Simulating Communication Dynamics in the Wikipedia Talk Network Using Agent-Based Modeling

1. Introduction

This project focuses on modeling message exchanges between users on Wikipedia's European talk pages. Each interaction in the dataset represents a user sending a message to another user at a specific time, forming a large network of directed connections. To analyze this network, we first process the raw data into a structured format and build a graph that shows how users are connected. Using the MESA framework for agent-based modeling, we simulate how messages travel across the network over time. Each user is represented by an agent that can send and receive messages, allowing us to observe how information spreads through the system.

The main goal of the project is to better understand the flow of communication in a real-world network. By simulating user behavior, we can explore questions:

1. How do users in the Wikipedia Talk network exchange messages over time?
2. Can we use an agent-based model (ABM) to realistically simulate that behavior?

To explore these, we used a dataset of message exchanges between Wikipedia editors and built a simulation that mirrors their interactions. The goal wasn't just to visualize patterns but to test whether a simple model could replicate the complexity of real human communication at least in structure and timing.

2. Methodology

2.1 Loading and Structuring the Data:

The journey started with a raw text file containing message records who sent what to whom, and when. Each line was a snapshot of a digital interaction. We wrote a custom loader to read this data, clean it up and organize it into a Data frame. The timestamps were normalized so that message timing could be simulated more efficiently.

2.2 Building the Network:

Once the data was structured, we used it to build a directed graph: users became nodes, and each message became a one-way edge. This graph revealed the invisible structure of the Wikipedia conversation network. We also generated an adjacency matrix to represent this same information numerically making it easier to pass into the simulation engine.

2.3 Designing the Agent-Based Model:

To simulate how messages are passed between users, we used the Mesa framework which is great for agent-based modeling. We created agents to represent each user. These agents can send and receive messages, and they keep track of how many messages they've sent or received during the simulation.

2.4 Simulation logic:

The model runs over a series of time steps. In each step, agents check their connections and decide whether to send a message to their neighbors. When a message is sent, it updates the internal state of both the sender and the receiver. This continues until the simulation reaches the desired number of steps.

2.5 Visualization and Analysis:

After the simulation, we collect and analyze data such as the total number of messages sent and received by each agent. We use plots and summary statistics to observe the overall communication patterns and identify highly active or central users in the network.

3. Result

After analyzing, the project provided valuable insights into the dynamics of message exchange within the Wikipedia European talk network. By modeling users as agents and simulating their interactions over 50,000 times steps, we were able to observe both individual and collective patterns of communication. The results revealed that message activity increased steadily throughout the simulation. However, this growth was not uniform across all agents. A relatively small number of users emerged as highly active communicators sending and receiving a disproportionately large share of messages. Meanwhile, most of the users maintained low or moderate levels of activity. This uneven distribution reflects a common characteristic of real- world social networks where activity is often concentrated among a few central participants.

Two key plots were generated to analyze these dynamics:

3.1 Simulation output:

This model tracked message traffic at intervals of 10,000 steps. Plots of sent versus received messages showed increasing dispersion over time, indicating greater variance in user activity. The use of a logarithmic x-axis made it clear that a small number of users dominated the communication flow.

3.2 Agent-based Modeling Output:

The data collected from the MESA simulation confirmed the patterns observed in the simplified model. The final plot at timestep 50,000 showed a similar distribution with most agents clustered in the lower activity range and a few standing out as highly active nodes.

Performance Comparison:

While the Mesa framework provides a detailed and extensible environment for agent-based modeling, it proved to be highly inefficient for simulating large-scale time-step networks such as the Wikipedia Talk graph. Due to the overhead of managing agents, schedules, and environment states at every tick, the simulation time with Mesa increased significantly with scale. In contrast, a non-Mesa model implemented with custom Python code tailored for time-stepped message passing ran considerably faster and proved more practical for long simulations. This efficiency gap is also evident in the plots below, which display message activity over time for both implementations. Although both models produce similar communication patterns, the non-Mesa simulation reaches results in a fraction of the time.

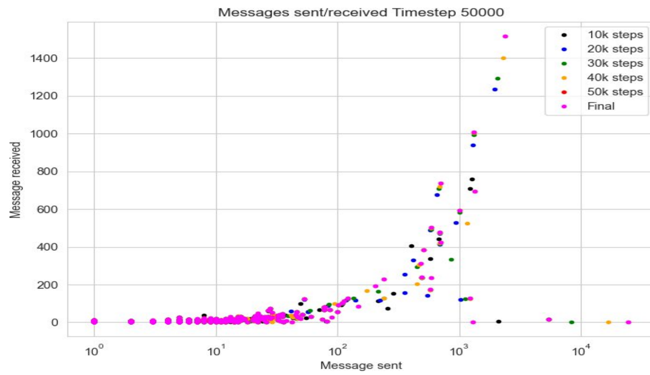

Figure 1: Custom Model Messages over Time (Iterations)

Communication dynamics over 50,000 steps using the custom non-MESA model.

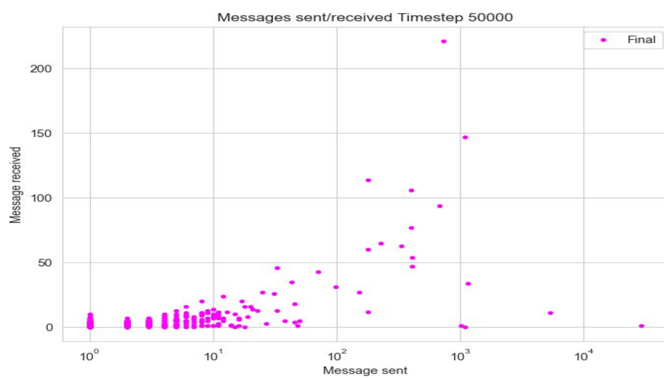

Figure 2: MESA Model Messages

Message exchange pattern simulated using the MESA agent-based model.

Observations and Implications

Network centrality played a key role in user behavior. Agents with more connections tended to be significantly more active in both sending and receiving messages. The simulation highlighted the asymmetric nature of communication in real-world digital networks, where influence and engagement are not evenly distributed. The simplified model proved effective for quick experimentation and scaling, while the full model offered more detailed tracking of agent states.

Overall, the simulation successfully replicated key characteristics of social communication networks, providing a meaningful representation of how messages spread across a real user interaction graph.

4. Research Question: Building Consent

In this part we take an empirical watch into the building of consent in our wiki network. Lets pretend all people involved make a majority vote concerning a new project, or change. Everybody does have an opinion on which direction to take. For simplicity's sake we will assume two believes, encoded by 1 and -1 for computational reasons. (With this setup we can use a simple multiplier to invert the Believe.)
Now we also tweak our model a little bit, instead of a directed graph we now use the complete graph. Every time-step each agent then has a chance (meeting_chance HP) to call a meeting where a number (meeting_size HP) of agents meet and discuss their believes. Depending on the meeting the active agent changes his believe.
In detail, we look at all believes of the agents in the meeting and take the mean of the believes with a twist. We introduce a trust (trust HP) parameter to the discussion.
If an agent has corresponded another agent before (a link in the graph) it trusts the opinion more depending on the hyper parameter. Sometimes an agent also changes his mind (currently a 1% chance if he does not participate in a meeting) as noise. This loop will be run 5 times for different hyper parameters with different starting points.

Hyperparameter Overview:

- model_agents: Number of agents in the network
- active_df: Time-sorted edge network, currently not on use for this task
- max_time: length of simulation in time-sorted sims, currently not in use
- trust: This is a multiplier to strengthen opinions of people we corresponded with
- meeting_size: Size of meetings to build a new consensus for the agent
- meeting_chance: Chance for a meeting to happen, this exists mainly to speed up the simulation
- run_length: How many rounds we try to build consensus
- UDG: undirected graph to gain neighbors we trust

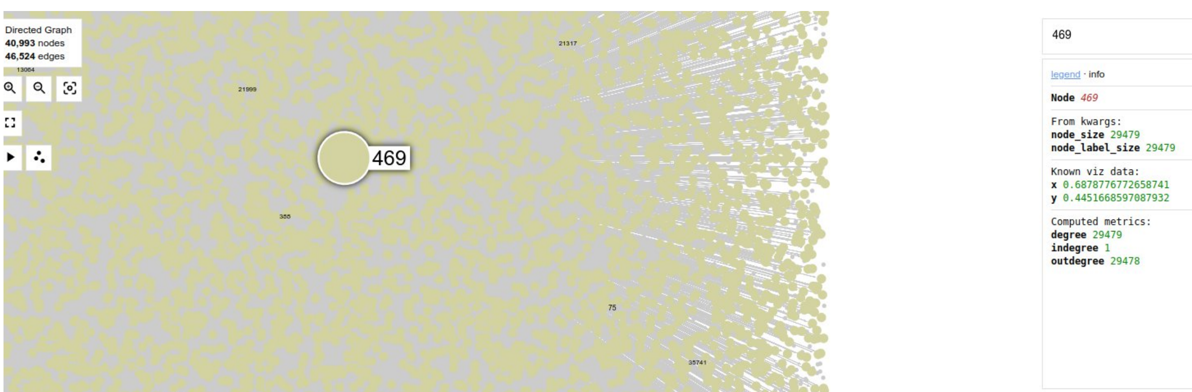

Figure 3: Graph Overview

Findings:
First we identify agents with a lot of trust: We already know them from our initial setup, which can be seen in the python notebook in the visualization part.
Our most connected agent is 469(29k connections) followed by 259 and 102 as well as 63 (This can be seen in Figure 3). Those will be the most influent nodes, with the first one in more than every other meeting per percentage. We will look into how their stance influences other agents over a computational feasible runtime. After each agent has taken an action we will print out the current stances in a bar-chart.

Here is a overview of all the different believes held by our highest connected agents (Trust 0.5):

Iterations	Agent 469	Agent 259	Agent 102	Agent 63
0	-1	-1	-1	-1
1	-1	-1	-1	-1
2	-1	-1	-1	-1
3	1	-1	-1	1
4	1	-1	-1	1
5	1	1	1	1

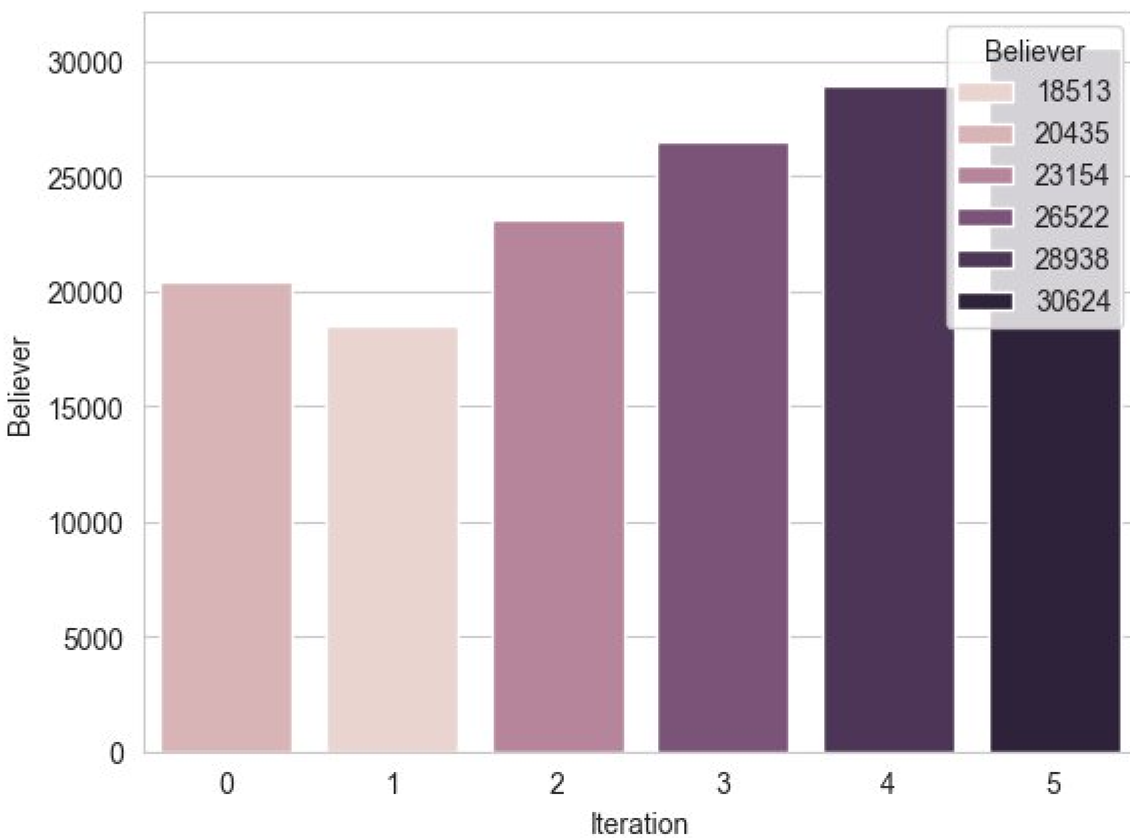

Figure 4: Number of believers in 1 per iteration

Clearly we can see that the high influence nodes didn't have that much impact, it was more or less random. So lets give trust more weight (0.1) All other parameters are the same. Each run unfortunately takes about an hour.

Iterations	Agent 469	Agent 259	Agent 102	Agent 63
0	1	-1	-1	-1
1	1	-1	-1	-1
2	1	-1	-1	-1
3	1	-1	-1	-1
4	1	1	-1	1
5	1	1	-1	1

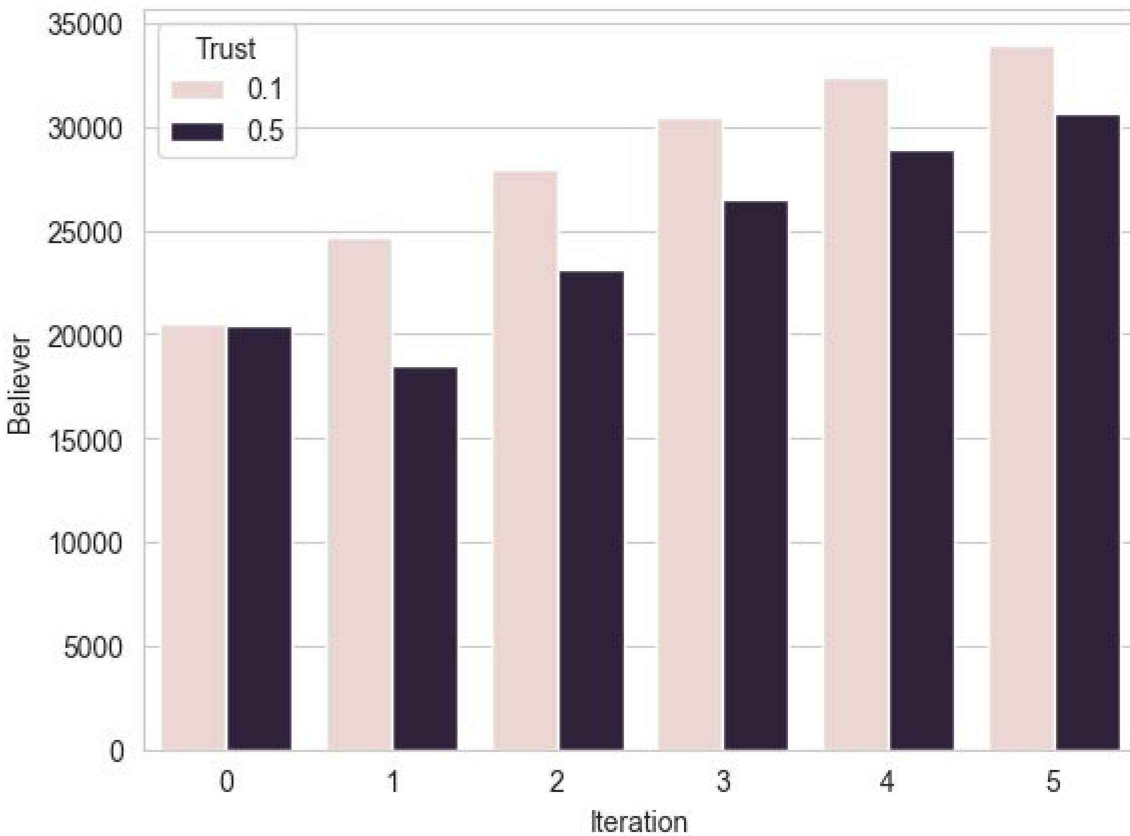

Figure 5: Both trust settings side by side

Conclusion:

Purely empirical we have not found any correlation of trust with the main leaning of the network, however, with more iterations we can surely find it. Otherwise influential nodes decide the direction of the network(see Fig 6.) Not only the most influent ones, however current breakpoints couldn't be found and could be a target of future work. Trust also plays a part to make growth of the dominant fraction smoother but an exact break-point couldn't be found.

```python
max_deg = 50
nodes_with_high_degree = [node for node, degree in UDG.degree() if degree > max_deg]

for node in nodes_with_high_degree:
    print(f"Hello, I'm agent {node} and my Believe is {raw_believe_data.iloc[1][0][node+1]}")
```

✓ [39] < 10 ms

```
Hello, I'm agent 8 and my Believe is 1
Hello, I'm agent 10 and my Believe is 1
Hello, I'm agent 14 and my Believe is 1
Hello, I'm agent 16 and my Believe is 1
Hello, I'm agent 17 and my Believe is -1
Hello, I'm agent 18 and my Believe is 1
Hello, I'm agent 19 and my Believe is 1
Hello, I'm agent 21 and my Believe is 1
Hello, I'm agent 22 and my Believe is 1
Hello, I'm agent 23 and my Believe is 1
Hello, I'm agent 24 and my Believe is 1
Hello, I'm agent 31 and my Believe is 1
Hello, I'm agent 35 and my Believe is 1
Hello, I'm agent 62 and my Believe is 1
Hello, I'm agent 74 and my Believe is 1
Hello, I'm agent 78 and my Believe is -1
Hello, I'm agent 88 and my Believe is -1
Hello, I'm agent 101 and my Believe is -1
Hello, I'm agent 150 and my Believe is 1
Hello, I'm agent 166 and my Believe is -1
Hello, I'm agent 258 and my Believe is 1
Hello, I'm agent 325 and my Believe is 1
Hello, I'm agent 468 and my Believe is -1
```

Figure 6: Agents with degrees higher than 50 and their believes

Future Work:
Change model to give the option to generate or change the believe of specific agents (implemented) to analyze changes of believe in popular nodes. Change meeting sizes or frequency with more computing power, and change length of simulation to search for convergence limits.

5. Discussion

The simulation results demonstrate that agent-based modeling can effectively capture complex communication dynamics in a real-world network. By using actual interaction data from Wikipedia talk pages, the model not only reflected the structural properties of the network but also provided insights into how messages propagate over time. One of the most striking observations was the highly uneven distribution of activity. A small number of users were responsible for most of the messages, while most remained minimally engaged. This aligns with known social network phenomena, such as the "80/20 rule" or power-law distributions where a minority of users drive most of the interactions. The model was able to reproduce this behavior organically, without explicitly enforcing such a distribution, which suggests that the network structure itself plays a significant role in shaping communication patterns. Additionally, the simulation confirmed that users with more connections tend to be more active. This suggests that structural centrality strongly influences user behavior in terms of both message visibility and response likelihood. Such findings are valuable for understanding how influence and information flow operate in online environments. However, the model does have limitations. For instance, agent behavior was relatively simple agents made decisions based on connectivity rather than more nuanced social factors like content relevance, reputation, or historical interactions. Incorporating such dimensions could lead to even more realistic simulations. Another constraint was computational efficiency. While the fast version of the model allowed for quicker execution, it did so by sacrificing some of the richer agent-level dynamics that the full model captures.

Despite these limitations, the simulation provides a useful foundation for exploring communication in social networks. It shows how large-scale interaction patterns can emerge from simple rules and structures, and it opens possibilities for further research such as simulating message spreading in response to external events or comparing networks with different topologies.

6. Limitations

The project successfully captured key aspects of message flow in an online communication network, there are several important limitations that should be acknowledged.

First, the agents in this model operate under simplified assumptions. Their behavior is driven primarily by network connectivity and timing without considering the content or intent of the messages being exchanged. In the reality, communication is rarely structure-driven alone it is influenced by topic relevance, user preferences, prior interactions and social incentives. As such, while the simulation reproduces general patterns of activity, it cannot fully explain why specific interactions happen.

Second, the model treats all users as functionally identical, aside from their network position. This ignores the individual variability that characterizes human behavior. In real networks, users exhibit diverse traits some are initiators, others are passive readers, and some respond only under specific conditions. Modeling such heterogeneity would require more complex agent rules and additional data sources. Another limitation lies in the temporal resolution of the simulation. Although the timestamps were normalized for practicality, this process discards finer-grained-timing information that could influence interaction patterns. Real-world communication often features bursts of activity, idle periods, or temporal clustering none of which are explicitly modeled here. From a technical standpoint, the simulation is also constrained by performance trade-offs. The fast version of the model improves scalability but lacks the state tracking and flexibility offered by the full MESA-based implementation. Conversely, the full model is richer but more computationally expensive, limiting its applicability on very large datasets without optimization. Finally, the project focuses on a single dataset from a specific domain Wikipedia talk pages while valuable may not generalize to all forms of online interaction. Other platforms like social media, forums, or messaging apps exhibit different user behaviors and network dynamics that would require model adaptation.

Despite these limitations, the current model serves as a meaningful first step. It highlights how network structure alone can drive complex outcomes and lays the groundwork for more detailed, behaviorally rich simulations in future work

7. Conclusion

This project set out to model and simulate the flow of communication within a real-world digital network, using agent-based modeling to bring static data to life. By leveraging Wikipedia's European talk page interactions as a foundation, the simulation successfully recreated key characteristics of social communication: uneven participation, emergent centrality, and structure- driven dynamics. Through the construction of a directed graph and the implementation of a simplified agent behavior model, we were able to observe how message activity evolves over time and how certain nodes those structurally advantaged naturally emerge as hubs of interaction. This reinforces a well-established concept in network science: that behavior often reflects structure, even in the absence of complex decision-making rules. What this project ultimately highlights is the power of minimal models to capture meaningful social dynamics. Despite the agents lacking detailed behavioral nuance, the system whole displayed emergent properties that closely mirror real online behavior. This suggests that even relatively simple simulations can offer valuable insights—provided they are grounded in empirical data and well-designed structures. Of course, the model does not attempt to reproduce the full complexity of human communication. Rather, it provides a conceptual and computational framework from which more detailed explorations can grow. It acts as a starting point for asking deeper questions about influence, inequality, and the mechanics of interaction in networked environments. In a broader sense, this work sits at the intersection of data science and social simulation. It demonstrates how computational modeling can transform static datasets into dynamic systems revealing not only what has happened, but how similar behaviors might unfold under varying conditions. It is this ability to move from data to dynamics that makes agent-based modeling such a powerful tool in understanding our increasingly connected digital world.