Analyzing StackExchange Expert Emergence Over Time
CS 224W Final Project Report

Ben-han Sung, Rebecca Wang, Arun Kulshreshtha

Introduction

The StackExchange network of question and answer websites has become one of the most popular ways of finding information online. Rather than sifting through hundreds of search results or trying to decipher difficult papers or articles in order to find the information they want, users can simply ask a question about any topic, and reasonably expect that a user more knowledgeable about that topic will respond with a clear and direct answer. Naturally, a system like this relies on having many knowledgeable and experienced users (so-called "experts") who can sustain the community with good answers.

In this project, we studied how expert users emerge on Q&A websites by analyzing the network properties and metadata associated with these users in data taken from StackExchange websites. By observing the evolution of HITS, PageRank, ELO, and Cumulative Aggregate Upvote (CAU) scores of expert users over the early part of their account lifetime, we were able to apply logistic regression and SVM classifiers to accurately predict whether a given user is likely to achieve expert status.

Prior work

There has been substantial work done on online Q&A communities and expert detection. In [1], Zhang et al. describe how analyzing a properties of a directed graph of users yields valuable insights such as that answers mostly come from a small group of "experts" who don't ask very many questions themselves, and that almost all network techniques were able to distinguish experts from normal users. By using simulated graphs to test hypotheses about experts' behavior on the forum, they also found that real experts tend to be willing to help all kinds of users. Similarly, in [4], Rode et al. describe how using graph-based features in several relevance propagation models can improve expert finding.

However, most of the existing literature does not perform temporal analysis. While other researchers have studied expert detection, their methodology applies only to graph models for a particular point in time, or to graph models that do not consider the time dimension at all. Pal et al. are one exception and they illustrate the potential for discovering new insights made possible by temporal analysis in [3]; however, limitations in their methodology give rise to one of the primary focuses of our project: using the evolution of network properties over time to understand the emergence and behavior of experts.

Data Collection

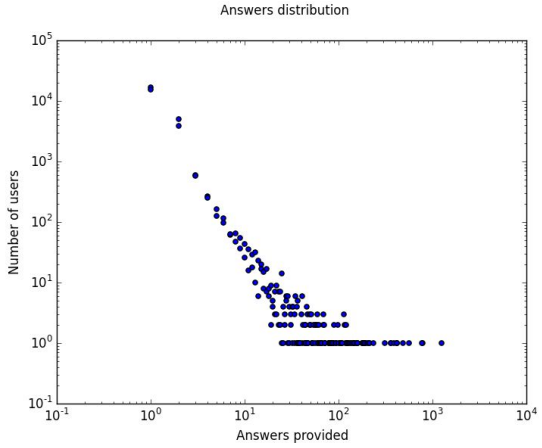
The dataset we are using for our project is the StackExchange Data Dump, an anonymized dump of the database tables from all of the websites in the StackExchange network of Q&A websites. This dataset is provided by StackExchange as a 27GB of compressed XML data under a Creative Commons license and expands to more than 100GB in size. In order to effectively mine the data, we bulk-loaded it into a PostgreSQL database so that we could manipulate it with SQL. From there, we queried the database to build network models using Python and SNAP.py.

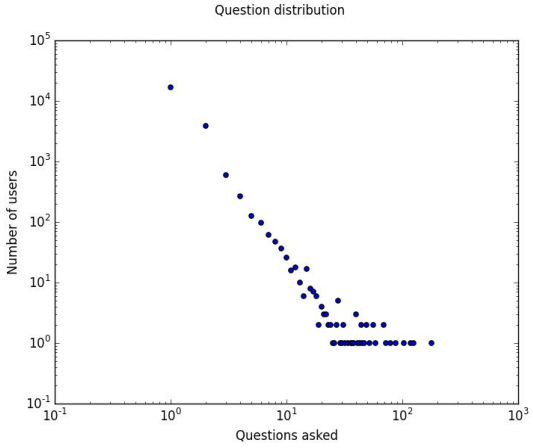
Because of the size of the dataset, it was not feasible to analyze all of the StackExchange communities within the amount of time we have allotted for this project. Even just the StackOverflow dataset, at 70GB, was too large for us to work with efficiently. In the end, we decided to work primarily with the small CS Stack Exchange community (cs.stackexchange.com) and scaled our results to the slightly larger Cooking Stack Exchange (cooking.stackexchange.com). The smaller dataset relaxed the time pressure and gave us

the flexibility to discard, revise, and regenerate graph models until we could arrive at the most meaningful results.

Dataset Overview and Statistics

The Cooking StackExchange dataset has tables with information on users, posts, post history, comments, badges, tags, and votes; it contains 22,061 users and 42,887 posts (includes both questions and answers). These posts range from July 2010 to March 2015. One observation we noticed (especially true in CS StackExchange, but also relevant in Cooking StackExchange) is that the number of posts is surprisingly small relative to what one would expect given the number of users. This led us to explore the distribution of questions and answers posted on the site, shown in the graphs below. It immediately became obvious that the activity of users is highly skewed: both the distribution of questions and answers very clearly follow a power law. 16,808 users have never asked a question, and 15,305 users have never answered a question, showing that the vast majority of users are completely inactive; this will affect the sample of users we choose to investigate.





Reputation Scores

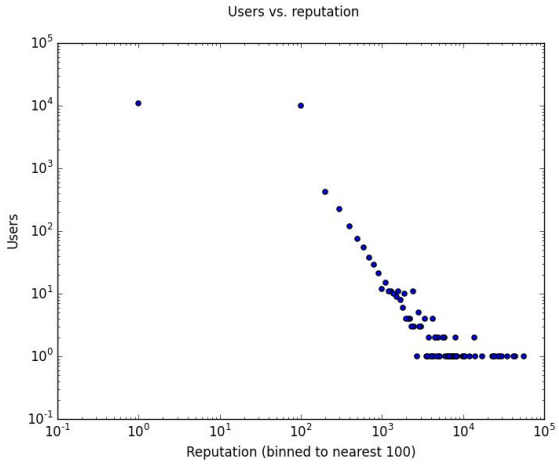
The canonical metric used to measure user expertise on StackExchange is reputation score. Each user has a reputation score that takes into account the upvotes on that user's questions and answers, the number of "accepted" answers and suggested edits, as well as several other factors[1]. For the sake of measuring expertise, the most important components are those related to post upvotes and accepted answers. Since these are the most heavily weighted components of user reputation scores, reputation appears to be a reasonable way to measure user expertise.

Given that it is the canonical way of measuring expertise on StackExchange, for the purposes of our analysis we will consider reputation scores to be the "ground truth" for expertise, and it will be the metric against which we compare the others we propose. However, it is important to remember that reputation scores do have some limitations, which will be discussed later in our analysis.

In particular, in our analysis we will refer to the top 1% as experts, and the remaining users as non-experts. During our analysis of expertise metrics, we will observe how each metric varies over time for these groups.

[1] "What is Reputation? How do I earn and lose it?" (http://stackoverflow.com/help/whats-reputation).

An interesting preliminary observation we made is that there is very sharp skew of reputation among users. The highest three reputations are 55871, 44069, and 41714, but everyone after the first 17 has under 10,000 reputation points and a score of 969 qualifies for the top 1% of all users. On the other end of the spectrum, 6225 users have only 1 reputation point (which is default for new users).



Plotting the distribution of reputation scores shows that reputation follows a power law. This suggests that there is "preferential attachment" when it comes to distributing reputation points, i.e. users with high reputation tend to gain more reputation. Furthermore, we conjecture that there is a distinct time advantage for users who join the community earlier, since people who earn some reputation points early will also gain the bulk of reputation points later under a preferential attachment model. To account for this time advantage, we will therefore need to apply some normalization strategies in our expertise measures; this is discussed later.

Sampling Users

Before we explore various metrics, we first consider the subset of users that we will apply these metrics to. While we consider the top 1% of users by reputation to be experts, for the rest of this paper we consider the sample of only the top 12 experts. For non-experts, we take a random sample, which consists of 500 users randomly selected from the population of users that are not experts, and have answered at least one question (as mentioned previously, this is necessary because most of the users in the community have had no activity, and our results are more meaningful if we only consider active users).

Later on, when we introduce ELO ratings, we also define a "loser" sample as a measure of comparison against experts and a random sample of non-experts. These are the users that consistently contribute posts of poor quality (defined more specifically as users whose ELO ratings have gone below 1490 at some point in time).

Analyzing Expertise Over Time

An important part of our analysis involves understanding how various measures of user expertise evolve over time in our dataset. We can't directly do this with reputation scores since we do not have reputation data over time in our data set. As such, we will consider several other time-based methods of measuring expertise ("expertise metrics") that we can derive with the dataset. For each expertise metric we consider, we will calculate that metric's value with respect to the cumulative data in our dataset up to and including that time step. This way, for any given user, we can see how that metric evolves over time.

In our analysis, we initially modeled time as a discrete series of monthly time-steps. However, this creates a problem when aggregating the expertise over time statistics for multiple users. In particular, suppose that we had two expert users, one of whom joined the site a year after the other. If we simply averaged their expertise scores at each point in time, this would change the shape of the averaged expertise curve, even if the shape of each user's expertise curve were the same. In order to meaningfully compare these metrics, we need to normalize each user's timeline of reputation scores so that we are comparing user's scores at the same relative point in each user's account lifetime.

To do this, we propose the following normalization scheme (inspired by the user lifecycle described in [4]): For each user, we will consider the user's entire post history. We will identify the point in time the user had made a certain fraction of the total number of posts that user ever made in our dataset. We call these points in times "lifecycle percentiles" for each user. For each expertise metric, we construct a curve of that metric over time by sampling it at regular percentile intervals. This makes the expertise over time curves independent of the actual account creation dates and activity times of the user in question, allowing different user's curves to be directly compared and averaged.

Modeling the Problem

StackExchange is a question and answer website in which users ask questions and answer other user's question. This structure of the data can be modelled as a network. In particular, it can be viewed as a bipartite graph of user and posts, with an edge between each user and each of the posts that user has replied to. This is the basic model of the dataset that we will consider in our analysis.

However, some techniques and algorithms do not work well with bipartite graphs. For expertise metrics that involve such techniques, we will consider a unipartite projection of this bipartite network. In particular, one useful unipartite model is a directed graph in which nodes are users, with an edge from user A to user B if used B answered a question posted by user A. Using these models, we will explore applying a variety of network analysis techniques to measure user expertise over time.

In the following sections, we will discuss the several metrics that we found to be useful for measuring expertise, including some, like HITS and PageRank, which were taken from other areas of network analysis, and others, such as CAU and ELO, which are new metrics we propose for expert detection.

Cumulative Aggregate Upvotes (CAU)

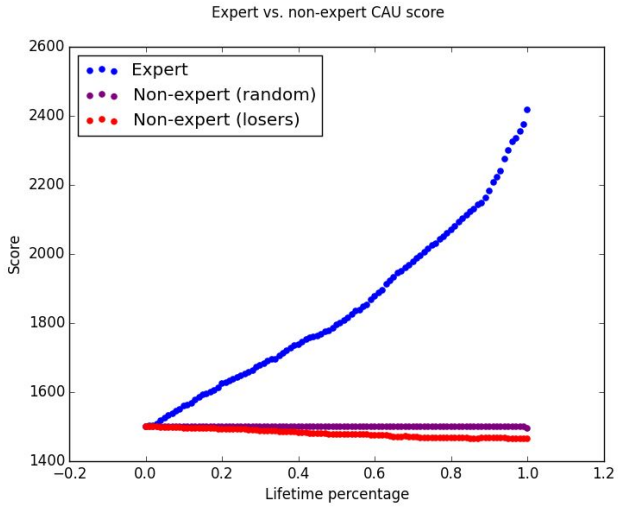
Another one of the initial metrics we tried was simply summing up the total number of upvotes each user had received on each of their posts up to a certain time. We call this metric Cumulative Aggregate Upvotes, or CAU. However, this extremely simplistic way of just summing up all of the cumulative upvotes is not particularly insightful; it will simply monotonically increase for all active users over time. It also only considers users in isolation, and does not take advantage of the network structure of the data. However, it does seem that upvotes should be indicative of expertise. As such, we propose a modified CAU metric that pits nodes in the network against one another when aggregating upvote counts.

The algorithm for computing modified CAU scores proceeds as a game over a bipartite network representation of the StackExchange dataset, as discussed previously. Nodes are either users or questions, and edges exist between users and the questions they answered. Each of these edges has an additional property: the time at which the edge was created, i.e. the question was answered.

Each user node is initialized with a CAU score of 1500. The algorithm proceeds by iterating over each edge of the bipartite graph in order of the edge creation time. At each iteration, the algorithm considers all of the other edges connected to the same question node as the current edge, whose creations times are less than that of the current edge. The user node for the current edge then plays "a game" with each of the user nodes for the other edges. The CAU scores for all nodes are then updated using the following procedure:

$$S_{player} = score\ of\ player's\ post\ (upvotes\ -\ downvotes)$$
$$S_{opponent} = score\ of\ opponent's\ post\ (upvotes\ -\ downvotes)$$
$$CAU := CAU_{player} + (S_{player} - S_{opponent})$$

Effectively, instead of aggregating the upvotes for a single user, this metric aggregates the amount by which each user's answer was "better" than the other answers to the same post, cumulatively across all posts up until a given point in time. This means that some users (namely, those that consistently "win") will have scores that consistently increase, and others (namely, those that consistently "lose") will consistently decrease. A plot of CAU scores over time is shown below:



From this plot, we can see that the experts appear to consistently "win," often by a large margin, causing their average CAU score to increase over time. From this, it appears that a consistently increasing CAU score is indicative of users who have high reputation. Conversely, for a random sample non-expert users, CAU remains a flat line, which is expected since we'd expect normal users to have both ups and downs in their CAU scores that would even out on average. Additionally, on the plot, we show a third set of users (the "loser" sample), namely, users whose scores consistently decreased. These users turned out to be all non-experts, which is expected since if they had been averaged into the experts, we would probably not see the consistent increase in the plot among experts.

One problem with CAU scores is that it becomes difficult to compare experts to one another at different lifetime percentiles. While CAU seems to be an effective way of distinguishing experts from non-experts, even at different lifetime percentiles, since experts' CAU scores seem to always increase, CAU always gives an advantage to older expert accounts. We will consider some other metrics in an attempt to resolve this problem.

PageRank and HITS

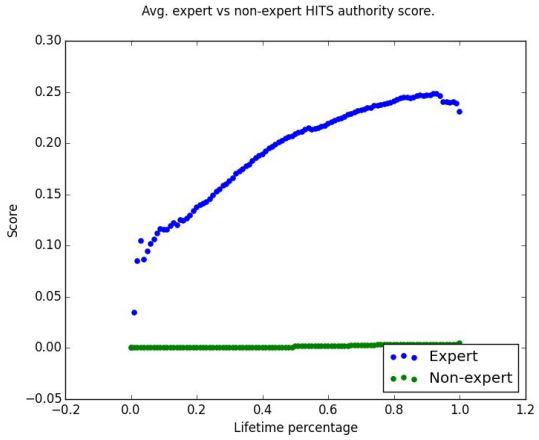
PageRank and HITS are two algorithms that were originally developed for ranking web search results. These algorithms attempt to use the link structure of the web to determine which web pages are

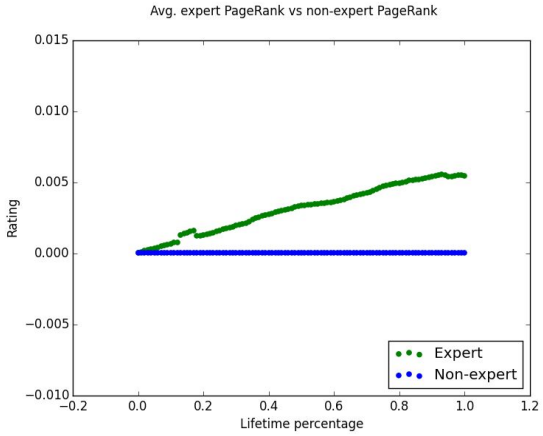
The PageRank algorithm, developed by Larry Page and Sergey Brin at Stanford University, models the probability that a user randomly search the web will visit a given web page. It proceeds by initially assigning each node an equal probability, and then repeatedly updating that probability by propagating the distributing the PageRank of each node to each node that the node links to. This is repeated until convergence[6].

The HITS algorithm, developed by Jon Kleinberg at Cornell University, operates on a directed graph, assigning each node a "hub" score and an "authority" score. These scores are defined in terms of each other, with the hub score consisting of the sum of the authority scores for all of the nodes a given node links to, while the authority score is the sum of the hub scores of all of the nodes that link to a given node. The algorithm proceeds by alternatively computing the hub and authorities scores for all nodes on the graph until convergence[7].

Both of these algorithms essentially try to find the nodes in the graph with the most incoming links from other authoritative nodes, meaning that nodes with high PageRank or high HITS authority score should be considered the most authoritative nodes in the graph.

We can apply these algorithms to find the most authoritative users in our dataset by taking our original bipartite graph, and creating a unipartite projection, in which the nodes are users and node A has an edge to node B if node B answered a question asked by node A. We performed this procedure on our dataset, and once again plotted the average score across user account lifetimes for both experts and non-experts. The results are shown below:





From the plots, we can see that both of these metrics show a distinction between experts and non-experts. This is an interesting result because both of these metrics rely solely on the network structure of the graph, and do not consider upvotes (or any of the other components of the reputation scores) at all.

Additionally, unlike CAU, it appears that as time passes and the number of edges in the dataset grows, the scores appear to start to level off. This is a nice property, because it seems to indicate that rather than increasingly boundlessly, these metrics could conceivably converge on a score that accurately reflects the "expertise" of the user, without constantly increasing as time goes on.

However, one big caveat for these metrics is that they are not as amenable to temporal analysis as CAU, because they require creating a new unipartite projection of a different subset of the graph at each time slice and then running an iterative algorithm on that graph. Needless to say, this becomes inefficient when attempting to sample at very fine time granularities, which makes these algorithms problematic for doing this kind of temporal analysis in practice.

ELO

ELO is a classic game-based expertise measure first created by Arpad Elo for chess and then adopted in many other places, from board games to Major League Baseball. We can fit ELO to Stack Exchange data by using the same bipartite graph construction as with CAU and by using the same "game' approach and the same playing order. The main difference will be in the update rule for the ELO score after each game.

Again, the user whose answer receives a higher score, where score is the sum of upvotes and downvotes, is considered to be the winner of the game. All users start with a default rating of 1500, which is updated after each game according to the following formula[2]:

$$Outcome = \begin{array}{ll} 1.0 & \textit{if win} \\ 0.5 & \textit{if draw} \\ 0.0 & \textit{if loss} \end{array}$$

$$K = \begin{array}{ll} 8 & \textit{if total games played} < 100 \\ 1 & \textit{if opponent's total games played} < 100 \\ 4 & \textit{otherwise} \end{array}$$

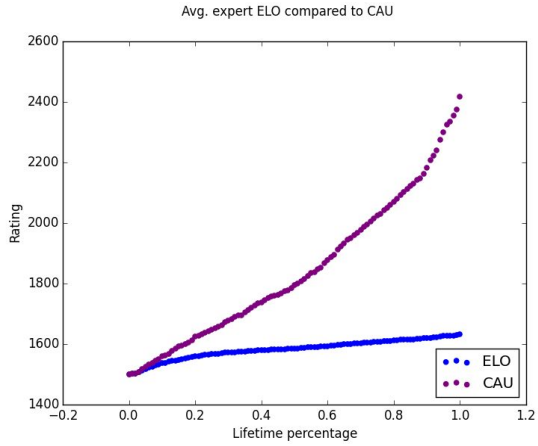
$$Diff = ELO_{player} - ELO_{opponent}$$
$$Expected = \frac{1}{1+10^{-Diff/400}}$$
$$ELO_{player} := ELO_{player} + K(Outcome - Expected)$$

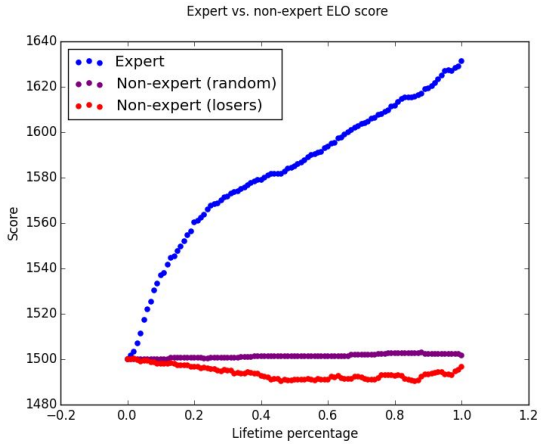
Considering the dataset as a series of games in this way generates n^2 datapoints (games) for updating a user's expertise rating, up from a set of n posts, therefore amplifying the available granularity of a user's expertise rating. However, the most important of ELO, which makes it different from CAU, and which arises from the fact that the update rule takes into account a game's expected outcome, is that it converges. This is evident in the plots below[3].

---

[2] Adopted from StackRating. (http://stackrating.com/about)

[3] We observed that convergence property of ELO was be much more obvious on a larger dataset, such as the Android Stack Exchange. If the trend is not clear enough here, it is probably the case that we don't have enough time data and the point of convergence occurs after the end of dataset. Nevertheless, you can still see that the ELO score approaches some asymptotic value whereas CAU continues growing.





After a player plays enough games, their ELO rating should eventually converge to a number that is interpreted to be their skill level. In the context of this project then, StackExchange users who answer enough questions will receive an ELO rating representative of their expertise. The expert vs. non-expert plot on the right demonstrates this correspondence - namely, that experts as identified by reputation also have high ELO, whereas randomly-sampled non-experts have ELO scores remaining around the baseline of 1500, as expected.

Because of the convergence property, ELO ratings for experts can be more meaningfully compared than CAU scores that will continue to grow as a function of time.

Machine Learning Results

In order to test the usefulness of our metrics, we attempted to use them to detect expert users (in terms of reputation score at the end of the dataset timeline) by using only metrics from the first half of each user's account lifetimes.

Our feature vectors consisted of the CAU and ELO scores for each user sampled at the 10th, 20th, 30th, 40th, and 50th percentiles in a user's account lifetime. We wanted to classify users on whether or not their ultimate reputation score at the 100th lifetime percentile would place them in the expert category or not. We intentionally omitted PageRank and HITS scores from our feature vectors, as computing them for many users at many given lifetime percentiles proved too computationally intensive.

We trained a logistic regression classifier and an SVM with a Gaussian kernel on our feature vectors. We trained the classifiers on a training set of 360 randomly-chosen users, 110 of whom were experts, and 250 of whom were non-experts. We then tested the classifiers on another set of 360 randomly-chosen users who were not in the training set, again with 110 experts and 250 non-experts.

Logistic regression correctly classified 301 out of 360 users, giving it 83.6% accuracy. The precision and recall values were 0.846 and 0.529 respectively. This indicates that of the misclassified users, most were non-experts incorrectly classified as experts. It is understandable that this would happen, as our definition for experts imposed a somewhat arbitrary cutoff that several users were probably close to but did not quite reach. Given that we trained on a random sample of the users, it is understandable that any users "close" to the cutoff would end up on the wrong side of the classifier's decision boundary.

We repeated the same process using an SVM with a Gaussian kernel. The SVM correctly classified 354/360 user, an accuracy of 0.983, with a precision of 1.0, and a recall of 0.951. We believe that this indicates that our data set did have a clear decision boundary, but this decision boundary was nonlinear, hence the better performance when using a nonlinear kernel with the SVM.

Overall, applying machine learned showed that our metrics could be a useful part of an expert user identification system. However, given the plots of expert vs non-expert scores on our metrics, it seems like using machine learning might be overkill, as it seems like it would be sufficient to just compare the magnitude of any of our metrics between a known expert user and non-expert user to classify other users.

Conclusion

We experimented with several metrics for identifying experts in our StackExchange dataset, and observed how those metrics changed over time. In particular, we modelled our data as either a bipartite graph, or as a unipartite projection of a bipartite graph, and applied both well-known algorithms like HITS and PageRanks, as well as metrics of our own such as CAU and ELO scores. What we found was that for all of these metrics, there is a very clear distinction between experts and non-experts, making all of these metrics suitable for expert identification purposes.

One caveat that we found was that in temporal analysis, in order to meaningfully compare users, it is important to normalize the times at which we draw samples for each user to correspond to the same stages in the user's account lifetime. This is because some of our metrics, like CAU, always increase, and therefore one needs to start from the same starting time in order to meaningfully compare CAU scores. This means that metrics that tend to converge, such as ELO, are probably more useful for temporal expertise analysis, as such metrics will reach a somewhat stable value that allow meaningful comparisons even without time normalization. As such, possible future work may involve looking into other such expertise metrics that converge over time, especially ones that converge quickly and obviously.

References

[1] Jun Zhang, Mark S. Ackerman and Lada Adamic. *Expertise Networks in Online Communities: Structure and Algorithms*. 2007.

[2] Aditya Pal, Shuo Chang and Joseph A. Konstan. *Evolution of Experts in Question Answering Communities*. 2012.

[3] Henning Rode, Pavel Serdyukov, Djoerd Hiemstra, and Hugo Zaragoza. *Entity Ranking on Graphs: Studies on Expert Finding.* 2007.

[4] Cristian Danescu-Niculescu-Mizil, Robert West, Dan Jurafsky, Jure Leskovec, Christopher Potts. *No Country for Old Members: User Lifecycle and Linguistic Change in Online Communities.* 2013.

[5] Andreas Lundblad. *StackRating: An ELO-based rating system for StackOverflow.* 2015.

[6] Larry Page and Sergey Brin. *The PageRank Citation Ranking: Bringing Order to the Web*. 1998.

[7] Jon M. Kleinberg. *Authoritative Sources in a Hyperlinked Environment*. 1999.