier is the source of the directed edge and where the replier is at the end of the directed edge. [21]

We implemented ExpertRank by modifying the PageRank implementation described in [6]:

1. Initialize all nodes to have a rank of $1/n$

[9] http://scikit-learn.org/

2. For 100 steps, have each node divide its rank score equally by the number of outgoing links, and "distribute" it score to its outgoing links. The score of each node at the next step will be the sum of each rank score it receives from its incoming links.
3. Select the top 50 ranked score users as experts. The rest of the users are non-experts.

The HITS-based algorithm, also described by Zhang et al. is another link analysis-based approach for expert finding. In this approach, we assign each node a hub and authority score. We then update the scores for some number of steps. The modified algorithm we implemented is again inspired by the description from [6]:

1. Initialize all nodes to have a hub and authority score of 1.
2. For 100 steps: update the authority score of each node to equal the sum of the hub scores of all nodes that point to it, update the hub score of each node to equal the sum of Authority scores of all nodes it points to.
3. Normalize all hub and authority scores so they sum to 1.
4. Select the top 50 authority scores. Rank these users as experts.

An interesting property of the HITs algorithm is that the hub score of the replies-are-outflow graph is equal to the authority score of the replies-are-inflow graph. For this reason, we only consider the authority score of each graph (in-flow and reverse) when using the algorithm to rank users.

Finally, we implement a variant of a z-score approach that is also approximately similar to the one described by Zhang et al. In this approach, we simply measure a user's deviation from a hypothetical mean number of replies and reply targets. We assume that a typical user with n links has on average $n/2$ of his links as replies and $n/2$ resulting from being replied to. Hence, the probability that a given link belonging to the user is from a reply the user himself made occurs with $p = 1/2$ and with standard deviation of $\sqrt{n * p * (1-p)}$. The user reply z-score is then

$$z_r = \frac{r - n/2}{\sqrt{n/2}}$$

Where r is the number of replies the user himself made and n is the total number of links the user has with other users. We also evaluate the effectiveness of the alternative z-score measure based on the number of links connected to a user that were made from another user's reply.

$$z_o = \frac{o - n/2}{\sqrt{n/2}}$$

Here o is the number of times another user has replied to the user.

4.3. Purely Random Classification

We also benchmark our results against a purely random classification approach. We consider the precision, recall and f1 scores of two random classifiers: the first classifier labels a user as an expert with random probability $1/10$, the second with random probability $1/2$.

Recall that

$$precision = tp/(tp + fp) \tag{1}$$

$$recall = tp/(tp + fn) \tag{2}$$

$$f_1 = 2 \cdot (precision \cdot recall)/(precision + recall) \tag{3}$$

Were tp is number of true positives, fp is the number of false positives, and fn is the number of false negatives.

Assume we have n total users with $n/10$ experts and $9n/10$ non experts. In the $1/10$ classifier, we expect that

$$precision_{10} = (n/100)/[(n/100) + (9n/100)] = 10n/(100n) = 1/10 \tag{4}$$

$$recall_{10} = (n/100)/[(n/100) + (9n/100)] = 10n/(100n) = 1/10 \tag{5}$$

$$f_1 = 1/10 \tag{6}$$

For $1/2$, we have

$$precision_{10} = (n/20)/[(n/20) + (9n/20)] = (n/20)/(n/2) = 1/10 \tag{7}$$

$$recall_{10} = (n/20)/[(n/20) + (n/20)] = 10n/(100n) = 1/2 \tag{8}$$

$$f_1 = 1/6 \tag{9}$$

4.4. Results

The Matthews correlation coefficient is a measure of binary classification performance. It gives a value between -1 and 1, with 1 indicating a perfect prediction, -1 indicating complete disagreement and 0 indicating no better than random performance.

Our results for precision, recall, f1 and Matthews correlation coefficient scores are shown in table 4 for programming. Cars and CFB are shown in tables 5 and 6 respectively. We split ExpertRank results between categories of "outdegree replies" and "indegree replies". Outdegree replies refers to an ExpertRank algorithm applied to the complete social network of all users where, if A replies to B, then A has a directed edge to B. "Indegree replies" refers to the opposite case: where B would have an edge to A. z-score is split into the two cases where we rank by the number of replies the user has made (number of replies) and the number of replies the user has received (number of responses), as described above.

For programming, we see that our decision tree algorithm outperforms the other techniques in all scores. Other algorithms worth mentioning include outdegree ExpertRank and response z-

Algorithm	Precision	Recall	F1 Score	MCC
***Ego-topic feature decision tree**	0.304	0.526	0.384	0.309
***ExpertRank (outdegree replies)**	0.222	0.200	0.211	0.128
HITS authority (outdegree replies)	0.021	0.020	0.020	-0.086
ExpertRank (indegree replies)	0.040	0.040	0.040	-0.067
HITS authority (indegree replies)	0.000	0.000	0.000	-0.110
z-score (number of replies)	0.040	0.040	0.040	-0.089
***z-score (number of responses)**	0.300	0.300	0.300	0.222
Random (p=.1)	0.100	0.100	0.100	0.000
Random (p=.5)	0.100	0.500	0.167	0.000

Table 4: For /r/programming: Precision, recall and f1 for our decision tree model and benchmark algorithms. The most successful techniques are in bold.

score. ExpertRank for outdegree nodes performed significantly better than both random algorithms in precision and mcc scores, losing slightly to random .5 in f1 score. The z-score technique outperformed every other algorithm except our decision tree, and the random .5 algorithm in recall. Note that though its precision is lower than our decision tree, the difference is very small. The HITS ranking algorithm underperformed in this model, failing to classify any expert in the indegree case, but classifying some in the outdegree case.

A similar result emerges for college football. As we mentioned earlier, we chose college football because it represented a denser, less technical Subreddit than programming. Notice that unlike in programming, the z-score metric performs better than our classifier in all scores except recall.

The results in cars is noticeably different from the rest. One thing that we can point out is that the HITS authority for number of replies is a substantially better metric than it is in the other two Subreddits, performing even better than ExpertRank. In addition, though the classifier method and z-score (responders) method are still the best classification mechanisms, their scores are substantially weaker than they were when used in CFB and programming.

5. Analysis

5.1. Inverse answer person metric

Our results appear to suggest that the number of responses a user receives is a fairly strong indicator of being in the top 10% of comments in both structured and semi-structured settings. In a sense,

Algorithm	Precision	Recall	F1 Score	MCC
***Ego-topic feature decision tree**	0.322	0.548	0.390	0.323
***ExpertRank (outdegree replies)**	0.300	0.300	0.300	0.222
HITS authority (outdegree replies)	0.208	0.200	0.204	0.118
ExpertRank (indegree replies)	0.060	0.060	0.060	-0.044
HITS authority (indegree replies)	0.149	0.140	0.144	0.053
z-score (number of replies)	0.000	0.000	0.000	-0.111
***z-score (number of responses)**	0.400	0.400	0.400	0.333
Random (p=.1)	0.100	0.100	0.100	0.000
Random (p=.5)	0.100	0.500	0.167	0.000

Table 5: For /r/CFB: Precision, recall and f1 for our decision tree model and benchmark algorithms.

Algorithm	Precision	Recall	F1 Score	MCC
***Ego-topic feature decision tree**	0.237	0.441	0.300	0.219
*ExpertRank (outdegree replies)	0.080	0.083	0.082	-0.018
HITS authority (outdegree replies)	0.041	0.042	0.041	-0.061
ExpertRank (indegree replies)	0.040	0.042	0.041	-0.063
HITS authority (indegree replies)	0.102	0.104	0.103	0.007
z-score (number of replies)	0.020	0.021	0.020	-0.086
***z-score (number of responses)**	0.240	0.250	0.245	0.163
Random (p=.1)	0.100	0.100	0.100	0.000
Random (p=.5)	0.100	0.500	0.167	0.000

Table 6: For /r/cars: Precision, recall and f1 for our decision tree model and benchmark algorithms.

experts in discussion threads appear to be similar to a reverse answer-person or reverse question and answer forum expert. While an answer-person or Java Forum expert is characterized by a high number of replies made, our results appear to demonstrate that an individual with many responses tends to make very popular posts. This is apparent from the strong precision performance of the z-score classifier which merely classifies based on how far above "average" a user is in terms of number of replies received. The strong performance of outdegree ExpertRank in programming and CFB also points to this fact. Intuitively, outdegree ExpertRank ranks users who are replied to often higher but even more so if the repliers are replied to themselves. Even in the case of HITS authority score, which demonstrated pretty weak indicator scores across the board in our unstructured Subreddits, performance was much better when we measured authority score based on outdegree replies. As we mentioned earlier, in HITS, outdegree authority score is equal to indegree hub score. This suggests that a "hub" user receiving many replies, regardless of degree, tends to

Feature	Programming	Cars	CFB
Clustering Coefficient	0.077	0.075	0.058
Network Density	0.019	0.083	0.035
***Low Degree Fraction (Successors)**	0.131	0.142	0.150
***Low Degree Fraction (Predecessors)**	0.371	0.281	0.373
Undirected Low Degree Ratio	0.049	0.025	0.035
Undirected Intense Ties Proportion	0.089	0.110	0.087
***Topical Entropy**	0.041	0.125	0.038
***Topical Divergence (repliers)**	0.181	0.084	0.185
Topical Divergence (replies)	0.042	0.073	0.038

Table 7: Gini importance for each of the 9 features used in our decision tree. The most important features are bolded.

have more expert tendencies than a hub making responses.

These results are either inverted or weakened in our semistructured setting of /r/cars. In this Subreddit, HITS performed relatively well in the indegree replies case. This is normally suggestive of experts being answer people. Recall again the implication of a strong HITS indegree authority score. Because of the duality of indegree and outdegree, strong indegree score suggests that a user is a "hub" who makes many replies. Hence only in our more structured setting does it reasonable to seek answer-people like individuals as experts. This is not a full endorsement of this method however, as cars does appear to have the same preference for outdegree ExpertRank and the ever powerful number of responses z-score measure. Though the effectiveness of these methods is substantially lower despite cars being more popular than programming.

5.2. Feature importance

The impurity of a decision tree node is the probability that an item will be misclassified, given all the items available at that node. When a feature split decision is made, the impurity level of the two resulting children will be less than that of their parents assuming a split better than 50-50. The importance of a feature is generally measured by the sum of decreases of an impurity measure for a given variable over all splits of the decision tree. In the case of the Gini importance measure, the Gini impurity measure is used to characterize node error:

$$\sum_{i=1}^{m} f_i(1 - f_i)$$

Where m is the number of classes and f_i is the fraction of items with class i. [3]

Table 7 lists the average Gini importance of each feature in our training sets, over all 5 folds of the validation step over the 3 Subreddits. A higher importance score indicates that a feature did more "work" in dividing the sets into the classes of "expert" and "non-expert". In our naming scheme, Low Degree Fraction (Successors) refers to the proportion of nodes that a user has replied to with low degree. Low Degree Fraction (Predecessors) on the other hand, refers to the proportion of nodes that have replied to the user. Topical Divergence of repliers refers to the average divergence between a user and his repliers. Divergence of replies, on the other hand, refers to the divergence between a user and those he replies to.

Notice that the most important features for CFB and programming are average topical divergence of repliers and low degree fraction of predecessors. For cars, topical divergence is less important than topical entropy. Low degree fraction of predecessors, however remains very important, albeit less so. Indeed, each Subreddit places relatively high importance in both Low Degree fraction indicators. Some other interesting difference between cars and the other Subreddits include slightly higher evaluation for intense ties proportion, entropy, and divergence of replies.

5.3. Correlation Coefficients: Pearson and Point Biserial

For each feature, we calculated the Pearson correlation coefficient between the feature values and average user score and the point-biserial correlation coefficient between the feature values and whether the user is an expert. Like the Matthews correlation coefficient, the Pearson correlation coefficient assigns each feature a score between -1 and 1. Where 1 indicates perfect correlation, -1 indicates perfect negative correlation and 0 indicates no correlation. The point-biserial correlation coefficient has a similar range of values and meaning as the Pearson coefficient, but we assume that one of the variables is a two class binary variable as opposed to a continuous range. In our case, we assume the two classes are "expert" and "non-expert".

Feature	Programming	p-value	Cars	p-value	CFB	p-value
Clustering Coefficient	-0.060	0.180	-0.072	0.108	-.151	0.001
***Network Density**	-0.166	0.000	-0.146	0.001	-0.102	0.022
Low Degree Fraction (Successors)	-0.106	0.017	0.016	0.728	-0.033	0.461
***Low Degree Fraction (Predecessors)**	-0.207	3.119e-6	-0.303	4.232e-12	-0.176	7.495e-6
Undirected Low Degree Ratio	-0.161	-0.150	-0.128	0.004	-0.093	0.038
***Undirected Intense Ties Proportion**	-0.150	0.001	-0.208	2.629e-06	-0.051	0.251
Topical Entropy	0.0323	0.471	0.107	0.017	0.095	0.0340
Topical Divergence (repliers)	0.070	0.121	0.062	0.162	-0.002	0.969
Topical Divergence (replies)	0.064	0.152	0.032	0.482	0.064	0.149
***Responder Ratio**	0.303	4.562e-12	0.391	1.016e-19	0.405	3.562e-21

Table 8: Pearson correlation coefficient for the decision tree features and for responder ratio. This correlation measurement is between the feature values and the *average user comment score*. Significant features are bolded.

Feature	Programming	p-value	Cars	p-value	CFB	p-value
Clustering Coefficient	-0.053	0.240	-0.063	0.162	-0.084	0.057
Network Density	-0.106	0.018	-0.042	0.349	-0.037	0.405
Low Degree Fraction (Successors)	-0.059	0.187	0.041	0.360	-0.013	0.776
***Low Degree Fraction (Predecessors)**	-0.172	0.000	-0.155	0.000	-0.079	0.075
Undirected Low Degree Ratio	-0.095	0.033	-0.030	0.502	-0.030	0.498
***Undirected Intense Ties Proportion**	-0.119	0.008	-0.117	0.008	-0.137	0.002
Topical Entropy	0.072	0.109	0.053	0.240	0.157	0.000
***Topical Divergence** (repliers)	0.146	0.001	0.104	0.020	0.094	0.034
Topical Divergence (replies)	0.068	0.127	0.060	0.180	0.030	0.505
***Responder Ratio**	0.297	1.241e-11	0.235	9.41e-8	0.327	5.602e-12

Table 9: Point-Biserial correlation coefficient for the decision tree and for responder ratio. This correlation measurement is between the feature values and *a boolean value indicating whether the user is an expert*. Significant features are bolded.

We show our results for the correlation coefficients in table 8 and table 9. Note that for each table, we included a bonus feature that was not included in our decision tree model. Namely, the responder ratio. The responder ratio is the fraction of a user's links that resulted from another user leaving a response to the user's comments. This is a very similar value as what would be used in our z-score classification benchmark.

We see that, indeed, responder ratio has a very strong positive correlation with both user score and with expertise classification in every single Subreddit, and very slightly weaker in cars. In addition, notice the relatively strong negative correlation between between low degree fraction and expertise measures across Subreddits. This suggests that the best commenters may have many responders, and these responders will often be relatively active participants of the community (ie. lower low degree replies indicates greater expertise). A similar argument applies for the negative correlation apparent between score and low degree ratio. Note that for topical divergence (repliers), the correlation is substantially stronger using the point-biserial measure than the Pearson correlation with average comment score. This feature appears to show that topical divergence can be a decent tool to classify a user as an expert but not as useful a tool to measure the depth of a user's expertise. In addition, both the Pearson and the point-biserial coefficient argue for a slight negative relationship between intense ties proportion and expertise. This suggests that though active users often reply to good comments, the best commenters may not partake in too many extended conversations with a single user. Finally, notice that network density is rather negatively correlated with expertise, as we expected in our hypothesis.

In terms of both importance and correlation, clustering coefficient, topical entropy and low degree fraction of successors seem to have relatively small relationships with the expertise of the users in the dataset. Unlike what we expected, topical entropy has a slight *positive* relationship with expertise on Reddit. This is contrary to the results of Adamic et al. Though for the most part it was not very important, we notice a slight bump in correlation between entropy and expertise using the Pearson metric in the cars Subreddit. Clustering coefficient appears to only be significantly relevant in CFB. We suspect that this may be because of the greater comment density present in the

Subreddit as CFB is substantially more popular than programming and cars. Low degree fraction, like the other two relatively insignificant measures, appears to be slightly more relevant in one topic (programming) than the other two Subreddits

6. Limitations and Future Work

6.1. Conclusions and Future Work

Through this survey of network structure, we managed to develop a procedure for inferring whether a user is in the top 10% of comment writers in an unstructured discussion forum with a precision of .304 and a recall of .526 in programming, with similar scores in two other Subreddits. This performance indicates high recall and good precision relative to the other social network-based algorithms. We explored a variety of social network features and topical features and their correlation with a user's expertise, discovering that responses to a comment – both in terms of quantity and topical content – has relatively strong correlations with a user being an expert in the community. We also found that sheer number of replies can be the most powerful feature measure in itself. We also compared our results with some other social network-based techniques for classifying a user's expertise level.

Future work may consider looking at some other linguistic features of a user's comment community. It may be useful, for instance, to implement a tool for sentiment analysis to characterize the positive or negative feelings of the community with respect to certain topics. Empirically, communities on Reddit tend to share a single, low variance opinion on a given topic. A potentially useful consideration would be to measure deviations from these opinions and observe their influence on the popularity of a given comment.

It may also be useful to place greater focus on the topical, structural and lingustic features belonging to all users that have replied to the users that will be classified. As we have seen in correlation and feature importance analysis, the response a user receives for his comments is a very important component of whether or not that user is an expert commenter, where expert is defined in terms of how the community receives the user's posts.

In this analysis, we revealed some interesting empirical trends. However our discussion focused largely on the what rather than the how. Future studies of social networks can perhaps look to explain these trends from a sociological viewpoint.

6.2. Limitation: Ground truth accuracy

In our tests, our algorithm demonstrated a consistent problem with precision in classification. A large portion of this issue may be because of the slightly arbitrary nature in which we defined a user as an "expert". We noticed that at an expert score cutoff of 10.2, some users that would by all means be considered an expert in terms of breadth of knowledge and Subreddit popularity were excluded from the ground truth expert set. For instance, one user with a comment score of 9.8 in programming was consistently "misclassified" as an expert according to our algorithm. This user turned out to in fact be quite an expert in reality and many of his posts were very well received.

We tried an alternative approach of obtaining ground truth data: namely, looking up a user's username for Twitter or Github handles to determine expertise from prior experience. It quickly became apparent, however, that a user with strong background may not contribute the most interesting or well-received posts. This can be for a variety of reasons. The user, for instance, may be inclined to make controversial statements outside of his domain of knowledge or perhaps partake in long conversational threads only relevant to a few users in the community. In the end, it became apparent that user score ranking was a much more reliable means of ground truth despite its tendency to exclude users. In future work, it may be beneficial to invest greater time in combining the manual and score-based approach in obtaining ground truth data.

6.3. Limitation: Number of examples explored

In our analysis, we focused our attention on two relatively small Subreddits and one larger one. Within these Subreddits, we also only focused our attention on a relatively small subset. Future analysis can perhaps seek to classify and analyze a larger number of Subreddits and individuals. It would be interesting to explore the features of the social network on a larger scale.

7. Acknowledgements

We would like to thank Professor Andrea LaPaugh for heading our seminar and advising this work. We would also like to thank Mehmet Basbug for providing guidance on machine learning techniques.

8. Appendix

8.1. Sample depth-5 Decision Tree Output

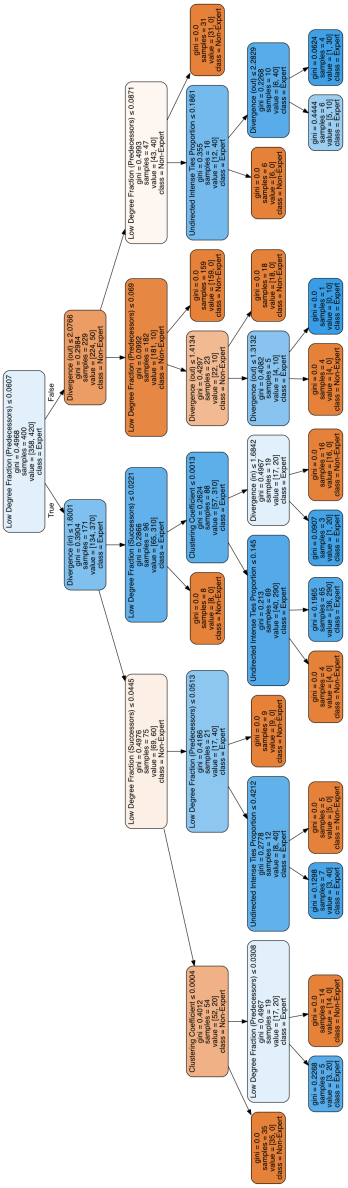

Figure 4: Example of a decision tree output from sci-kit learn. This particular tree is depth limited to 5. In our implementation, we restricted the depth to 8

References

[1] L. A. Adamic *et al.*, "Knowledge sharing and yahoo answers: Everyone knows something," in *International World Wide Web Conference*, 2008.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, 2003.

[3] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001. Available: http://dx.doi.org/10.1023/A:1010933404324

[4] C. Buntain and J. Golbeck, "Identifying social roles in reddit using network structure," in *International World Wide Web Conference*, 2014.

[5] A. Celikyilmaz, D. Hakkani-Tur, and G. Tur, "Lda based similarity modeling for question answering," in *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*, 2010.

[6] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[7] D. Fisher, M. Smith, and H. T. Welser, "You are who you talk to: Detecting roles in usenet newsgroups," in *Proceedings of the 39th Hawaii International Conference on System Sciences*, 2006.

[8] M. Forestier *et al.*, "Roles in social networks: methodologies and research issues," 2012.

[9] S. A. Golder and J. Donath, "Social roles in electronic communities," in *Association of Internet Researchers (AoIR) conference Internet Research 5.0*, 2004.

[10] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Pasadena, CA USA, Aug. 2008, pp. 11–15.

[11] T. J. Hazen, "Direct and latent modeling techniques for computing spoken document similarity," 2010.

[12] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, 1999.

[13] Y. Liu *et al.*, "Modeling class cohesion as mixtures of latent topics," 2009.

[14] A. McCallum, A. Corrada-Emmanuel, and X. Wang, "Topic and role discovery in social networks," 2007.

[15] H. Misra, O. Cappé, and F. Yvon, "Using lda to detect semantically incoherent documents," 2008.

[16] L. Page *et al.*, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Technical Report 1999-66, November 1999, previous number = SIDL-WP-1999-0120. Available: http://ilpubs.stanford.edu:8090/422/

[17] M. Pennacchiotti and A.-M. Popescu, "A machine learning approach to twitter user classification," in *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[18] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[19] M. Rosen-Zvi *et al.*, "The author-topic model for authors and documents," 2004.

[20] H. T. Welser *et al.*, "Visualizing the signatures of social roles in online discussion groups," *Journal of Social Structure*, 2007.

[21] J. Zhang, M. S. Ackerman, and L. Adamic, "Expertise networks in online communities: Structure and algorithms," in *International World Wide Web Conference*, 2007.