Artificial Social Media Campaign Creation for Benchmarking and Challenging Detection Approaches

Janina S. Pohl,[1] Dennis Assenmacher,[2] Moritz V. Seiler,[1] Heike Trautmann,[1,3] Christian Grimme[1]

[1] Data Science: Statistics and Optimization, University of Münster, Münster, Germany
[2] Computational Social Science, GESIS, Cologne, Germany
[3] Data Management & Biometrics Group, University of Twente, NL
janina.pohl@uni-muenster.de, dennis.assenmacher@gesis.org, moritz.seiler@uni-muenster.de,
trautmann@wi.uni-muenster.de, christian.grimme@uni-muenster.de

Abstract

Social media platforms are essential for information sharing and, thus, prone to coordinated dis- and misinformation campaigns. Nevertheless, research in this area is hampered by strict data sharing regulations imposed by the platforms, resulting in a lack of benchmark data. Previous work focused on circumventing these rules by either pseudonymizing the data or sharing fragments. In this work, we will address the benchmarking crisis by presenting a methodology that can be used to create artificial campaigns out of original campaign building blocks. We conduct a proof-of-concept study using the freely available generative language model `GPT-Neo` in this context and demonstrate that the campaign patterns can flexibly be adapted to an underlying social media stream and evade state-of-the-art campaign detection approaches based on stream clustering. Thus, we not only provide a framework for artificial benchmark generation but also demonstrate the possible adversarial nature of such benchmarks for challenging and advancing current campaign detection methods.

Introduction

Constructing machine-learning models that help identify coordinated mis- or disinformation (Wardle 2018) is regarded as a challenging problem for multiple reasons: first, there is a lack of available data resulting from a benchmarking crisis in social media research (Assenmacher et al. 2021). Platforms restrict data sharing (allegedly) to protect user privacy or (more probable) to keep their competitive advantage. Twitter, which is considered one of the more open platforms, only allows sharing tweet IDs. With knowledge of these IDs, original posts can be downloaded (also called rehydrated) – but only if they are available (Twitter 2021). Clearly, this leads to an ever-changing and, thus, incomparable dataset due to users and Twitter deleting or restricting content over time. Second, creators of coordinated disinformation used advanced technology and developed sophisticated strategies, making them harder to detect and annotate by external parties. Third, social media data is heterogeneous. Thus, several types of campaigns targeting different groups of users exist (Pacheco et al. 2021; Choraś et al. 2021).

In our understanding, a social media campaign consists of a coordinated group of users whose activities (such as posting, liking, etc.) are connected by a common goal (Lee et al. 2014). As noted by Ferrara et al. (2016), the definition of malignant campaigns in this context is more challenging due to the variety of strategies, motivations, and forms of user engagement. Except for anecdotal reports, almost no shareable data on campaigns is available. This leads to a lack of suitable benchmarking data and the inability to compare performances of detection algorithms adequately. Although the problem was identified (Bruns 2019; Pasquetto, Swire-Thompson, and Amazeen 2021), there is currently only little progress in methodology for general benchmarking creation and unified algorithm assessment for campaign detection.

Artificial data that closely reflects core statistics of the originally observed data do not have to adhere to the platform's strict data sharing rules; it can be distributed among researchers. This work presents a new highly flexible framework to create such artificial simulations of disinformation campaigns. Actual campaigns observed and recorded by existing campaign detection algorithms and verified by human annotators are used as a blueprint to recreate realistic campaigns. Moreover, several campaign dimensions can be flexibly adjusted to simulate coordinated actions of manipulators not yet observed. Once a campaign strategy exists, (a) many different campaign instances can quickly be created, and (b) shared without violating data-sharing regulations.

The general idea of our new framework can be seen in Figure 1. Initially, campaigns are detected, recorded, and analyzed using current detection approaches. Researchers like Lee et al. (2014) or Varol et al. (2017) already showed that campaigns follow specific patterns which we will call *stereotypes*. These stereotypes can simply be recreated, adjusted, or combined into more complex patterns. For example, coordinated spam attacks can be made more challenging to detect by replacing their content with various, more diverse artificial tweets. Large language models like Generative Pre-Trained Transformers (GPT) can create meaningful, realistic tweets, which are often even indistinguishable from human-written texts (Brown et al. 2020; Fagni et al. 2021). In a subsequent step, the artificially created campaigns can be embedded in any social media stream used as underlying noise. Since the characteristics of the artificial campaigns can be adjusted through various hyperparameters, researchers can create different benchmarking datasets comprising a variety of challenges. A benchmark could simulate

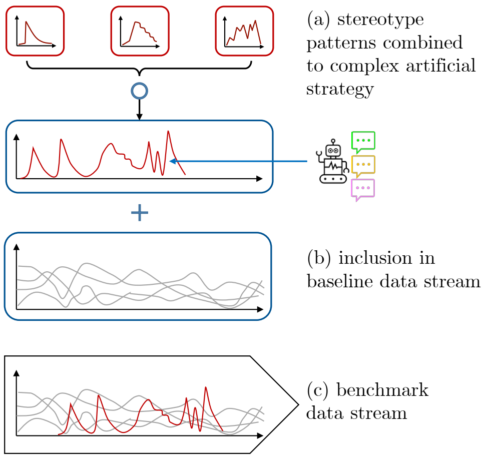

(a) stereotype patterns combined to complex artificial strategy
+
(b) inclusion in baseline data stream
(c) benchmark data stream

Figure 1: General concept of benchmark generation.

automated actors (social bots) spreading malicious links to challenge spam bot detection approaches, while another can simulate human trolls distributing fake content.

In this work, we instantiate our novel framework using GPT-generated texts to fill previously identified campaign stereotypes with meaningful content. This yields two critical insights. First, from a benchmarking perspective, it can be confirmed in a proof-of-concept that the proposed approach is feasible and can be used to evaluate a modern stream clustering-based method for campaign detection. Second (and qualitatively), however, the approach also demonstrates that automatically generated and meaningful content can be successfully introduced into campaign patterns. At the same time, this points to significant potential in using this technology in the context of malignant communication, thus confirming the urgency of evaluating current detection methods and their limitations under an adversarial regime.

The paper is structured as follows: after discussing related work, we introduce the framework concept before instantiating it for later experimental evaluation. We conclude with ethical considerations, discussion, and future perspectives.

Related Work

Social Media Benchmarks Existing benchmarks in campaign detection adhere to strict data sharing rules. For example, in the work of Naseem et al. (2021), the influence of tweets on public sentiment related to COVID-19 is examined; the authors publish the pseudonymized texts in combination with the sentiment label. In contrast, the social bot detection dataset by Cresci et al. (2017) only offers the Twitter IDs as well as their label instead of actual user or text data. However, it was shown that the persistence of Twitter data over the years subsides so that the benchmark is falsified when tweets are deleted (Samper-Escalante et al. 2021).

One solution to circumvent data sharing restrictions is to distribute the data within research groups, like in the Digital Democracy Lab in Zurich[1] or the Social Feed Manager of the George Washington University[2]. Exploiting this principle, the project initiators invite others to join their research team for projects to share the data among more researchers (Gilardi et al. 2021). Another solution is to set up algorithm competitions on platforms like Kaggle or Codalab so that the data never leave the realm of the data holder, but only performance values are shared. Assenmacher et al. (2021) propose a framework that enables researchers to set up and host algorithm competitions dynamically. Nevertheless, realizing algorithm competitions requires suitable infrastructure and sufficient computational resources.

Artificial Data Augmentation Recently, automatically machine-generated text became more prevalent in various natural language processing (NLP) domains such as abusive language or sentiment detection. Next to easy data augmentation (EDA) encompassing methods like synonym replacement, random swap, insertion, or deletion, researchers developed sophisticated algorithms for benchmark creation (Wei and Zou 2019). Calabrese et al. (2021) developed a new benchmark and evaluation system for hate speech detection. They use words and structural patterns predictive of abusive language and incorporate them into normal posts to generate partly artificial adversarials. With Polyjuice, Wu et al. (2021) present a fine-tuned `GPT-2` model to produce counterfactual examples of textual input data to improve model generalization in various domains, while Robeer, Bex, and Feelders (2021) utilize a combination of a generative adversarial network (GAN) and `BERT` encoder. Hartvigsen et al. (2022) use `GPT-3` to automatically create adversarials for hate-speech detection, focusing on creating hard-to-classify implicit abusive content.

In contrast to creating artificial content (i.e., text or images), the simulation of networks with artificial user nodes and edges is more common. Xia et al. (2014) developed an artificial data generator that applies a Poisson process to create user activities and simulate network behaviors. Bucur and Holme (2020) generate small user networks with ten nodes for simulating the spread of epidemics. Lotito, Zanella, and Casari (2021) model the spread of fake news by considering time-varying engagement and levels of trust. However, no simulation of manipulation campaigns has been proposed until now to the best of our knowledge.

Campaign Detection Aiming for campaign detection, social media data are examined by a plethora of detection algorithms, like user networks, images, or post content (Choraś et al. 2021), using human-based, purely ML-based, or mixed strategies (Orabi et al. 2020). Human experts or paid crowdsourcing workers are employed to inspect social media data to detect spam campaigns or simple social bots. However, this is impossible for sophisticated campaigns. Further, human inspectors are costly, require more time than an algorithm, and may decrease accuracy for speed (Alarifi, Alsaleh, and Al-Salman 2016; Cresci et al. 2018).

Different types of ML algorithms were developed e.g., (un-)supervised, or adversarial learning (Derhab et al. 2021). Supervised methods like in Pritzkau, Winandy, and

[1] https://democracy.dsi.uzh.ch/project/digital-democracy-lab/

[2] https://social-feed-manager.readthedocs.io/en/master

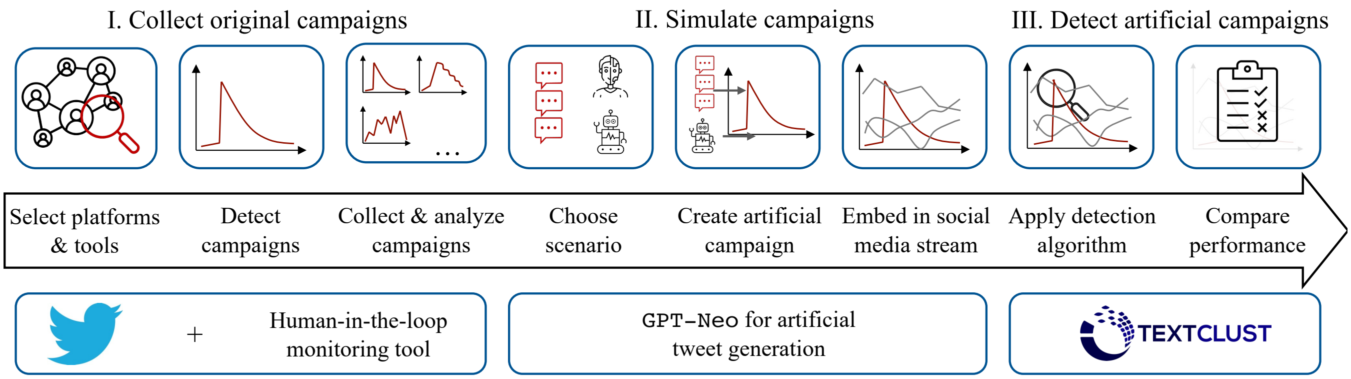

I. Collect original campaigns
II. Simulate campaigns
III. Detect artificial campaigns
Select platforms & tools
Detect campaigns
Collect & analyze campaigns
...
Choose scenario
Create artificial campaign
Embed in social media stream
Apply detection algorithm
Compare performance
+
Human-in-the-loop monitoring tool
GPT-Neo for artificial tweet generation
TEXTCLUST

Figure 2: Overview of the proposed benchmarking framework for creating artificial campaigns. While the abstract framework components are depicted on top, our concrete proof-of-concept instantiation is shown at the bottom.

Krumbiegel (2021) flag posts as being trustable or illegitimate by using a `BERT` (fine-tuned bi-directional encoder representations from Transformers) model. Varol et al. (2017) use a k-nearest neighbor classifier with dynamic time warping to deal with time series data. However, these approaches depend on a large training dataset with unbiased labels, which is hard to obtain in a suitable size (Cresci 2020). In the unsupervised domain, patterns like synchronized user behavior or similar texts are identified (Chen and Subramanian 2018). However, if accounts do not behave as anticipated or are heterogeneous, these approaches fail (Cresci et al. 2019). Adversarial algorithms focus on generating challenging examples, like the mutation approach by Cresci et al. (2018). Close evaluation is required to prevent the generation of implausible scenarios.

Human-in-the-loop approaches rely on the interaction between human intuition and algorithms. One method is the two-phase monitoring framework proposed by Assenmacher et al. (2020b). A continuous textual data stream is processed and summarized in real-time for topic detection and rating (w.r.t. importance). Expert users can analyze suspicious topic patterns based on time-dependent topical changes through an interactive dashboard. Although computational and human resources are required for monitoring, the approach has been successfully applied during the Brexit or the German or US governmental elections (Assenmacher et al. 2020a). Thus, we will use it subsequently to detect campaigns used as blueprints for artificial recreation. An overview of more detection algorithms for several types of campaigns can be found in the works of Choraś et al. (2021); Orabi et al. (2020) or Da San Martino et al. (2020).

Artificial Campaign Framework

The detailed process of our proposed benchmarking methodology is displayed in Figure 2. The process consists of three main steps: campaign collection, simulation, and detection.

Original Campaign Collection In the collection phase, the target social media platforms have to be selected. The framework is agnostic of the concrete detection algorithm and input data. Selected detection algorithm(s) can then be applied to the data stream to identify any campaign. Eventually, researchers must examine all detected campaigns in an exploratory data analysis to understand important distributional characteristics such as how many accounts participated, or which kind of message was spread. Since manipulators supposedly use various strategies to manipulate other users, several types of campaigns can be collected.

Campaign Simulation The collected campaign characteristics are used in the second phase to design various artificial campaigns, e.g., disinformation prior to elections or malicious links promotion with click-baits. As campaign participants, either a single or a network of accounts can be used, which can mimic the behavior of social bots or legitimate users. Next to insights from the analysis of the real campaigns, insights regarding the working principle of the subsequently used detection algorithm can also be incorporated to generate challenging problems. Researchers can then create artificial campaigns following these configurations. Since campaigns are not executed in isolation but accordance with other users' actions, researchers must choose the basic stream in which the artificial campaign will be inserted. Note that only the recreated campaign is artificial; it is inserted in any static dataset or social media stream consisting of actual posts collected earlier. Thus, conveniently, the ground truth regarding labeling artificially created campaigns by researchers and others is directly available.

Artificial Campaign Detection The last step of our framework is to test the performance of campaign detection algorithms. Hence, researchers do not only gather insights into the algorithms' functionality but also reveal opportunities for improvements. The algorithms used to create artificial campaigns can be shared without problems and applied to any problem setting or data. Since campaigns are used as blueprints that were observed in a social media stream for the simulation, they closely reflect the real manipulator's actions. Nevertheless, it must be noted that it is possible that other users would react differently to the artificially generated campaign than they reacted to the original campaign in a real social network environment. Since the analysis of the artificial campaign is conducted in hindsight, possible deviating user reactions cannot be simulated realistically.

Framework Instantiation

To instantiate our artificial campaign framework, it is necessary to specify detection methods, the models used to create the content of the campaigns and the algorithms used to simulate the users' actions and behaviors. Since the proposed framework is flexible w.r.t. the instantiation of each step, we evaluate a concrete setup in this study.

Stream Clustering for Collecting Campaigns

For campaign detection, we rely on an unsupervised human-in-the-loop approach proposed by Assenmacher et al. (2020b). The tool is built upon the idea of monitoring the outputs of a stream clustering technique called `textClust` to identify homogeneous groups of potentially coordinated topics in an unbounded textual data stream. The algorithm was initially proposed in 2017 (Carnein, Assenmacher, and Trautmann 2017) but was recently updated and benchmarked on various Twitter datasets (Assenmacher and Trautmann 2022). `textClust` utilizes a TF-IDF cluster representation which is updated incrementally to deal with distributional changes (concept drifts). Therefore, the whole clustering process is separated into two phases. Within the online phase, incoming observations are continuously clustered into statistical summaries called micro-clusters. Due to concept drifts, they can change over time, i.e., can become more important by incorporating new observations or less important if they are not updated anymore. Each time a new text is assigned to an existing cluster, its weight, i.e., its importance, is increased. On the contrary, if a cluster is not frequently updated, the weight is decreased according to an exponential fading function. Within Figure 3a, one example of the changing micro-cluster weight over time is reflected by the unmodified micro-cluster C_o. In the second offline phase of the algorithm, a snapshot of micro-clusters can be used at any specific point in time to asynchronously re-cluster them using traditional clustering techniques.

Generative Language Models for Simulation

Generative language models compute a probability distribution over sequences of tokens to predict future words, given the previous token in the sequence. State-the-art models are based on the Transformers architecture defined by Vaswani et al. (2017). Previously, primarily convolutional neural networks (CNNs) or recurrent neural networks (RNNs) were used for natural language generation (NLG) tasks. However, these models only consider signals from close input or output positions, i.e., examine a narrow word neighborhood (Wolf et al. 2020). In contrast, Transformers utilize the attention mechanism that draws global dependencies between input tokens since, in natural language, coherent words may not occur consecutively (Vaswani et al. 2017).

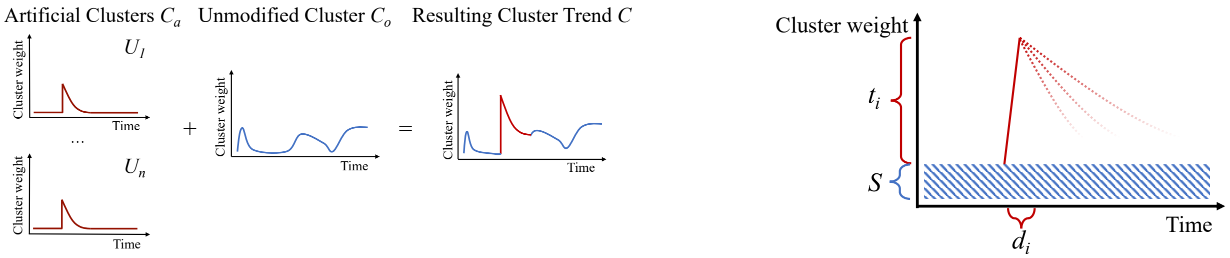
OpenAI developed GPT models from 2018 onward, which are based on the decoder part of the Transformer architecture (Radford et al. 2018). The latest version, `GPT-3` by Brown et al. (2020), outperformed other language models by the time of its publication. Short texts generated by GPT models are nearly indistinguishable from human-written texts (Ippolito et al. 2020). Nevertheless, the model is not freely available, and researchers must pay for each generated token. Therefore, EleutherAI developed `GPT-Neo`, a freely available version. It was trained on donated computational power using a dataset called The Pile, an 825-gigabyte English text corpus (Gao et al. 2020). EleutherAI reports that it shows similar performance to `GPT-2` and only slightly worse performance than `GPT-3` in NLG (EleutherAI 2022).

There are several other models available for NLG, for example the `MAGMA` model – Multimodal Augmentation of Generative Models through Adapter-based Finetuning – developed by Aleph Alpha (Eichenberg et al. 2021), Turing-NLG (`T-NLG`) by Microsoft (2020), and the Text-To-Text-Transformer (`T5`) developed by Raffel et al. (2020). However, since `GPT-3` achieves the best performance on various NLG tasks and `GPT-Neo` is slightly worse and freely available, we decided to use `GPT-Neo`.

Generation of Simulated Campaigns

We create artificial campaigns in which `GPT-Neo`-generated tweets are inserted by mimicking the actions of disinformation spreaders (Figure 3a). We assume that manipulators try to promote their opinions on social media by managing multiple user accounts in parallel. As these coordinated accounts send coherent content, they are grouped in one cluster, leading to an increase in cluster weight observable in the monitoring tool. Thus, depending on how many coordinated users U_i participate, their aggregated actions determine the resulting micro-cluster pattern C_a. However, their actions are not examined in isolation but are assumed to be embedded in an original, content-related, and unmodified micro-cluster C_o emerging from the actions of other users. Thus, the shape and the volume of C_o influence the final trend of the resulting micro-cluster C, as can be seen on the right in Figure 3a. Researchers using our framework can determine the aggregated user actions C_a, while the basic micro-cluster C_o is created by the actions of other users and the stream clustering algorithm and cannot be influenced.

Consequently, the resulting pattern of the artificial campaign as displayed in the monitoring tool mostly depends on how the actions of single users are simulated, as can be seen in Figure 3b: basically, for each user U_i, we define the duration d_i and the submitted text volume t_i of his action. For example, a high t_i and a short d_i will lead to a sharp increase in cluster weight, while a high t_i and a long d_i will result in a moderate increase. Additionally, the final increase level depends on the subjacent stream S in which the campaign is inserted: first, the more users produce similar content, the more likely the stream clustering algorithm will assign these posts and the artificial posts to the same cluster. Second, the more active other users are, the higher the cluster weight increase and vice versa. In contrast, the weight decrease can only indirectly be influenced by stopping the posting activities of simulated users. Then, the decrease depends on S and the fading configuration of the clustering algorithm. However, note that the exact shape of S is unknown, and thus, modeled as a blue shaded box in Figure 3b. A complementing trend of S may blur the artificial campaigns' pattern when many users post on a similar topic and vice versa.



(a) Aggregation of malicious users' activities and an original cluster of the underlying stream with similar / topic-related content.

(b) User action's patterns depend on activity duration, tweet volume, the basic stream and the clustering algorithm's configuration.

Figure 3: Composition of the cluster trend depending on the participating user's actions and the stream clustering algorithm.

Framework Evaluation

After we instantiated our flexible framework with specific detection algorithms and content creation models, we assessed our setup and the framework's general capabilities.

Experimental Setup

We used the Twitter API academic access for streaming one percent of *climate change*-related tweets in real-time. The topic is polarizing, heavily discussed online, and, thus, likely to be targeted by disinformation campaigns (Prokova 2020). These hashtags were used to filter the stream:

- **Climate change**: climate climateChange climateCrisis climateAction actOnClimate climateChangeIsReal globalWarming saveTheEarth saveThePlanet thereIsNoPlanetB climateEmergency climateJustice
- **Climate change denial**: climateLies climateDenial climateSceptics climateHoax
- **Movements**: fridaysForFuture fridays4future fffUnitedStates studentsForFuture students4future parents4future parentsForFuture extinctionR greenpeace wwf
- **People**: gretaThunberg sophiaMathur

We collected 3.6 million tweets from 1/10/2021 until 7/11/2022. To identify malicious campaigns, we used the previously described monitoring tool by Assenmacher et al. (2020b). Based on reported experiences (Assenmacher et al. 2020a; Assenmacher and Trautmann 2022), we used `textClust` with its standard configuration, i.e. automatic distance threshold adaption; we set the fading factor λ to 0.01, and generated uni- and bi-grams during preprocessing.

To generate artificial tweets, we used the 1.3 billion parameter version of `GPT-Neo` (Black et al. 2021) as implemented in the Hugging Face Transformers library[3]. As `GPT-Neo` did not receive any social media data during its pre-training, we fine-tuned the default model with our collected tweets. It received the data in the format `<username> <tweet>` within a single field separated by a space for learning to create tweets similar in sentiment and content to original user's tweets. Since, on average, in the dataset are only 400 tweets per user, `GPT-Neo` does not mimic each user's style perfectly but learns the general probability distribution of tweets. We used an 80/20 train/test split, and to differentiate distinct tweets, we added a start-of-tweet and an end-of-tweet token to the training dataset. We fine-tuned the model on a Nvidia A100 (40GB) for five epochs with a batch size of 112. The fine-tuned is freely accessible as Hugging Face model repository[4].

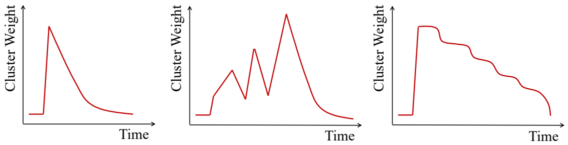
Since `GPT-Neo` has a variety of parameters, we conducted a small pre-study to find a suitable model configuration. Following the advice in the guidelines by OpenAI (2021) and von Platen (2020), we selected the parameters *temperature*, *top-k*, and the *repetition penalty* for configuration. Through scaling the raw predictions (logits), the *temperature* controls the output's randomness. We tested the configurations 0.7, 0.8, and 0.9. The parameter *top-k* influences the width of the probability mass, which in turn influences the *temperature*: only the fixed number of *top-k* tokens is considered so that the probability mass is redistributed. We checked the values 50 and 150. Lastly, the *repetition penalty* determines whether the model can repeat tokens multiple times. We test its effect by applying it with a parameter value of 2 and by deactivating it with a value of 1.

To test how our fine-tuned `GPT-Neo` model can reflect specific users' tweets, we selected a climate change denier as test subject, since, due to the distinct opinion, it is straightforward to check whether `GPT-Neo` can match it. For each configuration, we generated 1 000 tweets. In a first statistical evaluation, we tested the general tweet characteristics, i.e., the number of user mentions, inserted URLs, and the average length. Second, we asked two human ML experts to manually verify whether the tweet's content is logical, coherent, and matches the climate change denier's opinions.

Using human experts to assess the quality of machine-generated texts e.g. was done in by Hartvigsen et al. (2022), where annotators evaluated whether the automatically generated toxic language presents a potential harm to readers. Nevertheless, the automatic evaluation of artificially generated data without human annotators remains an important future work endeavor. Established NLP metrics such as `BLEU` (Papineni et al. 2001), `ROUGE` (Lin 2004) or `BERTScore` (Zhang et al. 2019) come with their own limitations: as they capture the token agreement (based on n-grams or semantic word/sentence embeddings), they are not

[3] https://huggingface.co/docs/transformers/index

[4] https://huggingface.co/Nijana/gpt-neo-1.3B-climate_change_tweets/tree/main



Stereotype I ($n = 33$) Stereotype II ($n = 12$) Stereotype III ($n = 5$)

Figure 4: Schematic display of the three different patterns that were identified by analyzing manipulation campaigns.

suitable to verify whether the produced content matches the users' opinions. Thus, in the context of this study, we only rely on the manual assessment by human annotators.

We decided on closely simulating one campaign of each stereotype we observed in our data for the final campaign creation, i.e., to adopt the number of participants, submitted tweets, and duration. To assess the effect of the automatically mass-generated tweets in detail, we only replaced the monotonous tweets of the original campaigns with `GPT-Neo`-generated tweets.

Results

The results of each phase of our proposed framework will be discussed next: the original campaign collection, the simulation with a fine-tuned and configured `GPT-Neo` model, and the recreation of the campaigns to test competitive detection algorithms. The results of each study will be discussed next.

Original Campaign Collection We identified campaigns by focusing on examining the change of `textClust`'s cluster weights over time visually in the human-in-the-loop monitoring tool. Since each cluster represents different topics discussed in the stream, a rapid weight change indicates the automated activity or unusual behavior. Further proof was given by accompanying metrics displayed, e.g., the tweet-per-user ratio or the analysis of the content. Using this approach, we identified ninety campaigns during the thirty-seven days of data collection, fifty of which were malicious. Most non-malignant campaigns were created by benign flight trackers, weather, or news bots. Some were also false positives, like the announcement that the British Queen would not attend the UN climate conference. The news created a sharp increase in tweet activity but was organic in its origin since triggered by a real-world event.

Overall, we identified three stereotypical campaign patterns, which can be inspected in Figure 4. The first consists of a single peak created by the synchronized tweet activities of several accounts within a short time. Although there are some varieties, for example, the steepness of the increase or the number of participants, thirty-three of our fifty malicious campaigns followed this general pattern. The second stereotype consists of two or three consecutive peaks created by stop-and-go tweeting activities. We collected twelve campaigns in this category. Lastly, stereotype three consists of a sharp increase in activity with prolonged posting behavior over hours. Due to the decreasing intensity of the activities, the decrease in cluster weight happens gradually. Organic campaigns triggered by a real-world event often follow this pattern, but we also found five malicious campaigns.

Campaign Simulation Using GPT A first inspection of the 12 000 tweets generated for testing configurations reveals that `GPT-Neo` generally captured the climate change denier's style and sentiment well. In Table 1, we provide original tweets on the left, next to variations on the right[5].

Original	`GPT-Neo`
@GretaThunberg There is Zero proof there is a climate crisis ZERO PROOF Just another fabricated emergency to fleece the sheep	@GretaThunberg There is Zero proof there is a climate crisis ZERO PROOF Just another fabricated emergency to strip you of your rights
@user @globalnews Ya, SO ??? None of your business ! Climate Change is a SCAM, just like your Fake Pandemic	@user No one is buying the climate crisis Climate change hoax is the next Covid
@GretaThunberg She wont recover, her mind has SNAPPED	@GretaThunberg What a hypocrite. All she does is lie and manipulate people

Table 1: Examples of original stream (left) and topic-related `GPT-Neo`-generated tweets (right).

The model generally reflects the opinion of the climate change denier we used as a test subject: it uses user mentions, and some words are written in upper case, just like in the originals. The result of the first study we conducted by automatically checking tweet characteristics can be seen in Table 2. The climate change denier did not append any URLs to the tweets, while `GPT-Neo` added randomly generated URLs to some. `GPT-Neo` probably adds URLs since in the fine-tuning dataset, URLs are attached to many tweets, and the model infers knowledge from those as well. The difference between tweets generated with a *repetition penalty* is apparent to those created without (i.e., a configuration with the values 1 and 2, respectively): tweets without *repetition penalty* contain URLs in approximately 2 % to 5 % of cases, while tweets with *repetition penalty* contain links in 10 % to 15 % of cases. Moreover, in each tweet without a *repetition penalty*, approximately two users are mentioned with a standard deviation between 1.5 and 2.4, while it is merely one user with a very low standard deviation (< 0.09) in the tweets with a *repetition penalty*. There, tweets are less than 135 characters long with a standard deviation of 77, whereas the tweets without *repetition penalty* are, on average, longer than 150 characters with a standard deviation of 55. Nevertheless, most artificial tweets are shorter than the original ones. However, while the effect of the *repetition penalty* can be seen, no such clear result emerges for the parameters *temperature* and *top-k*. Nevertheless, since the *repetition penalty* affects `GPT-Neo` to create shorter tweets with fewer user mentions and more URLs, we exclude these configurations from further consideration.

Next, we manually inspected each tweet of the remaining configurations to check the percentage of tweets being il-

[5] More examples can be found on GitHub: https://github.com/JaninaPohl/artificialcampaigns

Configuration	Perc. URLs	Avg. mentions	Avg. length
Original	0.000	2.129	180.574
0.7 / 50 / 1	0.024	2.118	161.428
0.8 / 50 / 1	0.031	2.141	154.719
0.9 / 50 / 1	0.054	1.974	154.954
0.7 / 150 / 1	0.019	2.262	163.375
0.8 / 150 / 1	0.042	1.994	157.023
0.9 / 150 / 1	0.053	1.945	154.590
0.7 / 50 / 2	0.138	1.004	134.231
0.8 / 50 / 2	0.123	1.005	115.021
0.9 / 50 / 2	0.119	1.003	119.110
0.7 / 150 / 2	0.136	1.004	132.957
0.8 / 150 / 2	0.147	1.003	114.350
0.9 / 150 / 2	0.105	1.002	120.432

Table 2: Percentage of URLs per tweet, the average number of mentions, and the average length of GPT-Neo-generated tweets per configuration in the format *temperature / top-k / repetition penalty.*

Configuration	Reflect opinion	Logic/coherence	Both
Original	0.000	0.000	0.000
0.7 / 50 / 1	0.016	0.018	0.030
0.8 / 50 / 1	0.012	0.023	0.029
0.9 / 50 / 1	0.037	0.040	0.059
0.7 / 150 / 1	0.012	0.019	0.032
0.8 / 150 / 1	0.012	0.019	0.028
0.9 / 150 / 1	0.030	0.024	0.046

Table 3: Percentages of tweets not reflecting the climate change denier's opinion or being nonsensical for each remaining configuration in the format *temperature / top-k*.

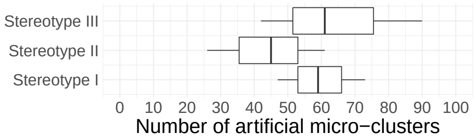


Figure 5: Analysis of the number of clusters textClust distributes the artificial campaigns into.

logical or reflecting another opinion than the climate change denier's. Results are shown in Table 3. Notably, the number of unusable tweets is relatively low for all configurations; the worst result is five percent for the configuration *0.9 / 150*. However, note that GPT-Neo had slightly more significant problems generating logic and coherent tweets than reflecting the user's opinion. When examining the column values more closely, it can be seen that the influence of the *top-k* value is not strong; the percentages of faulty tweets are mostly slightly lower for a *top-k* value of 150 than for a value of 50. Differences between the *temperature* values of 0.7 and 0.8 are not severe, just 0.5 % in the worst case. The individual percentages of 0.8 are slightly higher. Though, when considering the overall value, the configuration with a *top-k* of 0.7 performs worse. Nevertheless, there are more unusable tweets created with a *temperature* of 0.9. Consequently, the configuration of the fine-tuned GPT-Neo model used in this work will include a *top-k* value of 150 and a temperature of 0.8. No *repetition penalty* will be applied since this leads to significantly shorter tweets. With this parameter configuration, GPT-Neo created meaningful tweets reflecting the original user's style and content with a high probability.

For each of the identified stereotypes, we recreated one campaign artificially. The original cluster trend can be seen in the upper part of Figure 6. We selected a network of 120 seemingly legitimate user accounts posting news headlines in combination with a corresponding link for the first stereotype. Supposedly, the network should increase traffic to the news agency's website. Stereotype two was generated by a group of three accounts tweeting two hundred similar messages for approximately one-and-a-half hours. The user accused several news agencies of not reporting on illegal climate-damaging projects on protected Native land in the USA. Lastly, we chose a group of 160 accounts targeting Joe Biden's Twitter account for the third stereotype to urge him to hold on to certain projects despite missing support from the US senate. We recreated each campaign by copying the parameters but replacing the original tweets with ones generated with our fine-tuned and configured GPT-Neo model.

Artificial Campaign Detection The result of the recreation can be seen in the lower part of Figure 6 with the pattern of the recreated campaign plotted in red, while other clusters in which textClust also distributes artificial tweets plotted in light red. The observed sharp cluster spikes are not prevalent anymore, implying that textClust does not assign the generated tweets to the same cluster but distributes them over multiple ones. Hence, human experts would not be able to identify these artificial campaigns by monitoring the cluster weight's change over time anymore.

In the bottom-left plot, it can be seen that textClust distributed the artificial tweets over three clusters. For clarity, we do not show all clusters into which the stream clustering algorithm has distributed the artificial tweets. Instead, in the boxplots in Figure 5, an overview of the number of clusters containing artificial tweets per campaign can be seen. Depending on the stereotype, textClust integrates the GPT-Neo-generated tweets into 26 – 90 different clusters. For stereotype three and especially for stereotype one, textClust distributes the tweets over more clusters than stereotype two. This is related to the fact that our fine-tuned GPT-Neo model uses the participating users as a blueprint. Since the campaigns were created by networks of accounts instead of small groups for stereotypes one and three, the artificial campaign's content is more heterogeneous. Although GPT-Neo catches the user's style, the variety of all tweets created by one user is high. Thus, it might be more realistic in the future to use a model fine-tuned for creating different topics instead of mimicking single users to create more coherent campaigns created by a network of accounts. Nevertheless, textClust was also unable to group the tweets belonging to stereotype two into one cluster, which was generated by only a few users and, thus, is more coherent content-wise. Overall, the tweets created with GPT-Neo are too heterogeneous content-wise and currently impede textClust's ability to retrieve the original clusters.

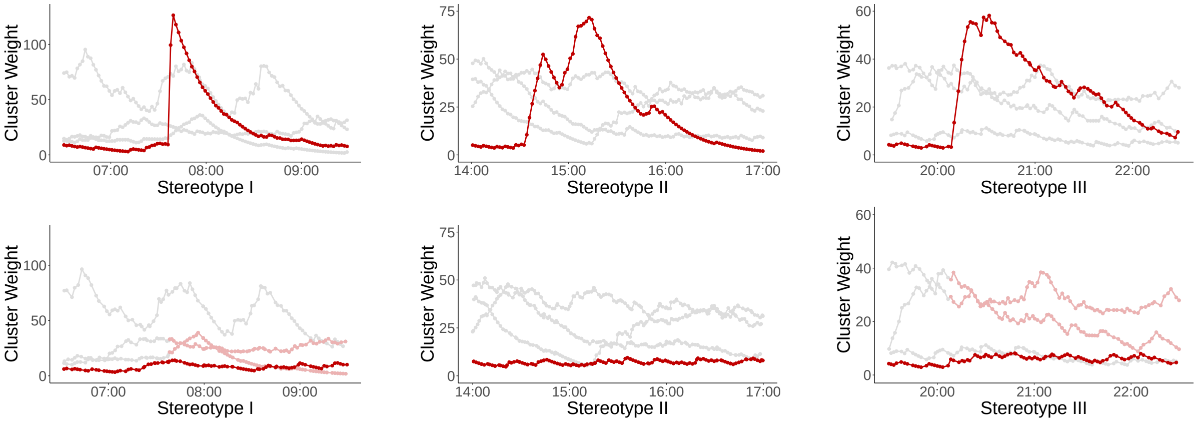


Figure 6: Artificial data augmentation impact on micro-cluster weights. Above, the original campaigns are displayed while their recreated counterparts are below with clusters in which artificial tweets are grouped into by `textClust` plotted in light red.

Ethical Considerations

In theory, malicious campaigns created automatically can be shared without violating data-sharing regulations introduced by social media platforms. However, even for these artificial contents, privacy considerations are essential. First, a failsafe must be implemented to prevent GPT from reproducing tweets as they originally occurred. Here, finding a balanced `GPT-Neo` configuration that generates more heterogeneous but meaningful tweets helps in reducing identity projections. Second, real users may be mentioned within the artificial tweets as the fine-tuned model learned from the original data. Thus, implications on individuals or organizations implicitly mentioned in the textual data may be drawn that have to be handled with care. Therefore, the artificially created campaigns that we used in this study are published together with some original examples in a pseudonymized form[6]. Third, the adversarial approach of benchmarking involves an ethical trade-off: by confronting state-of-the-art campaign detection with new challenges, their development can indeed be pushed forward. At the same time, the adapted and fine-tuned language models can be used for the generation of malignant campaigns themselves. Therefore, the benefits to the research community and the potential risks of publication must be carefully weighed when publishing either models or details about the training process.

Discussion and Future Works

We have developed a new, flexible pipeline that allows us to enrich social media data streams with artificial campaigns. Since these are based on real events that we use as blueprints and mimic real users' posts, our the artificial campaigns within our framework reflect the original ones closely. Further, the flexible approach can also be altered to create more challenging settings. Meanwhile, we gained insights into campaign executions that can be used in future works to develop more robust detection algorithms, e.g., improving our selected detection approach, since it could not recognize `GPT-Neo`-generated campaigns. Our methodology can be used to create artificial campaigns automatically, the initial data and campaign collection, analysis, and implementation are laborious, requiring more effort than streaming a dataset from an API. Although the campaigns themselves are recreated as realistic as possible, real users' reactions to this campaign cannot be modeled. In future works, self-hosted social networking services like Mastodon[7] might be used to create social media instances populated with users who agreed to participate in such artificial campaign experiments.

Also, a systematic and objective evaluation of GPT's ability to generate realistic tweets is required. We assessed the artificial tweets by examining specific features and manually inspecting their logic and coherence. As standard NLG metrics do not cover the criteria needed, in future works, a large-scale study needs to be designed that, for example, consults a larger expert group for inspection. Additionally, other types of data, campaigns, and detection approaches must be tested in future works to validate the framework's capabilities further. We revealed shortcomings of the monitoring tool we used to identify campaigns and especially the clustering algorithm `textClust`. Since, supposedly, the reason for failure is the algorithm's focus on similar words instead of synonymous words, using another word embedding approach or similarity measure might overcome this shortcoming. For example, the soft cosine similarity uses a lexicon to incorporate the meaning of words into the distance calculation. Further, the monitoring tool can be extended by models to identify artificially generated tweets. However, the overload of training and updating these models must be pondered to the benefits these metrics must provide since the main advantage of the tool is its ability to display relevant information in real-time, possibly enabling moderators to counteract the detected campaign instantly. Being able to detect and possibly react to modern mis- and disinformation campaigns becomes a more and more pressing issue as more and more interpersonal interactions have shifted from the real world to the online ecosystem.

[6] https://github.com/JaninaPohl/artificial_campaigns

[7] https://mastodon.social

Acknowledgments

The authors acknowledge support from the BMBF-funded project Hybrid (funding ref. 16KIS1531K) and the ERCIS network.

References

Alarifi, A.; Alsaleh, M.; and Al-Salman, A. 2016. Twitter Turing Test: Identifying Social Machines. *Information Sciences*, 372: 332–346.

Assenmacher, D.; Adam, L.; Trautmann, H.; and Grimme, C. 2020a. Towards Real-Time and Unsupervised Campaign Detection in Social Media. In *Proceedings of the 33rd Florida Artificial Intelligence Research Society Conference*, FLAIRS'33, 303 – 306. Miami, FL, USA: AAAI.

Assenmacher, D.; Clever, L.; Pohl, J. S.; Trautmann, H.; and Grimme, C. 2020b. A Two-Phase Framework for Detecting Manipulation Campaigns in Social Media. In Meiselwitz, G., ed., *Social Computing and Social Media. Design, Ethics, User Behavior, and Social Network Analysis*, 201–214. Springer International Publishing.

Assenmacher, D.; and Trautmann, H. 2022. Textual One-Pass Stream Clustering with Automated Distance Threshold Adaption. In *14th Asian Conference on Intelligent Information and Database Systems (ACIIDS)*. Springer International Publishing. Accepted.

Assenmacher, D.; Weber, D.; Preuss, M.; Calero Valdez, A.; Bradshaw, A.; Ross, B.; Cresci, S.; Trautmann, H.; Neumann, F.; and Grimme, C. 2021. Benchmarking Crisis in Social Media Analytics: A Solution for the Data Sharing Problem. *Social Science Computer Review*, 39(3).

Black, S.; Leo, G.; Wang, P.; Leahy, C.; and Biderman, S. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. Technical report, Zenodo.

Brown, T.; et al. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

Bruns, A. 2019. After the 'APIcalypse': Social Media Platforms and Their Fight against Critical Scholarly Research. *Information, Communication & Society*, 22: 1544 – 1566.

Bucur, D.; and Holme, P. 2020. Beyond ranking nodes: Predicting epidemic outbreak sizes by network centralities. *PLOS Computational Biology*, 16(7): 1–20.

Calabrese, A.; Bevilacqua, M.; Ross, B.; Tripodi, R.; and Navigli, R. 2021. AAA: Fair Evaluation for Abuse Detection Systems Wanted. In *13th ACM Web Science Conference 2021*, WebSci '21, 243–252. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383301.

Carnein, M.; Assenmacher, D.; and Trautmann, H. 2017. Stream Clustering of Chat Messages with Applications to Twitch Streams. In de Cesare, S.; and Ulrich, F., eds., *36th International Conference on Conceptual Modeling*, 79 – 88. Valencia, Spain: Springer International Publishing.

Chen, Z.; and Subramanian, D. 2018. An Unsupervised Approach to Detect Spam Campaigns that Use Botnets on Twitter. *ArXiv:1804.05232*.

Choraś, M.; Demestichas, K.; Gielczyk, A.; Álvaro Herrero; Ksieniewicz, P.; Remoundou, K.; Urda, D.; and Woźniak, M. 2021. Advanced Machine Learning techniques for fake news (online disinformation) detection: A systematic mapping study. *Applied Soft Computing*, 101: 107050.

Cresci, S. 2020. A Decade of Social Bot Detection. *Communications of the ACM*, 63: 72 – 83.

Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; and Tesconi, M. 2017. The Paradigm-Shift of Social Spambots: Evidence, Theories, and Tools for the Arms Race. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, 963 – 972. Perth, Australia.

Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; and Tesconi, M. 2018. Social Fingerprinting: Detection of Spambot Groups Through DNA-Inspired Behavioral Modeling. *IEEE Transactions on Dependable and Secure Computing*, 15(4): 561–576.

Cresci, S.; Petrocchi, M.; Spognardi, A.; and Tognazzi, S. 2019. On the Capability of Evolved Spambots to Evade Detection Via Genetic Engineering. *Online Social Networks and Media*, 9.

Da San Martino, G.; Cresci, S.; Barrón-Cedeño, A.; Yu, S.; Pietro, R. D.; and Nakov, P. 2020. A Survey on Computational Propaganda Detection. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI-20, 4826–4832. International Joint Conferences on Artificial Intelligence Organization.

Derhab, A.; Alawwad, R.; Dehwah, K.; Tariq, N.; Khan, F. A.; and Al-Muhtadi, J. 2021. Tweet-Based Bot Detection Using Big Data Analytics. *IEEE Access*, 9: 65988–66005.

Eichenberg, C.; Black, S.; Weinbach, S.; Parcalabescu, L.; and Frank, A. 2021. MAGMA–Multimodal Augmentation of Generative Models through Adapter-based Finetuning. *arXiv preprint arXiv:2112.05253*.

EleutherAI. 2022. GPT Neo - An Implementation of Model & Data Parallel GPT3-Like Models Using the Mesh-Tensorflow Library.

Fagni, T.; Falchi, F.; Gambini, M.; Martella, A.; and Tesconi, M. 2021. TweepFake: About Detecting Deepfake Tweets. *PLOS ONE*, 16(5).

Ferrara, E.; Varol, O.; Menczer, F.; and Flammini, A. 2016. Detection of promoted social media campaigns. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 10, 563–566. Cologne, Germany: Association for the Advancement of Artificial Intelligence.

Gao, L.; Biderman, S.; Black, S.; Golding, L.; Hoppe, T.; Foster, C.; Phang, J.; He, H.; Thite, A.; Nabeshima, N.; et al. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *arXiv preprint arXiv:2101.00027*.

Gilardi, F.; Baumgartner, L.; Dermont, C.; Donnay, K.; Gessler, T.; Kubli, M.; Leemann, L.; and Müller, S. 2021. Building Research Infrastructures to Study Digital Technology and Politics. *PS: Political Science & Politics*, 1–6.

Hartvigsen, T.; Gabriel, S.; Palangi, H.; Sap, M.; Ray, D.; and Kamar, E. 2022. ToxiGen: A Large-Scale Machine-Generated Dataset for Adversarial and Implicit Hate Speech Detection.

Ippolito, D.; Duckworth, D.; Callison-Burch, C.; and Eck, D. 2020. Automatic Detection of Generated Text is Easiest when Humans are Fooled. In *Proceedings of the 58th Annual Meeting of the ACL*, 1808 – 1822. Online: ACL.

Lee, K.; Caverlee, J.; Cheng, Z.; and Sui, D. Z. 2014. Campaign Extraction from Social Media. *ACM Trans. Intell. Syst. Technol.*, 5(1).

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: ACL.

Lotito, Q. F.; Zanella, D.; and Casari, P. 2021. Realistic Aspects of Simulation Models for Fake News Epidemics over Social Networks. *Future Internet*, 13(3).

Microsoft. 2020. T-NLG: A 17-billion-parameter language model by Microsoft. https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/. Accessed: 2022-03-19.

Naseem, U.; Razzak, I.; Khushi, M.; Eklund, P. W.; and Kim, J. 2021. COVIDSenti: A Large-Scale Benchmark Twitter Data Set for COVID-19 Sentiment Analysis. *IEEE Transactions on Computational Social Systems*, 8(4): 1003–1015.

OpenAI. 2021. Guides: Completion – Learn How to Generate or Manipulate Text, Including Code. https://beta.openai.com/docs/guides/completion. Accessed: 2022-03-14.

Orabi, M.; Mouheb, D.; Al Aghbari, Z.; and Kamel, I. 2020. Detection of Bots in Social Media: A Systematic Review. *Information Processing & Management*, 57(4).

Pacheco, D.; Hui, P.-M.; Torres-Lugo, C.; Truong, B. T.; Flammini, A.; and Menczer, F. 2021. Uncovering Coordinated Networks on Social Media: Methods and Case Studies. *Proceedings of the International AAAI Conference on Web and Social Media*, 15(1): 455 – 466.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2001. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, 311. Philadelphia, PY, USA: Ass. for Computational Linguistics.

Pasquetto, I.; Swire-Thompson, B.; and Amazeen, M. A. 2021. Tackling Misinformation: What Researchers Could Do With Social Media Data. *Harvard Kennedy School Misinformation Review*.

Pritzkau, A.; Winandy, S.; and Krumbiegel, T. 2021. Finding a Line Between Trusted and Untrusted Information on Tweets Through Sequence Classification. In *2021 International Conference on Military Communication and Information Systems*, ICMCIS'21, 1–6. The Hague, Netherlands.

Prorokova, T. 2020. The Madhouse Effect: How Climate Change Denial is Threatening Our Planet, Destroying our Politics, and Driving us Crazy. *International Journal of Environmental Studies*, 77(3): 537 – 538.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving Language Understanding by Generative Pre-Training. Technical report, OpenAI.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140): 1–67.

Robeer, M.; Bex, F.; and Feelders, A. 2021. Generating Realistic Natural Language Counterfactuals. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 3611–3625. Punta Cana, Dominican Republic: Association for Computational Linguistics.

Samper-Escalante, L. D.; Loyola-González, O.; Monroy, R.; and Medina-Pérez, M. A. 2021. Bot Datasets on Twitter: Analysis and Challenges. *Applied Sciences*, 11(9): 4105.

Twitter. 2021. Developer Platform Terms: Developer Policy.

Varol, O.; Ferrara, E.; Menczer, F.; and Flammini, A. 2017. Early Detection of Promoted Campaigns on Social Media. *EPJ Data Science*, 6.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 6000–6010. Red Hook, NY, USA: Curran Associates Inc.

von Platen, P. 2020. How to Generate Text: Using Different Decoding Methods for Language Generation with Transformers. https://huggingface.co/blog/how-to-generate. Accessed: 2021-12-17.

Wardle, C. 2018. The Need for Smarter Definitions and Practical, Timely Empirical Research on Information Disorder. *Digital Journalism*, 6(8): 951 – 963.

Wei, J.; and Zou, K. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. *arXiv preprint arXiv:1901.11196*.

Wolf, T.; et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Online: Association for Computational Linguistics.

Wu, T.; Ribeiro, M. T.; Heer, J.; and Weld, D. 2021. Polyjuice: Generating Counterfactuals for Explaining, Evaluating, and Improving Models. In *Proceedings of the 59th Annual Meeting of the ACL and the 11th International Joint Conference on Natural Language Processing*, 6707–6723. Online: Association for Computational Linguistics.

Xia, F.; Li, Y.; Yu, C.; Ma, H.; and Qian, W. 2014. BSMA: A Benchmark for Analytical Queries Over Social Media Data. *Proceedings of the VLDB Endowment*, 7(13): 1573–1576.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.