9^{th} International Conference on Computational Social Science IC^2S^2
July 17-20, 2023, Copenhagen, Denmark

100-Bot Convo: Social Media Simulation Using a Pre-Trained Language Model-Based Agent Network

Keywords: discourse simulation, generative pre-trained language models, natural language generation, agent-based simulation, information diffusion

Extended Abstract

This study presents a novel methodology developed to simulate the dynamics of information diffusion in the online social network environment. The diffusion process is modeled by a network where the nodes are agents that make use of a pre-trained language model (PLM) to generate dialogue among their neighbors. Since the advent of massive online social networking sites (SNS), interest on studying how information is spread in these platforms has been on the rise [6]. These studies regarding SNS are bound to using existing, limited, real-world SNS data which makes it difficult to isolate mechanisms under review. Additionally, previous studies tend to use language formalism to represent information passed on from one agent to another[7]. Given this, we propose using PLM-based agents to generate natural language discourse to allow analysis on the development of the discourse and enable researchers to model diffusion dynamics in a controlled environment. This study would build on past research that simulate natural language content via PLMs in applications across disciplines (e.g., judicial or argumentative content generation) [1, 2] to simulate the information diffusion process in SNS.

The main goal of the proposed methodology is to emulate the discourse that occurs in SNS and validate its outputs by comparing it with real-life information diffusion dynamics. We propose that, with the generation of natural language content added into the simulation, the development of the discourse could now be tracked using new metrics to analyze the progression of network-wide discourse over time. We present a preliminary attempt for a proof-of-concept of our idea, applied to tracking prevalence and sentiment attribution towards a topic of interest, which in this case is "masks" in the context of the COVID-19 pandemic.

In this study, the simulation environment consists of the two main components: (1) the agents, that represent the users of SNS, and (2) a social network where all agents are connected, that represents the SNS platform. The agents are simple actuators whose main behavioral triggers are to generate, send, and receive messages with each other then keep a history of all these messages. All of the agents are created with random neighbors, connected by edges in the network, who act as a federation of the instance of Grounded Open Dialogue Language Model (GODEL) [5]. GODEL is the PLM used to generate messages regarding the topic of "masks". It makes use of the messages exchanged between agents as prompt to generate a response upon. Algorithm 1 details the loop that the environment followed in the simulation.

In order for the simulation model to create outputs that are bound to reality, real-world data was used as a seed to create the starting point of the discourse. The input data are English tweets scraped with the keyword set as "mask". These tweets were manually filtered and annotated to be of relevance to the COVID-19 pandemic mask discussion. After annotation, the sentiment scores of each tweet were analyzed using the VADER sentiment model [3] wherein the compound sentiment score was documented along with the tweet. It is with this sentiment metric that the initial conditions of the simulation were configured so that there is an equal number of agents having seeds of positive and negative sentiments toward the topic of "masks".

The sentiment attribution scores were computed per agent each time step and were averaged across the network as well. Sentiment attribution scores were computed with the VADER model, based on the recent 20 messages that they have sent, received, and stored in their history per time step. Messages in the agent history were filtered so that instances with the topic of "masks" as the nominal subject were the only ones included in the sentiment measurement. On the other hand, topic prevalence was tracked all throughout the simulation by keeping track of how many messages in the network talks about the topic of "masks" as the nominal subject.

The simulation consisted of 100 bots, evenly split to contain both positive and negative seeds, and it lasted for 50 time steps. The configuration of the simulation is visualized through the social network with agents classified according to their sentiment attribution scores as shown in Figure 1. The proposed metrics revealed diffusion dynamics in the simulation capturing the waning interest in the discourse regarding the topic of "masks". Figure 2 shows how the number of messages with the topic of "masks" as the nominal subject decreased as the simulation progressed. This is affirmed with the rising sentiment attributions in the discourse as there are decreasing messages about the topic of "masks" that in turn uplifted the network sentiment attributions to the topic by virtue of averaging, as presented in Figure 3. Additionally, qualitative metrics were performed to explore the discourse regarding the topic. Figure 4 shows the top modifiers generated in the discourse where the topic of "masks" is the nominal subject.

Although the proposed simulation methodology was able to capture the expected stable diffusion process of information [4] throughout the discourse, we recognize that it is limited and could be improved as it is a work-in-progress. Extensive analysis and validation methods, both quantitative and qualitative, can be applied to the generated discourse of the simulation to better frame and understand its dynamics. Furthering the methodology could allow more robust applications in varying fields of public information, social psychology, and other fields.

References

[1] Gregor Betz. Natural-language multi-agent simulations of argumentative opinion dynamics. *Journal of Artificial Societies and Social Simulation*, 25(1), 2022.

[2] Sil Hamilton. Blind judgement: Agent-based supreme court modelling with gpt, 2023.

[3] C.J. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. 01 2015.

[4] Philipp Lorenz-Spreen, Bjarke Mørch Mønsted, Philipp Hoevel, and Sune Lehmann. Accelerating dynamics of collective attention. *Nature Communications*, 10, 2019.

[5] Baolin Peng, Michel Galley, Pengcheng He, Chris Brockett, Lars Liden, Elnaz Nouri, Zhou Yu, Bill Dolan, and Jianfeng Gao. Godel: Large-scale pre-training for goal-directed dialog, 2022.

[6] Abdul Razaque, Syed Rizvi, Meer Jaro khan, Muder Almiani, and Amer Al Rahayfeh. State-of-art review of information diffusion models and their impact on social network vulnerabilities. *Journal of King Saud University - Computer and Information Sciences*, 34(1):1275–1294, 2022.

[7] Chao Wang, Zong Xuan Tan, Ye Ye, Lu Wang, Kang Hao Cheong, and Neng-gang Xie. A rumor spreading model based on information entropy. *Scientific Reports*, 7(1):9615, Aug 2017.

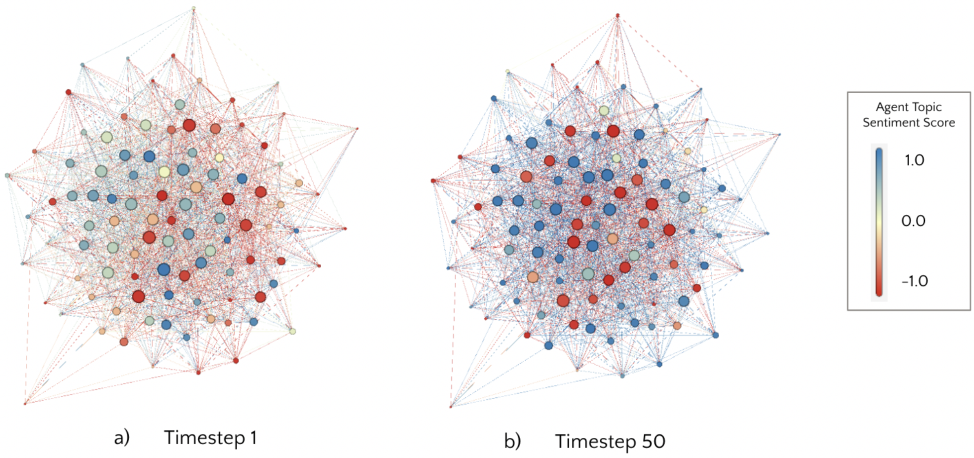

Figure 1: Topic sentiment attributions of the agents in the social network in time step 1 (a) and time step 50 (b).

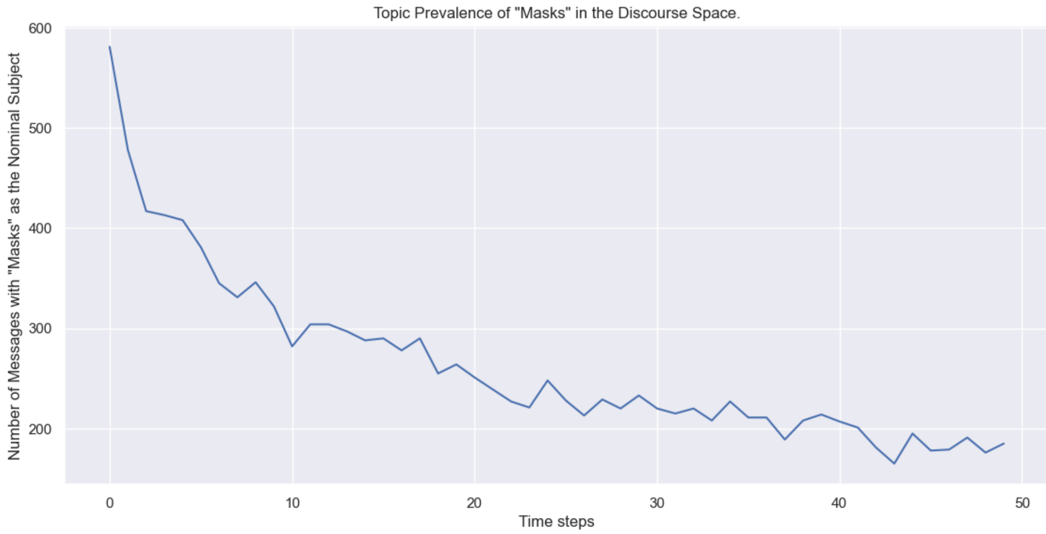

Figure 2: Topic prevalence throughout the simulation.

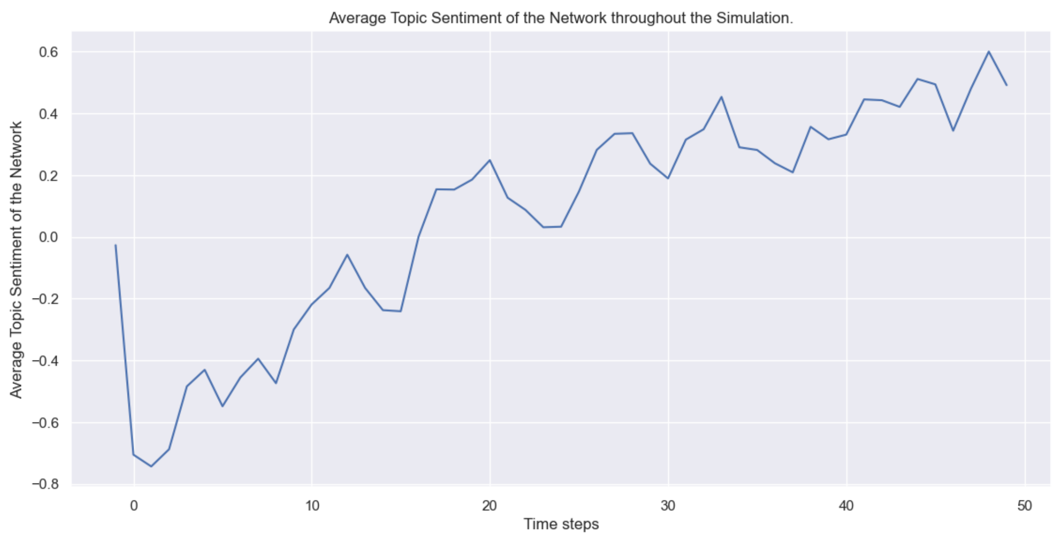

Figure 3: Average topic sentiment attributions of the network throughout the simulation.

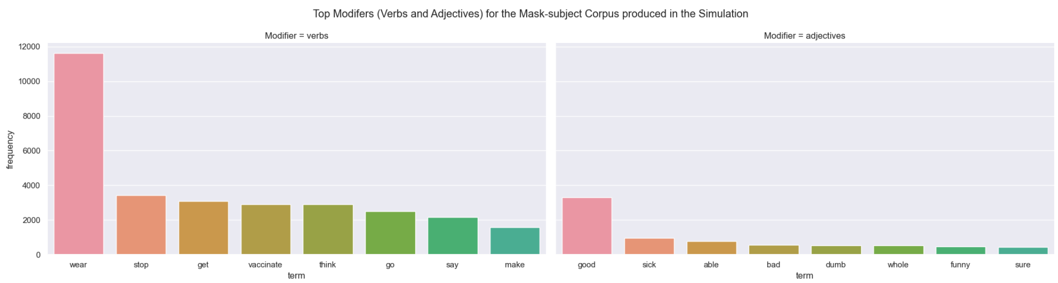

Figure 4: Modifiers used to describe masks throughout the discourse.

Algorithm 1 Main loop for the simulation

```
procedure SIMULATECONVERSATIONS(t, n, p)
    Instantiate the Social Network;                    ▷ Instantiate the PLM instance
    Instantiate n Agents with p as Initial Prompt;
    for each agent do
        Send the initial prompt to neighboring agents;
        Update chat history;
    end for
    while timestep ≤ t do
        for each agent do
            Feed chat history as prompt to the PLM;    ▷ Latest 20 messages as prompt
            Generate a message to neighboring agents;
            Send the message to neighboring agents;
            Receive messages from neighboring agents;
            Update chat history;
        end for
    end while
    Compile all agent's chat history;
    return CompiledChatHistory
end procedure
```