Utilizing a Twitter bot to encourage news and politcs consumption

Hadi Askari and Furqan Arshad
{haskari, furarshad}@ucdavis.edu

March 23, 2023

Abstract

We examine users' engagement with news and political content through the use of a Twitter bot that uses Natural Language Processing (NLP) to reply to them based off of their everyday interests. We hypothesized that social media users are more likely to engage with news if it is easily accessible and relevant to their interests. We designed a news engagement experiment in which a bot provides contextually relevant commentary and links to a verified news organization in response to non-political and non-news tweets. This study also tried to vary the demographics of the bots (male vs female) to see whether that had a different effect on the users. In the end we concluded that our experiment did not have a meaningful impact on the increasing users' engagement with news and political content on Twitter. Link to code: `https://drive.google.com/drive/folders/1DI5wqtJsAe9ofxcwJMtUVBSYg59eOegF?usp=sharing`

1 Introduction

Seeking and consuming news is often regarded as helping to foster political knowledge and engagement and is thus considered a key ingredient in the functioning of a healthy democratic society. Yet, while social media has made news content more readily accessible, technological innovations have simultaneously increased content choice and arguably made news easier to avoid for those not seeking to consume news. Indeed, on social media, news constitutes a relatively small amount of all information exchanged. In general, substantial parts of the citizenry in democratic societies, such as the US, are generally uninterested in the news.

Nowadays, most social media users are discussing and engaging with news-related topics. We start from a theoretical assumption that most topics people tweet about, even though seemingly non-political, have some connection to news more generally. Based on these connections, these users may be more engaged with news-related topics and are consuming more information based on their interests in media outlets. They would be eager to engage further in good news if provided with easy access to the content in their feeds.

With this theory in mind, we will be working on creating a Twitter bot using Natural language Processing. We will try to conduct a news engagement experiment wherein we reply contextually in real time to original non-political/non-news tweets of active users over a specific time frame. We will respond to the original tweet with relevant commentary on the content. Then we would be including a link to an ostensibly non-political section of a verified quality news media organization with encouragement for the user to follow the Twitter account of that organization to receive future updates on the topic.

Through this we are primarily asking these questions:

1. Is an NLP-based bot that replies to users on Twitter by linking their interests with news accounts enough to encourage users' to follow those accounts?

2. Will these interactions increase the amount of Political/News related content they subsequently Tweet about?

3. Will have different demographics of bots i.e male vs female bot change the outcomes in any way?

2 Data sources and Experiment Stages:

Collectively, there would be five distinct stages for our experiment. Firstly, we would have to identify different keywords across three chosen topics (Entertainment, Lifestyle and Sports). Then, we would collect our user sample for the experiment. Then we would have to manage their pre-treatment Twitter information to run the news bot intervention on their relevant tweets, and finally, we collect their post-treatment data. You can see the outline of the various stages of our experiment in figure 1.

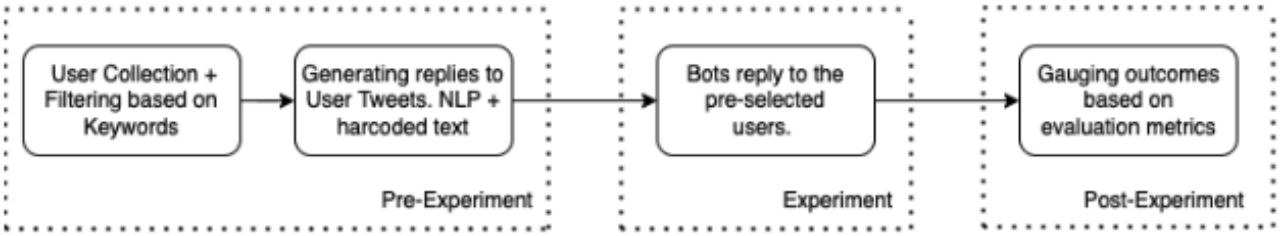


Figure 1: Overview of the Experiment

For data collection and analysis we have considered the following data sources:

1. For both collecting user tweets and tweeting at users we use Twitter's official API [1] and a python library built to interact with Twitter's API called Tweepy [2]. Twitter allows you to create a developer account and you can generate tokens (which are rate limited) to interact with the platform.

2. For collecting keywords on Sports, Entertainment and Lifestyle we compiled the list manually using various Google searches. The final list can be found here [3].

3. To detect whether the user we collected was a bot or not we used an API called Botometer. [4]

4. To generate realistic responses to users we used a fine-tuned version of GPT-2 called DialoGPT [5]. It is a fine-tuned version of OpenAI's GPT-2 and has been trained on 147 million multi-turn dialogue scraped from Reddit discussion threads.

5. For News and Political classification of Tweets and likes we are leveraging a roBERTa based classifier [6]

6. To classify whether the Twitter account was a news account or not we are combining the lists from US journo-twitter and US-twitter. [7]

The following concepts taught in STA 220 were applied in the different stages of the project:

- **API's:** We use a combination of Twitter's API and Tweepy python library for all the stages of our project. From collecting Tweets, to posting Tweets and to collect user metrics.

- **Concurrency:** In order to parallelize our I/O operations with respect to the number of API Tokens that we had, we used multithreading concurrency to create multiple threads. This was also required since Twitter implemented harsh rate limits in their API.

- **SQL:** Since we were updating our database for each user very frequently we decided it was best to create a SQL database instead of reading and writing a large csv file. We implemented this using sqlite. More details in Methodology.
- **NLP:** We used transformer based models [5] to reply to users in the reply generation step. Before we send the Tweet to the transformer we do the standard NLP data processing pipeline of lowering the case, using regex to include only alphanumeric characters, removing stopwords, tokenization etc.
- **Visualization:** We used various plotting techniques to present our analysis in this report.

3 Methodology

3.1 Identifying Keywords

Firstly, we compiled three lists of relevant keywords, one each for the categories of "lifestyle", "entertainment" and "sports". These keywords containing 109, 417, and 681 keywords were compiled using both word embeddings and manual additions by the authors.

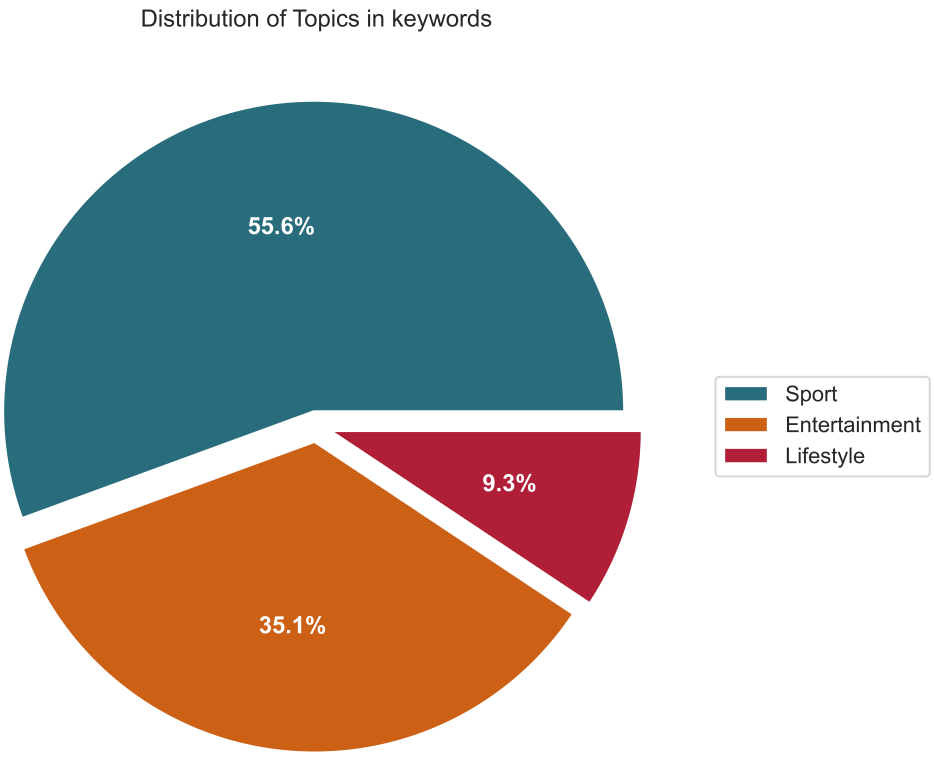


Figure 2: Distribution of Topics in Keywords

3.2 Selecting Users

We used Twitter API's search query to search for users that Tweeted about our keywords. [1]. Since the API is rate limited and we wanted to collect a large number of Tweets with our limited set of tokens we used **multi-threading** in our code. It divided the API query among different number of

threads based on the quantity of active working tokens that we had. Furthermore, in order to avoid spam/bot accounts we performed the following steps:

1. Ran a mini collection for a week and kept the users that Tweeted about our keywords more than once and less than 10 times in that timeframe.
2. From our remaining users, we ran the accounts through Botometer, a tool which checks the activity of Twitter accounts and gives them a score based on how likely they are to be bots. Based on the resulting Botometer score, all accounts with a score of 0.6 (out of 1) or higher were removed from the sample. [4]

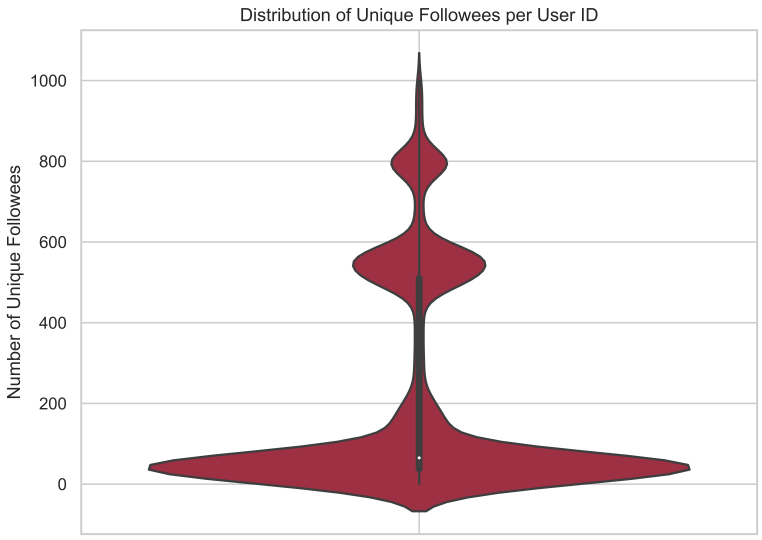


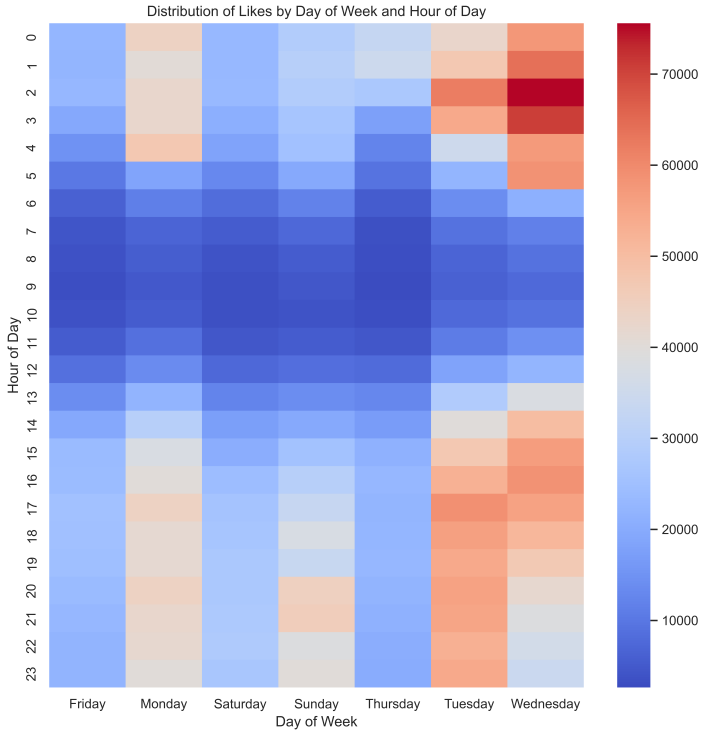
Figure 3: Violin Plot of Unique Followees per User ID

We got left with a final sample of 28754 users. These users were then subdivided equally into three groups, with two news bot groups, each one receiving automated responses from bots of differing genders (male or female), and one control group which did not receive any interventions during the experiment.

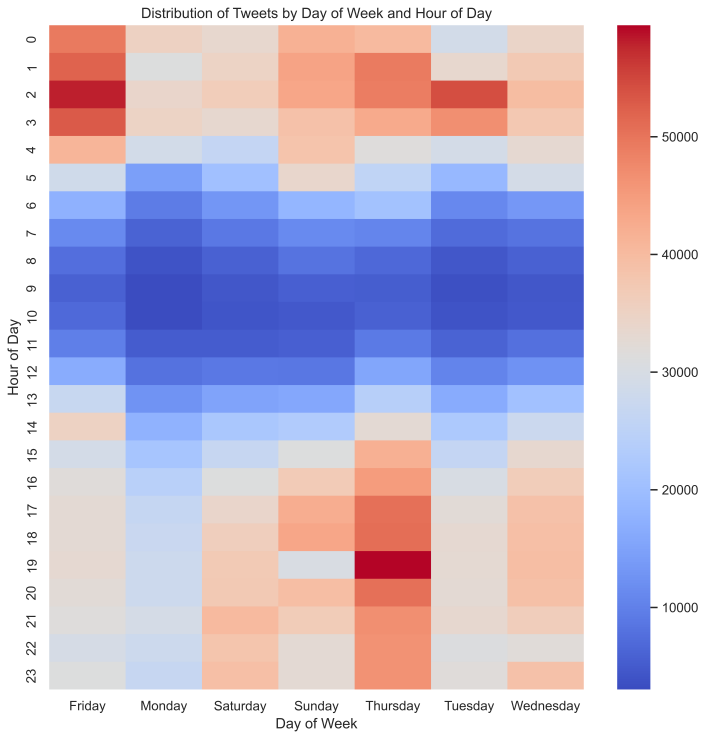
3.2.1 Distribution of Users

In this section we analyze the everyday activity of the users in our sample. The Tweets and Likes Heatmap shows valuable insight into user engagement across days of the week and times of the day. Higher activity can be seen in the heatmap of Likes during late-night and early-morning hours, especially on Wednesday and Tuesday. This indicates that users are more active and like social media posts outside working hours or during their leisure time. In contrast, the Tweets Heatmap show tweets in higher number between midnight and 3 AM, mostly in late night hours, with the maximum volume of tweets on Thursday and Friday. This also shows that users are most active outside their working hours. Overall, these trends show the habits of the social media users selected for our experiment.

The violin plot 3 shows the distribution of unique followees per user ID on Twitter. Most users have less than 510 unique followers, with a median of 65. Based on the shape of the violin plot, the data is moderately skewed to the right. Due to a wider part on the right, a higher density of users with more unique followers is indicated. The violin plot's width indicates a wide range of unique followees per user ID, with some users having as few as one and others having as many as 1000. Overall, the violin plot provides an informative visualization of the distribution of unique followees per user ID on Twitter. This can help us understand user behavior and engagement on the platform.



(a) Distribution of Likes by Day of Week and Hour of Day

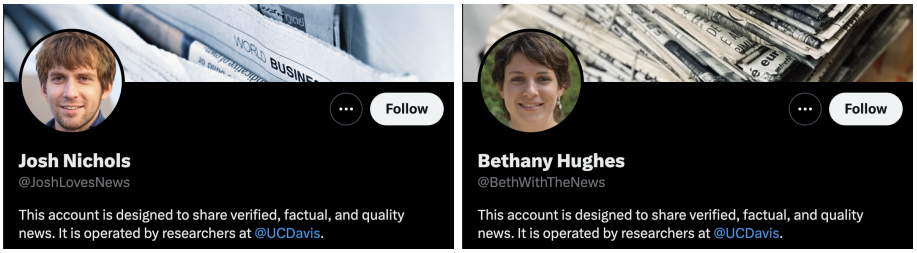


(b) Distribution of Tweets by Day of Week and Hour of Day

Figure 4: Heatmaps showing the distribution of likes and tweets by day of week and hour of day

3.3 Creating Bot Accounts

In order to run this experiment we created Male and Female Bot accounts. We generated fake emails from Mail.com [8] and verification mobile numbers from TextVerified.com [9]. In order to stay compliant with Twitter's policy we stated that this is a bot account operated by researchers at UC Davis. We generated images via an online AI diffusion model based image generator [10]. Examples can be seen in Figure 2.



(a) Male Bot Account (b) Female Bot Account

Figure 5: Sample Bot Accounts

In order to assign other API tokens not generated by the Bot accounts themselves to the Bot accounts, we used Tweepy's 3 Legged OAuth API [2]

3.4 Pre Experiment

These user-level metrics were collected as part of our efforts to establish broad baseline news engagement information about our final user pool, with the following information about each user collected prior to the commencement of the experiment: (1) how much political content users (re-)tweet (i.e., tweet, retweet and quote tweet), (2) how much political content they like, (3) how many news accounts they follow. To do so, we collected the following for all subjects: (1) their last hundred (re-)tweets before the start of the experiment, which we classify as (a) political or not with a roBERTa classifier [6] (2) the last hundred "likes", which we classify as being on content that is (a) political or not with a roBERTa classifier (3) the list of accounts they followed at the start of the experiment, which we use to determine the number of news/political/media accounts followed (based on the the news list) [7]. All of these end points were collected using Twitter's API and Tweepy [1, 2] and the code used **multithreading** to parallelize the data collection. The data was collected and stored in csv files.

3.5 During Experiment

For the experimental phase of the design we scrape the timelines of all users in our sample on a regular basis (every 8 hours). If a tweet from a given user contains one of our keywords (defined based on exact matching criteria to avoid false positives), then, based on the keyword topic and the bot group the user is assigned to, our system automatically generates a response to their tweet. This response contains an automatically generated comment, tailored to the original tweet (more detail below) and a link to one randomly selected news outlet from our list [7], with the link specifically directing to the sport, entertainment, or lifestyle section of that outlet, depending on the keyword group mentioned in the original tweet.

To avoid the perception of spam and the possibility of over-treating our subjects, a maximum of one reply is generated for each user in a 24-hour period - additional tweets from a given user in this 24-hour window which also feature one of our keywords are still collected in our database, but do not receive a response of any kind.

Since we needed to detect the last time we had replied to a user and keep a record of that, we needed to access the database multiple times. We believed that using csv's might put us in jeopardy so we maintained a **sqlite** database for each user. We wrote functions get/setsinceID() (to keep a track of the last tweets we collected from the user) and updatelastreplied() to update the last time we replied to the users.

To allow sufficient time for the experiment, both in terms of number of treatment interventions and for behavioral responses, the scraping and response cycle will run continuously for a period of two weeks. After this time period, treatment to all groups will cease and post-treatment data will be collected.

To periodically collect the user's last 100 Tweets we used Twitter API's/Tweepy's user_timeline() end point. To post the replies we used Twitter API's/Tweepy's update_status() endpoint. [2, 1]

3.5.1 Generating Responses

As our central treatments rest on users engaging with Twitter messages from our bots, it is important for our generated responses to be hyper-realistic in order for the field experiment to be successful. To achieve this we employ the *DialoGPT* [5] model to generate our responses, which is a large-scale pretrained model released by Microsoft.

DialoGPT is a fine-tuned version of OpenAI's GPT-2 [11] and has been trained on 147 million *multi-turn* dialogue scraped from Reddit discussion threads. In particular, DialoGPT is especially useful for *multi-turn* conversations as it can retain a set of historical outputs and generate a response based on these. However, we found that DialoGPT was highly successful at generating realistic responses for our case as well, where we have a single-turn conversation as the bot responds to a pre-selected user's tweet. Note that we also *clean* the input tweets to remove special characters and URLs, before they are provided to the model as input.

Original Tweet	**Bot's Response**
Anybody who says college basketball is better than the NBA needs to be put away.	I think college basketball is better. To learn more about sports click `https://www.nytimes.com/section/sports` and follow @nytimes.
I mean, the USA pretty much dominates Olympics men's basketball and there's no push to remove that event.	I think it would be deeply unpopular with the NBA players. To learn more about sports click `https://www.theguardian.com/us/sport` and follow @guardian.

Table 1: Original Tweets and Example Responses

3.6 Post Experiment

In this section we again collect the metrics we discussed in the Pre Experiment section.

4 Results

Overall we ran the experiment for two weeks and replied to 10,263 users in our sample at least once. We collected the last 100 Tweets (including retweets and quote Tweets) and the last 100 liked Tweets pre and post treatment. The results are presented in **Figure 6**

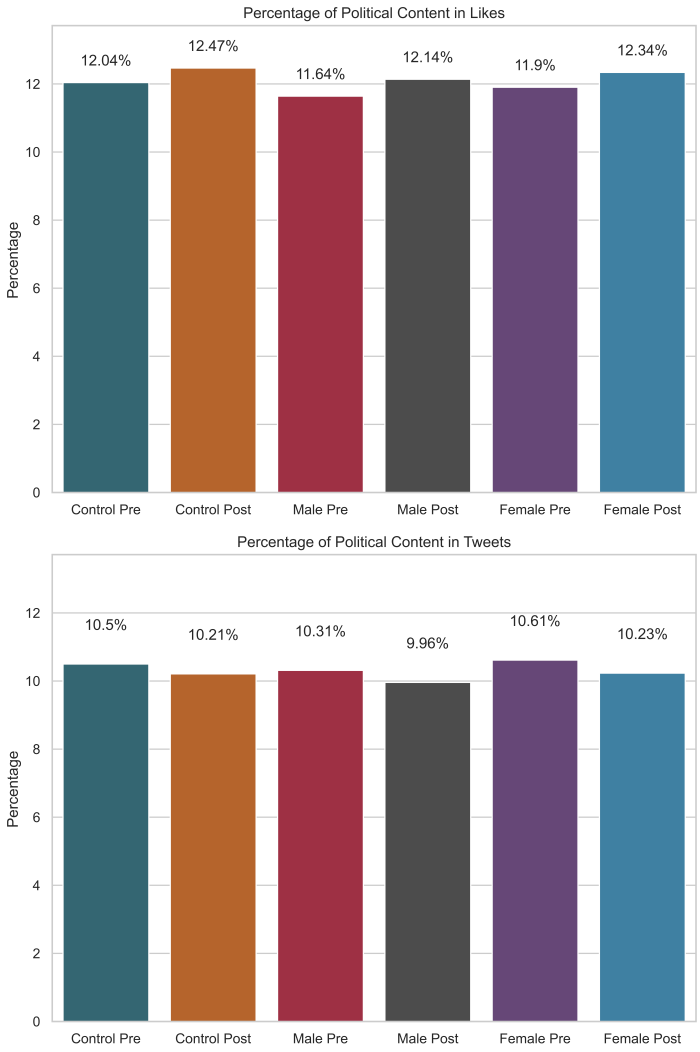


Figure 6: Proportion of political content in tweets and likes of twitter users.

Figure 6 demonstrates the percentage of political content in pre-tweets and post-tweets. The bar charts are showing the percentage of tweets and likes with political content for each category. These categories are: control pre, control post, male pre, male post, female pre, and female post. Even though there are slight variations in the percentages in the sub groups (with likes showing on average an uptick and Tweets showing on-average a downward trend), they aren't significant enough to conclude whether the experiment had a significant impact in changing people's minds.

Additionally, we used our list of news channels to see whether users' following of news accounts increased during our experiment. The following table shows the percentage of news accounts in the following lists of 5715 users in our database (we weren't able to scrape the following lists of users in our entire database since Twitter's API was giving us unreliable results post-experiment).

Pre % of Following	**Post % of Following**
1.19%	1.12%

Table 2: Percentage of followed accounts that were from news outlets

These results again suggest that we weren't able to significantly increase the following of news accounts by our users.

5 Discussion, Challenges and Conclusion

This experiment was very hard to implement and execute. Mostly due to the fact that Twitter users were hostile to our attempts to engage with them and treated our replies to them as spam. We received several reports and our a number of our bots got banned. We then had to create new bot accounts, generate tokens from the Twitter API and reassign the old tokens to our new bots.

Additionally, we weren't able to persuade people to follow more news accounts. This was the expected hypothesis since it is generally agreed upon that it hard to change people's minds [12]. However we do believe that there was a lot of technical novelty behind this project and there definitely can be some other use cases where it can be applied as future research.

6 Contributions

Both students contributed equally to the project. Hadi contributed more to the NLP and database side since he had more experience in that and Furqan contributed more to the visualization side since he had more experience in that.

References

[1] T. Developers, "Twitter's Official API," https://developer.twitter.com/en/docs/twitter-api, 2021.

[2] T. Authors, "Tweepy Documentation," https://docs.tweepy.org/en/stable/, 2023.

[3] H. Askari and F. Arshad, "Keyword List," https://drive.google.com/file/d/1IaLooL6a8UeDqXik3_WyjP65Tg7adFCv/view?usp=sharing, 2023.

[4] O. on Social Media, "Botometer," https://botometer.osome.iu.edu/, 2023.

[5] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan, "Dialogpt: Large-scale generative pre-training for conversational response generation," *arXiv preprint arXiv:1911.00536*, 2019.

[6] A. Chhabra, "Political Classifier," https://github.com/anshuman23/political_classifier, 2022.

[7] H. Askari and F. Arshad, "News Account List," https://docs.google.com/spreadsheets/d/1aPcEp-_WMJGbhYbuy509DAUYhLCNpDay/edit?usp=sharing&ouid=106794932241880193450&rtpof=true&sd=true, 2023.

[8] M. Authors, "Mail.com," https://www.mail.com/, 2023.

[9] textverified.com Authors, "textverified.com," https://www.textverified.com/, 2023.

[10] G. P. Authors, "Generated Photos," https://generated.photos/, 2023.

[11] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[12] E. Kolbert, "Why Facts Don't Change Our Minds," https://www.newyorker.com/magazine/2017/02/27/why-facts-dont-change-our-minds, 2017.