UNIVERSITY OF AMSTERDAM

An Agent-Based Model to Study Stack Overflow Interactions

February 10, 2023

Students:
Aaron De Clercq
aaron.de.clercq@student.uva.nl
UvAnetID 14483610

Jing Hu
jing.hu2@student.uva.nl
UvAnetID 14530252

Aizhan Shagadatova
aizhan.shagadatova@student.uva.nl
UvAnetID 14685531

Lijing Luo
lijing.luo@student.uva.nl
UvAnetID 13817426

Xinhai Li
xinhai.li@student.uva.nl
UvAnetID 14305623

Lecturer:
Debraj Roy

Course:
Agent-based Modelling

Abstract

Online Q&A communities are now an essential tool for developers, students, and managers learning programming and problem-solving on a daily basis. In this paper, we propose an agent-based model (ABM) to simulate the behaviour of Stack Overflow users under the influence of the platform's reward and sanction system and personal preferences. The model we build characterises the distribution of users' upvotes and reputations and their sensitivity. With this model, we aim to discover the correlation between users' information, such as upvoting and reputation, and power law distribution, and to study the exponents under the power law distribution. In conclusion, users' upvote behaviour and reputations were found to follow a power law distribution, and the analysis revealed that the upvote bias was considered the most sensitive parameter, followed by the mean of the normal distribution of activity and upvote probabilities. However, regression analysis shows that the model does not consistently achieve a close fit to the actual data.

1 Introduction

Social Q&A communities are social media platforms that focus on user knowledge exchange and have become an important platform for users to ask questions, seek information, and share knowledge. Compared with traditional information acquisition channels, social Q&A communities have the advantages of timeliness, diversity, and personalisation, and play an important role in meeting users' information needs, expanding their social circles, and increasing exposure to their professional fields. Stack Overflow was founded in 2008 and has grown into the world's largest English-language IT technical Q&A community with over 11 million users and over 250 million average monthly visitors [1].

Knowledge contribution behaviour of users in Q&A communities is very important and is a prerequisite for users to obtain the information they need in the communities. Wang et al [2] proposed a method to assess users' knowledge diffusion ability and promote active participation in Q&A communities by finding core users with strong knowledge diffusion ability. Jin et al [3] showed that users' self-presentation, other members' perceptions and social learning opportunities have positive effects on users' knowledge contribution behaviour. Liu et al [4], on the other hand, found that users' factors such as participation, interest and relatedness have a significant impact on users' willingness to answer in social Q&A communities. Stack Overflow adopts reputation value as the incentive mechanism of the community in its functional design and community mechanism and adopts a voting mechanism to assign reputation value. Users can choose to vote on questions and answers in the community to assist the community administrator to complete the evaluation and review of Q&A quality [5].

Most current Stack Overflow user and topic studies are empirical comparative studies based on actual data, ignoring individual autonomy and heterogeneity, and failing to accurately describe the micro-interaction behaviours of web users as well as the Q&A process and voting trends among agents. Compared with traditional empirical studies, this paper further analyses the micro-interaction behaviours among Agents on Stack Overflow in-depth and explores whether a simple Agent-based model of question-and-answer upvoting mechanism can replicate two stylized facts: 1. The probability distribution of users' upvoting on Stack Overflow is a power-law distribution [6]; 2. The probability distribution of reputation values on Stack Overflow is a power-law distribution [7]. By using ABM to model question-answer upvoting interactions among users, this paper allows us to explore the underlying upvoting feedback mechanisms that drive these stylized facts and illuminate the complex dynamics of the micro-interaction behaviours among users on Stack Overflow.

We conducted experiments to investigate the effect of the following six parameters: user ask probability, user answer probability, user upvote probability, user active probability, user upvote threshold and user upvote bias on the exponent in the power law distribution in our model.

The rest of this paper is organized as follows. In Section 2, we provide an overview of the model, its design concepts, and experimental details, using the ODD (Overview, Design Concepts, Details) protocol. In Section 3, we describe the research methodology for system simulation, including any tools, techniques, or methods used. In Section 4, we present the experimental results and discuss them, including validation and sensitivity analysis. Finally, in Section 5, we conclude with a summary of the key findings and highlight the contribution of the study to the research field, including any implications for future work. The code we wrote to perform all the experiments and produce the results and graphs in this report can be found on this GitHub repository.

2 Model description

2.1 Overview

2.1.1 Purpose

The purpose of our project is threefold. Firstly, we aim to recreate the Stack Overflow interaction network by modelling interactions between the users based on the known emergent behaviour. It is known from the empirical studies mentioned in the introduction that probability distributions of the reputation and the upvoting follow power law distribution. This brings us to our second aim which is to perform sensitivity analysis to understand under what conditions these probability distributions are not observed. This can be important due to the stability properties of scale-free networks against random failures [8]. This way Stack Overflow networks are more stable against users' random dropout. Lastly, we want to compare reputation and upvoting in bigger and smaller size communities. In Stack Overflow the communities are formed based on the users' interests such as specific programming languages or some technologies. Some topics are more popular than others. This affects the size of the corresponding community. Therefore, it is of interest to explore patterns in reputation and upvoting across different-sized communities.

2.1.2 Entities, state variables, and scales

The entities in our model are agents representing Stack Overflow users. The agents are heterogeneous in their attributes such as a probability of answering, a probability of asking, a probability of upvoting an answer or a question, and a probability of being active. These are the main attributes that define a user's activity.

The spatial structure is modelled in two layers. The first layer of space is based on Stack Overflow tags and by this we create non-connected communities to aggregate users according to their interests. The second spatial structure is temporal and involves the probability of being active, which indicates whether a user is online. The probability of being active affects questions the user can see, and in turn answer and upvote.

This leads to the following attributes: a list of visible questions and number of answers, a list of asked questions and the number of asked questions, number of upvotes to questions and answers. Depending on the number of upvotes to the user's questions and answers, the user can gain reputation. Similar to Stack Overflow, the starting reputation is equal to 1. This follows with the attribute called upvote bias that determines the minimum number of upvotes received to be encouraged to ask more questions. Additionally, each agent is assigned a Stack Overflow tag which represents a user's interests. In the model proposed here, the 12 most popular tags will be used.

There are also passive entities in our model, which are a question and an answer. The attributes of a question are a tag, a number of upvotes and a list of answers for this question. Regarding an answer's attributes, there is a number of upvotes. The table with all parameters explored in the model is provided.

The proposed model is not subject to exogenous factors.

2.2 Process overview and scheduling

A single time step of a user is split into two major parts. The first part is about the interaction between users and is visualised in Figure 1. Firstly, a user can ask a question and make it visible to active users with the same tag. Next, the user visits their visible questions and for each question can decide to upvote the question. If he gives an upvote,

Table 1: Parameter settings

Parameter name	Type	Default value	Range (SA)
upvote threshold	int	15	[0, 40]
upvote bias	int	12	[0, 40]
$\mu(P_{ask})$	float	0.5	[0, 1]
$\mu(P_{answer})$	float	0.5	[0, 1]
$\mu(P_{upvote})$	float	0.5	[0, 1]
$\mu(P_{active})$	float	0.5	[0, 1]
$\sigma(P_{ask})$	float	0.25	[0.01, 1]
$\sigma(P_{answer})$	float	0.25	[0.01, 1]
$\sigma(P_{upvote})$	float	0.25	-
$\sigma(P_{active})$	float	0.25	-

the owner of the question gains 10 reputation points. Then, the user makes a decision to answer the question. If the decision is negative, then he has the choice to upvote already existing answers to this question. Upvoting an answer adds 10 points to the reputation of the user that gave the answer.

In the second part of the time step, there is a feedback mechanism that updates the probabilities of asking, answering and being active. This mechanism is visualised in Figure 2. A user will evaluate their own questions by checking the number of upvotes and the probability of asking and being active is updated using the following equations.

$$x = log\left(\frac{p}{1-p}\right) + 0.1 * (n_{upvotes} - bias) \tag{1}$$

$$p = \frac{1}{1+e^{-x}} \tag{2}$$

Next, we loop over all the answers and update the probability of answering and being active of the user that gave the answer in the same way. Once all the answers are checked, the person that gave the answer with the most upvotes gains another 15 reputation points.

2.3 Design Concepts

2.3.1 Theoretical and Empirical Background

Stack Overflow is an open community-driven Q&A platform. To incentivize questions and answer there is a reputation system in place. Receiving an upvote on a question or an answer increases reputation. In contrast, getting downvotes or downvoting decreases one's reputation. This system ensures the quality of the platform and has been one of the catalysers of the success of Stack Overflow as a biggest community for developers [9]. The incentive/reputation system made a basis of our model.

In addition to this, we modelled increasing activity if a user receives upvotes. This concept is based on theories of reward-driven behaviour. In psychology, it is believed that humans exhibit increased engagement in response to rewarding stimuli [10]. We modelled this behavioural change by increasing the probability of asking, or answering, or being active with every upvote the user receives. This rewarding-based behavioural change has been further elaborated by addition of upvote bias to the model. The upvote bias is a number of minimum upvotes to change one's behaviour. Additionally, a single answer can

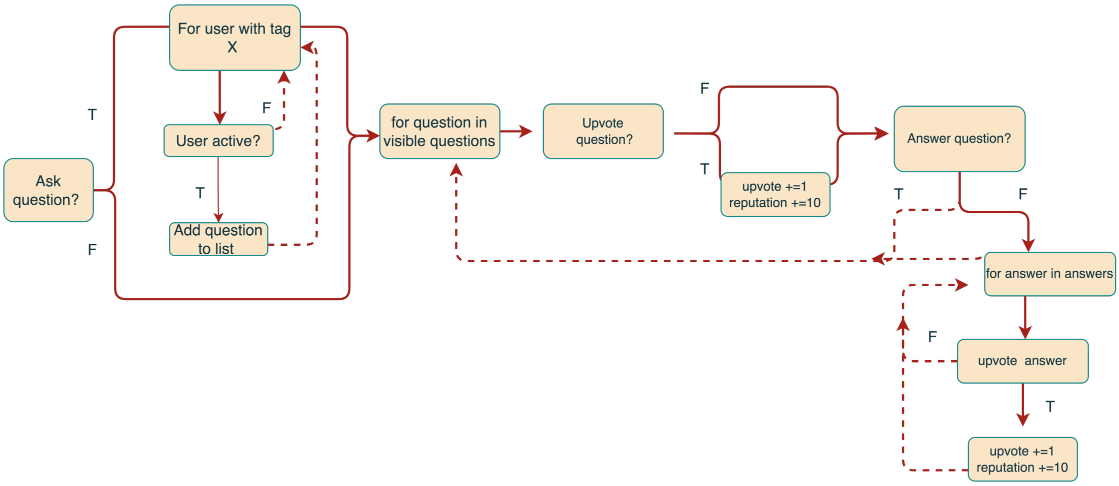

Figure 1: Diagram of the model. User's steps in one simulation step from left to right. The user can ask a question, make it visible for other online users with the same Stackoverflow tag. Next, the user can upvote a question from his list of visible questions. Then he can answer the question. Lastly, he can upvote the aswers.

be marked as 'accepted' by the person that asked the question, which can be interpreted as the answer that was useful to the asker. The user whose answer is accepted gains an additional 15 reputation points. To keep it simple, the answer that received the maximum number of upvotes is considered the accepted answer in our model.

To differentiate between good and bad questions/answers, you can downvote them. However, downvoting removes one reputation point from the user who downvotes and removes two reputation points from the owner of the post. In our model, the downvoting system is not modelled explicitly. However, we have a simple analogue arrangement. If there are no upvotes on the post, then the owner of the post automatically loses two points.

To add complexity and address the research question we modelled communities within the platform based on the user's interests. Currently, there are 60000 tags registered on Stack Overflow [11]. However, to keep the model simple and computationally feasible, we only take the 12 most popular tags to form communities. The relative frequency of the different topics is obtained by checking how many current questions there are about a topic compared to the total number of current questions on Stack Overflow. When normalized, this yield an empirical probability density function of the 12 most popular communities on Stack Overflow.

It is of particular interest to build agent-based model to observe emerging behaviour using simple rules described before. To consider emerging behaviour, we can look at empirical studies. According to Wang et al., 75% of users only ask one question, 65% -answer only one question, and only 8% - answer more than five questions [12]. Furthermore, the study has shown that 0.46% of users have a reputation of greater than 5000. Most Stack Overflow questions are answered by very few people. Another study exploring degree distribution of users has shown that most Stack Overflow users have small degrees with few users having large degrees [6].

2.3.2 Individual Decision-Making

As was mentioned previously, agents in the model can act by asking a question, giving

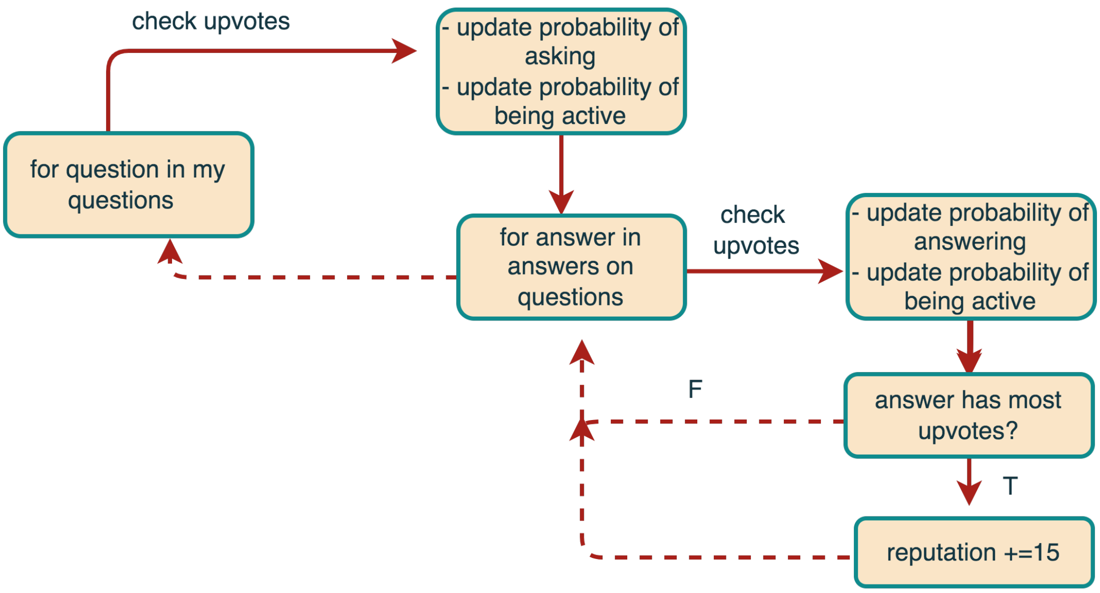

Figure 2: The feedback mechanism in the model in one simulation step. Upvotes on a question or an answer increase the probability of asking and probability of answering respectively, and probability of being active. Receiving maximum upvotes on an answer also increases the reputation.

an answer, upvoting and being active. These activities depend on probabilities drawn from different distributions. Agents decide to act if a probability is smaller than a random number drawn from a uniform distribution.

2.3.3 Learning

Learning is not considered in the model

2.3.4 Individual Sensing

Sensing is not considered in the model

2.3.5 Individual Prediction

Individual Prediction is not considered in the model

2.3.6 Interaction

Interaction is one of the most significant features of the Agent-Based Model, it builds up the complex system, and provides a way of understanding how individuals' behaviour impact the system level lead to the emergent phenomena and collective results. In the model, agents can interact with other agents indirectly through asking a question, giving an answer or upvoting.

2.3.7 Collectives

The formation of aggregates are imposed in the model. Those aggregates represents 12

Stack Overflow tags. The tags are assigned to each user with the probability according to a popularity on the Stack Overflow website. This way the most popular tags have higher probability to be assigned. As a results communities evolve. In our model communities are not connected.

2.3.8 Heterogeneity

Each Stack overflow user is different. In the model, users are different in asking, answering, and upvoting answers and questions. For example, the purpose of using the Q&A function of the platform is different for everyone, some are looking for answers, some are answering questions for others, and some just want to increase the reputation value. These differences generate heterogeneity. Furthermore, the feedback mechanism is also heterogeneous as the change in probability for a user depends on the local interactions of that user. In addition, agents have different interests, so this difference is reflected in their tags.

2.3.9 Stochasticity

The parameters or attributes assigned to agents in the model to model their interactions are generated stochastically. In the model, the decision for a user to perform a certain action is probabilistic, which means that their actions can be different in every time step. For instance, in some time steps a user will ask a question and sometimes the user does not decide to ask one. Due to this stochastic nature of the model, no two simulations will be exactly the same and multiple reruns of the simulation are necessary to obtain statistical significance.

2.3.10 Observation

As our aim is to reproduce the distributions about the number of upvotes given by users and their reputation, we need to keep track of these properties for every user. The reputation is one of the user's attributes and additionally we store a counter in every user that keeps track of the number of upvotes a user has given. During the simulation, we update the reputation of the user every time it changes and every time the user upvotes a question or answer, we increase the counter by one. At the end of the simulation, we can easily extract this data by looping over all the users. From this data, we can construct the distributions and calculate the exponent of the power law. This exponent is the main output variable used during the testing and sensitivity analysis.

To get an idea of the overall activity on the site, we also kept track of the total number of questions and answers during the simulation. This was done by having a list in the model and adding every question to the list once it is generated by a user. Because every question contains a list with all the answers to that question, we immediately know the total number of answers as well.

2.4 Details

2.4.1 Implementation Details

The entire network is represented as a class and one of its attributes is a list with all the users in the system. Additionally, it stores a two-dimensional array with information about how the users are divided over the different communities. In the array, there is a list for every community that contains the IDs of the users in that community. Based on the empirical probability distribution of the tags, we store the corresponding cumulative distribution as well because this can be used to determine the tag of newly created users during the simulation. The mean and standard deviation of the distribution from which the four probabilities are sampled is also stored in a list. The network contains a simple time step function where it iterates over all the users in the list sorted by their activity and executes the time step function of every user. The flowchart in Figure 1 and 2 is a good representation of how the time step for a user is implemented. The full code of the model can be found in this GitHub repository.

2.4.2 Initialisation

At the initialisation of the network, there are no users in the network yet. There is just the option of setting the parameters of the model and indicate how many users should be added during every time step. For our experiments, we added 250 new users every time step and let the simulation run for 20 timesteps. The default settings with which the model is initialized are presented in Table 1.

At every time step, there are new users initialised. During this initialisation, the user is given a tag which is determined by the empirical cumulative distribution function. Additionally, the probability of asking, answering, upvoting and being active of the user are determined by sampling them from a distribution. The default distribution is a normal distribution with a mean of 0.5 and a standard deviation of 0.25. For this sampling, the truncnorm function from the Python package scipy.stats is used because this truncates the distribution between 0 and 1. The code allows changing both the mean and standard deviation of this distribution as well as changing to a uniform or exponential distribution. For the sampling from the exponential distribution, we used the truncexp function from the Python package scipy.stats.

3 Methodology

The main part of the model validation is trying to reproduce the power law distribution of the number of upvotes given by users. The number of given upvotes is stored during the simulation in the user entity, so these values can be easily extracted once the simulation is finished. To get a cleaner graph, we create bins of size 5 when calculating the probability density function of the number of upvotes. In the case of a power law distribution, the exponent can be calculated by plotting the probability distribution on a log-log scale and performing a linear fit. This linear fitting is done in this study using the Python package scikit-learn. The exponent of the reputation distribution can be calculated in a similar way, but because the range of possible values is significantly higher compared to the number of upvotes, we used bins of size 125.

For the development of the model, we started with a basic model that had only some simplified Stack Overflow dynamics. In this model, there was only the gaining of 10 reputation points for receiving an upvote on a given answer or question, the gaining of 15 reputation points for giving the most upvoted answer and only users with a reputation of 15 and above are allowed to upvote. Once a version of the model was finished, we checked how the distribution of the number of upvotes looked and added a layer of complexity to the model if it did not yet follow a power law.

To test the effect of the input parameters on the model outcome, we performed a local and global sensitivity analysis. In the local sensitivity analysis, we selected the upvote threshold, upvote bias and the mean of the normal distributions from which the four probabilities were sampled as the varying input parameters. These factors were varied one at a time (OFAT) and the range over which they were varied is indicated in Table 1. Over this range, we took 15 sample points and let the model run with all the other parameters in their default setting. Due to the stochasticity of the model, 10 repetitions were done for every run. The output parameters that were used in this analysis were the exponent of the upvote distribution and the reputation distribution. Additionally, we also kept track of the total number of questions and answers as an outcome variable to check the overall interactivity during the simulation.

For the global sensitivity analysis, we used the Sobol sensitivity analysis. In this instance, the selected input parameters were the upvote bias, the mean of the normal distributions from which the probability of upvoting and being active are sampled, and the standard deviation of the normal distributions from which the probability of asking and answering are sampled. For the estimation of the first-order and total-order sensitivity indices of the parameters, the Python package SALib was used. With N equal to 512 and 5 input parameters (d), N(d + 2) = 3584 parameter settings were generated by SALib. Using every parameter setting, the model was run for 8 repetitions and the output variables were the same ones as in the local sensitivity analysis.

To look at how the number of upvotes given by users and their reputation is divided over the different-sized communities, we can create a boxplot of these measures for every community separately.

4 Results and discussion

4.1 Model development

Figure 3 shows the distribution of the number of upvotes given by users from different versions of the model. With the first version where there were only some basic Stack Overflow dynamics, we can see that there is a portion of the users that gives a very low amount of upvotes indicated by the red dot in the top left corner. This is the result of the upvote threshold that exists on Stack Overflow. These users are primarily users that ask or answer almost no questions and thus remain with a reputation below 15. The other part of the users is located mainly in the bottom right corner of the distribution graph. This is because most people do manage to gain enough reputation and are allowed to upvote. Most users eventually receive enough upvotes because they keep asking or answering questions even if they don't get any upvotes in the beginning.

In the second version of the model, we added the feedback mechanism that corresponds to users asking or answering more questions in the future if they received many upvotes and asking or answering less if they didn't receive enough upvotes. The right part of the blue dots in Figure 3 already looks like a straight line but the maximum probability does not appear at the lowest number of upvotes. One phenomenon that might explain this behaviour is that users who have gained enough reputation in the past to upvote but no longer ask or answer any questions due to the feedback mechanism will still see many questions and answers and might give these an upvote. However, we could assume that these users might lose interest in visiting the Stack Overflow website.

For the final version of the model, the concept of being active on the website is introduced. From Figure 3, we can see that this behaviour tends to result in a power law distribution. As mentioned before, the main difference with the previous version is that users that are discouraged to ask or answer questions will no longer visit the website and thus not upvote any questions or answers.

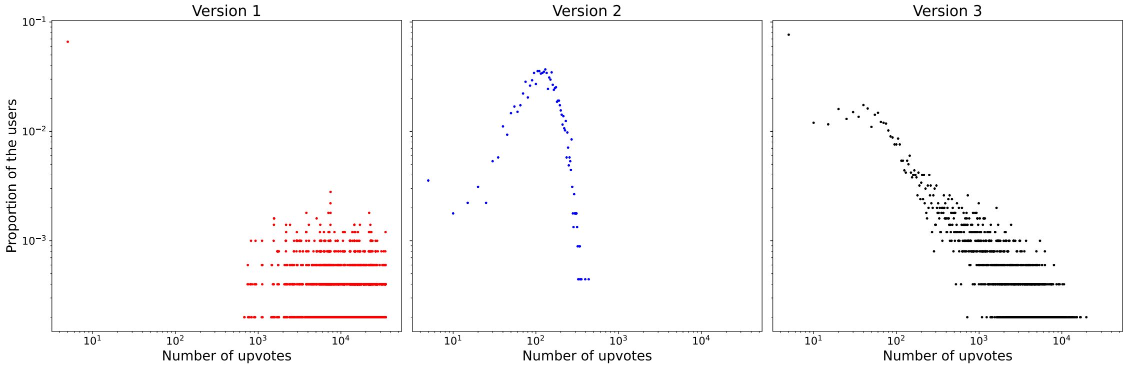

Figure 3: Distribution of upvotes given by users for several versions of the model. The first version only contained some basic Stack Overflow dynamics. In the second version the feedback mechanism was added and in the final version the concept of being active was present.

4.2 Model validation

The graphs in the first column of Figure 4 are the probability distributions of the number of upvotes given by users according to the model and according to actual data found in literature. In the default setting where the probability of upvoting is sampled from a normal distribution, we can see that the exponent of the power law is higher in the actual data compared to the exponent obtained from the model. This indicates that model overestimates the total number of upvotes given by users. One possible explanation for this is that the original assumption that the probability of upvoting is normally distributed around 0.5 is incorrect. For instance, when using an exponential distribution to sample the probability of upvoting, the distribution of the number of upvotes has a higher exponent (bottom left plot in Figure 4).

In the second column of Figure 4, the probability distributions of the reputation are shown. When sampling the probability to upvote from a normal distribution, similar to actual data, it behaves like a power law with a fat tail. The exponent obtained by the model only slightly overestimates the actual exponent. Again, when switching to an exponential distribution the exponent increases because most users tend to have a low probability of upvoting which results in users having an overall lower reputation as a collective. Note that an accurate result in terms of the absolute values of reputation is not expected in this model because in reality there are many more ways to gain and lose reputation on Stack Overflow such as an accepted edit on questions or answers, an upvote on a posted article, bounties, etc. To keep the model relatively simple, these complex dynamics were omitted.

4.3 Dependence of interactions on community size

Figure 5 gives more information about how the number of upvotes and reputation is spread over different-sized communities by representing this data as a boxplot for every community separately. We notice that across all communities there are users that give no upvotes and users with a reputation of 1. As expected, the most interactive users are present in the larger communities about the more popular topics and there is a higher average number of upvotes given by users in the larger communities. In terms of reputation, the average does not seem to be a lot higher for larger communities, however, there are more outliers that are also more extreme. This is similar as on Stack Overflow where the

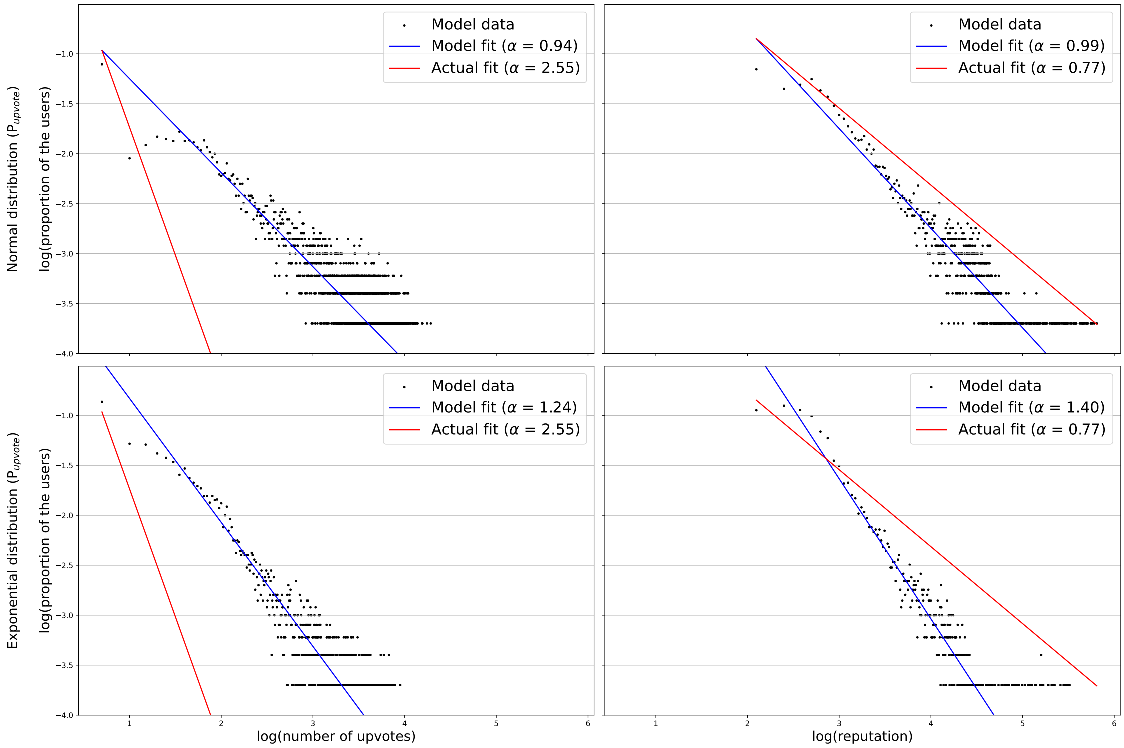

Figure 4: Comparison of the probability distributions on a log-log scale between the model and actual data. The left graph shows the distribution of the number of upvotes given by users and the right graph is the distribution of the reputation gained by users. In the top row the probability to upvote is sampled from a normal distribution and in the bottom row from an exponential distribution.

'main' tag of the users with the highest reputation corresponds to a popular topic. Another similarity with Stack Overflow is that the users with the highest reputation are users that primarily answer questions instead of asking, which is the result of the additional 15 reputation points for the user with the best answer.

In this model, the difference between large and small communities results from two factors. Firstly, fewer users see questions and answers, which results directly in a lower number of upvotes and reputation. Secondly, the upvote bias is independent of the community size, which means that there will be a relatively high amount of users in small communities that have a decreased activity due to the feedback mechanism.

4.4 Sensitivity Analysis

4.4.1 Local sensitivity analysis

Figure 6 is the local sensitivity analysis with the mean of the normal distributions from which the 4 probabilities are sampled as varying input parameters. In terms of the exponent of the upvote and reputation distribution as the outcome variables, $\mu(P_{active})$ and $\mu(P_{upvote})$ are more sensitive compared to $\mu(P_{ask})$ and $\mu(P_{answer})$. As mentioned before in the discussion of using an exponential distribution to sample P_{upvote} instead of a normal distribution (Figure 4), using a distribution where most users have a relatively low probability of upvoting results in a higher exponent of the power law. Using a normal distribution with a low mean has a similar effect, which is why we can see a high exponent

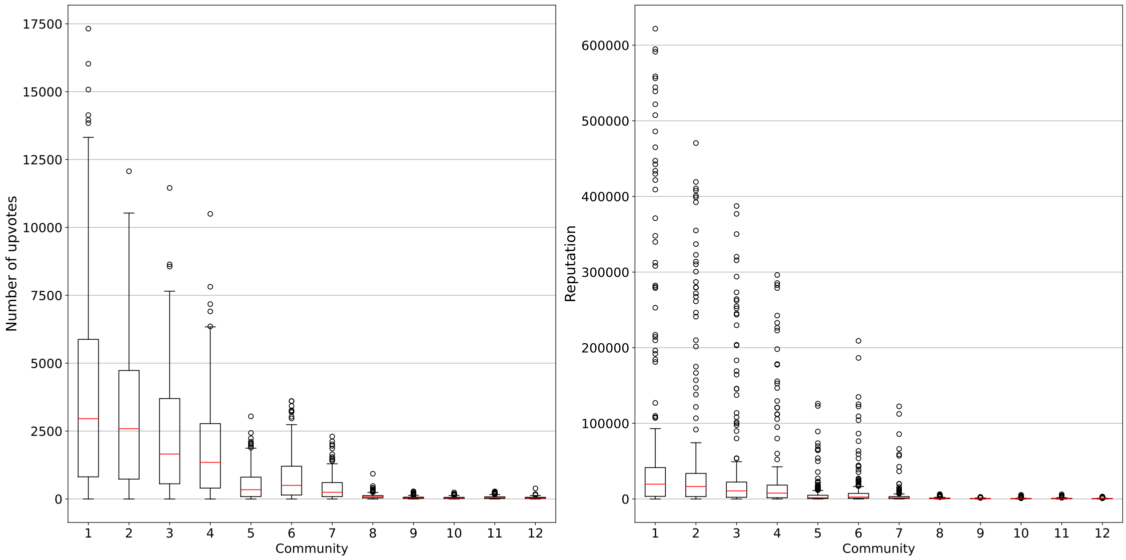

Figure 5: Comparison of the number of upvotes given by users and the reputation of users between the different communities (tags). The left graph shows the boxplots based on the number of given upvotes and the boxplots based on the reputation is located on the right graph. The different communities are ordered based on popularity/size from left to right.

when $\mu(P_{upvote})$ is close to zero. $\mu(P_{active})$ is especially sensitive at low values because when most people have low activity, only a few people will see questions and have a chance to interact with them. The same reasoning also explains why this parameter is sensitive in terms of the total number of questions answered. The effect of both $\mu(P_{upvote})$ and $\mu(P_{active})$ on the total number of questions asked is more indirect. A high average probability for upvoting will result in more users having an increased probability of asking in the step with the feedback mechanism. A high average probability of being active has a similar effect because if more people are active, there is a higher chance of getting more upvotes.

The sensitivity of $\mu(P_{ask})$ is visible when looking at the total number of questions and answers during the simulation. With a higher average asking probability, there will be more questions asked and this also results in more possibilities to answer questions. $\mu(P_{answer})$ seems relatively insensitive even in the total number of answers because every user has a reduced chance of answering a question if there are already a lot of answers to a question.

Figure 7 shows the local sensitivity analysis with the upvote threshold and upvote bias as varying input parameters. As expected, the upvote bias is a very sensitive parameter. At a value of zero, a single upvote will result in an increased probability of being active and asking/answering. In its turn, this increased activity will result in seeing more questions and answers and thus giving more upvotes. Eventually, this leads to a situation where most people are very active. The exponent seems to reach a maximum value when the upvote bias is above 30. In these parameter settings, most users don't obtain enough upvotes that will lead to an increase in their probability of being active and asking/answering. A lower activity will result in seeing almost no questions and thus giving almost no upvotes. This leads to a downward spiral where there is very little overall activity on the site.

In our model, it seems that the upvote threshold is a relatively insensitive parameter,

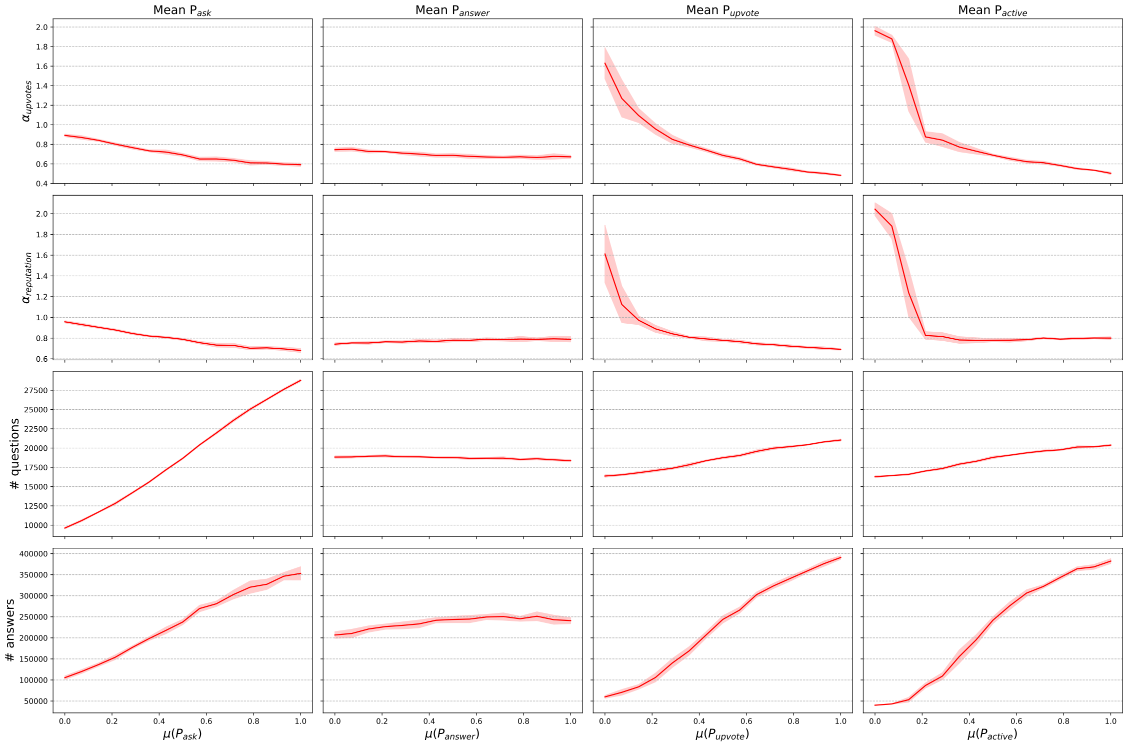

Figure 6: Local sensitivity analysis with the mean of the normal distributions from which the 4 probabilities are sampled as varying input parameters. The outcome variables are the exponent of the upvote distribution in the first row, the exponent of the reputation distribution in the second row, the total number of questions in the third row and the total number of answers in the last row.

which suggests that the lack of upvoting is mostly due to the feedback mechanism instead of the threshold. However, this is an aspect where we think that our model differs from the real system. In reality, the requirement of having at least 15 reputation points before a user can upvote, is introduced to prevent people from creating new accounts and boosting their reputation by upvoting their own questions with these new accounts. As we did not include this (potential) behaviour in our model, this is not something we see reflected in the model data at a low upvote threshold.

4.4.2 Global sensitivity analysis

In Figure 8, we can see the first-order and total-order sensitivity index of the upvote bias, the mean of the normal distributions from which the probability of upvoting and being active are sampled, and the standard deviation of the normal distribution from which the probability of asking and answering are sampled. For all input parameters, the values for the first-order sensitivity index are only slightly lower than the total-order sensitivity index, which indicates that there is only little interaction between the input parameters. Based on this analysis, we can state that the upvote bias is the most sensitive parameter followed by $\mu(P_{active})$ and $\mu(P_{upvote})$. The upvote bias directly influences the updating scheme in the feedback mechanism, while $\mu(P_{active})$ and $\mu(P_{upvote})$ have an indirect effect on the feedback mechanism. This suggests that the feedback mechanism where the probabilities of the users are updated plays a crucial role in the model. Note that the two most sensitive parameters are also the two major changes that were made to the first basic model (Figure 3). From this analysis, we can also see that the standard deviation of the normal distribution from which the probability of asking and answering are sampled are the least important input parameters.

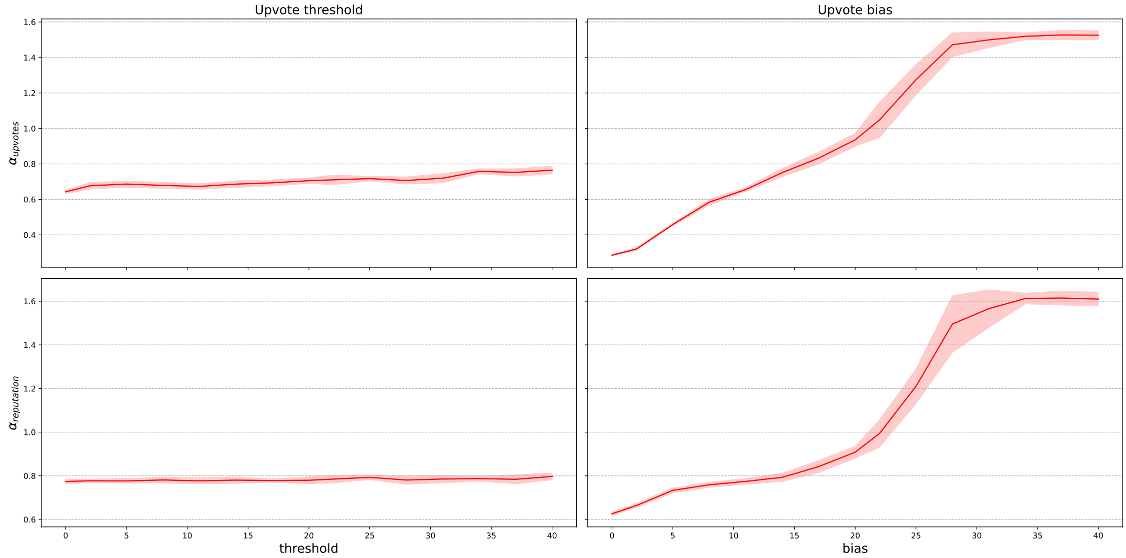

Figure 7: Local sensitivity analysis with the upvote threshold and upvote bias as varying input parameters. The outcome variables are the exponent of the upvote distribution in the first row and the exponent of the reputation distribution in the second row.

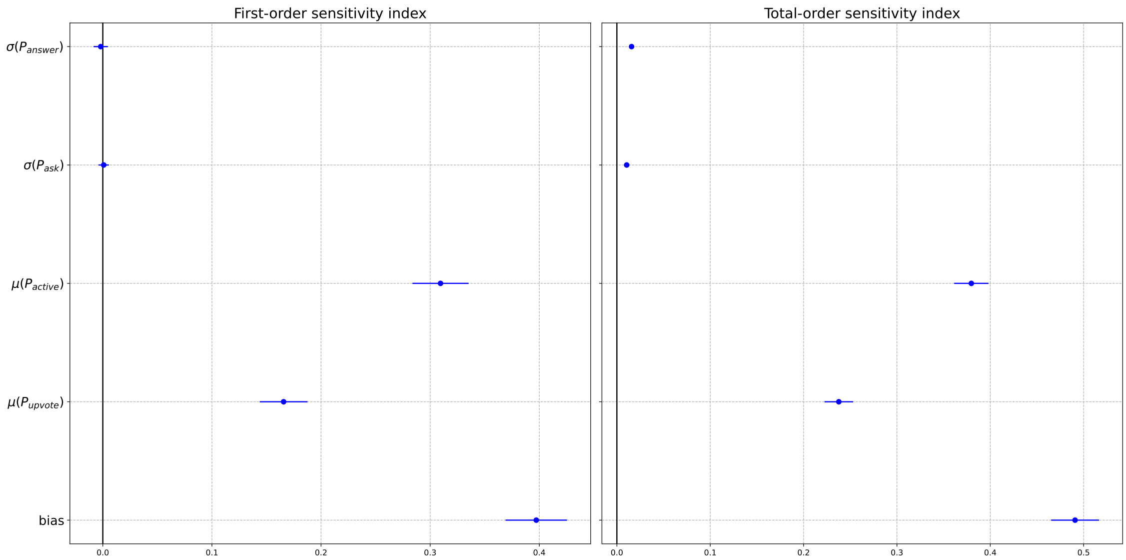

Figure 8: Global sensitivity analysis with the upvote bias, the mean of the normal distributions from which the probability of upvoting and being active are sampled, and the standard deviation of the normal distribution from which the probability of asking and answering are sampled. The analysis is based on the exponent of the upvoting distribution as outcome variable.

5 Conclusions and future work

Our final ABM to represent a Stack Overflow interaction network consists of two major parts. The first part is where users can interact with each other by asking and answering questions, and upvoting both. These actions are mainly determined by four probabilities that every user has and follow the basic Stack Overflow dynamics. The second part is the feedback mechanism that updates the probabilities under the assumption that users will interact more if they receive rewards in the form of upvotes.

The model is able to reproduce the trend of the distribution of giving upvotes and the distribution of reputation. Both distributions follow a power law, but the model is not always capable of getting an exponent that is close to the exponent based on actual data. However, we do have to keep in mind that the model does not contain the complete Stack Overflow dynamics. Another similarity between the model and the real system is that top users are people that primarily answer questions and people that are part of communities about popular topics.

In terms of the sensitivity of the different input parameters, the upvote bias is the most sensitive followed by the mean of the normal distributions from which the probability to be active and the probability of upvoting are sampled.

The next step in this project should be to do a full sensitivity analysis with all the input parameters because due to time constraints, we had to leave out some interesting parameters.

Possible future research could be the analysis of which strategy (mostly asking or answering) is the best one to gain reputation as a new user, as this is a concern of many new users who want to actively participate in the Stack Overflow community. Additionally, there is also the question of what would happen to the network if a new, similar platform emerges which has perhaps a more attractive reward system. However, before tackling these questions, it might be necessary to add a few more details to our model. For instance, there is a certain bias towards upvoting answers from users with a high reputation and also including the feature of downvoting could help increase the accuracy of the model. Another change of the model that can be thought of is to only iterate over active users every time step as this seems like a more realistic approach.

References

[1] *2023 stack exchange network overview.* URL: https://stackexchange.com/about/.

[2] Jun Wang et al. "Who are influential in Q&A communities? A measure of V-Constraint based on knowledge diffusion capability". In: *Journal of Information Science* 45.4 (2019), pp. 488–501.

[3] Jiahua Jin et al. "Why users contribute knowledge to online communities: An empirical study of an online social Q&A community". In: *Information & management* 52.7 (2015), pp. 840–849.

[4] Zhe Liu and Bernard J Jansen. "Questioner or question: Predicting the response rate in social question and answering on Sina Weibo". In: *Information Processing & Management* 54.2 (2018), pp. 159–174.

[5] Muhammad Asaduzzaman et al. "Answering questions about unanswered questions of stack overflow". In: *2013 10th Working Conference on Mining Software Repositories (MSR)*. IEEE. 2013, pp. 97–100.

[6] Xin-Yi Lu et al. "Empirical Analysis of the Online Rating Systems". In: *arXiv preprint arXiv:1510.08142* (2015).

[7] Bhandal Colm. *Distribution of Reputation: Power Law?* URL: https://math.meta.stackexchange.com/questions/21259/distribution-of-reputation-power-law..

[8] Piotr Fronczak. *Scale-Free Nature of Social Networks.* 2018.

[9] Chenbo Fu et al. "Patterns of interest change in stack overflow". In: *Scientific Reports* 12.1 (2022), p. 11466.

[10] Adriana Galvan. "Adolescent development of the reward system". In: *Frontiers in human neuroscience* (2010), p. 6.

[11] Stack Overflow. *What is topic tag targeting on stack overflow?* URL: https://resources.stackoverflow.co/topic/product-guides/topic-tag-targeting/.

[12] Shaowei Wang, David Lo, and Lingxiao Jiang. "An empirical study on developer interactions in stackoverflow". In: *Proceedings of the 28th annual ACM symposium on applied computing*. 2013, pp. 1019–1024.