The (In)comprehensive Nature of Political Ad Definitions

Christopher Choy
Carnegie Mellon University

Jack Gardner
Carnegie Mellon University

Soha Jiwani
Carnegie Mellon University

Rakshit Naidu
Carnegie Mellon University

Lujo Bauer [1]
Carnegie Mellon University

Abstract

We study the accuracy of political advertisement identification on major online platforms through comparison against the platforms' existing ad-tag patterns. If definitions for political advertisement identification are inconsistent within or across platforms, the validity of transparency reports and the effectiveness of platforms at preventing misinformation becomes suspect.

To determine the accuracy of platform ad identification, we train six machine learning models on a corpus of already classified ads from Meta (formerly Facebook) and Google. We test the model against a separate portion of the ad corpus including a set of ads from each platform and a mixed set from both platforms. We also manually classify advertisements to be political or non-political and compare them with our model classifications. We find that companies consistently categorize ads according to the definitions they use. We also show that the model trained on Google advertisements exhibits higher accuracy when compared to manual advertisement classifications than the model trained on Meta's advertisements.

1 Introduction

Technology companies publish transparency reports that provide researchers and users with an understanding of the data collected and used on their platforms. Political advertising activities are a major feature of these reports, and they provide a means to review the political ads issued by campaigns over time. However, these reports only make transparent the ads that the platform considers political [11] [10]. The consequence of this is that transparency reports created by companies may only capture a sliver of the ads that meet the specific definitions of political ads that differ from company to company. Additionally, companies may still misidentify ads that meet their definitions. Thus, readers may see that transparency reports do not reflect their own browsing experience, or that the reports do not provide an accurate, comprehensive view of political advertising on the platform.

In this paper, we use web-measurement techniques to study ads that are tagged as "political" by Google and Meta in order to understand whether the definitions employed by the technology companies issuing these transparency reports fail to capture content that is not tagged as political but that individuals may consider political. Specifically, we scrape political and non-political ads from these platforms to construct a data set of ads on which we train machine learning models. We first show that our model is able to accurately classify the ads run on these platforms according to their definitions and then compare the results of our model against a small subset of manually classified ads to understand if the definition the model represents (via the data on which it is trained) align with what content general users may consider political. This process revealed that the ads tagged as political across platforms align differently with the ads people consider political. In particular, ads classified as political by our model aligned well with the ads we manually tagged as political, but the model classifications did not match an additional set of the ads manually tagged as political, suggesting that people have a broader conception of what constitutes a political ad than what our model captures.

As our results suggest that definitions used by these technology companies may misidentify some ads as non-political that people consider political, the categories into which ads are placed may be misleading. Additionally, since there is not an agreed upon definition for a political ad [26], media platforms create their own definitions, which could cause the content of ads in similar categories to differ across media platforms. By training our model on ads from a specific media platform and testing the ability for it to classify ads on another platform, we find some evidence to suggest that the definitions employed by different platforms leads to inconsistencies in the categories into which ads are placed across platforms.

[01] Professor advisor as part of the "Privacy, Policy, Law, and Technology" course at Carnegie Mellon University, Fall 2021

2 Background and related work

While media platforms employ their own definition of what constitutes a political ad, this is for good reason. As the type of content contained in a political ad varies across the medium through which ads are served, definitions for political ads change accordingly. Moreover, as people differ in terms of the content they regard as political, it is necessary for organizations to define for themselves a set of characteristics that are specific to political ads. In the case of technology companies, such characteristics provide a foundation for them to consistently categorize content as political and enable them to monitor the content that is posted on their platforms [14].

Defining political ads and having a means to identify them on media platforms also helps users be better aware of the content they are viewing and how it differs from other user-generated content. While political ads differ from "standard" content on media platforms, they have also proved to influence users' actions and have the ability to spread information to a broad audience at speed.

Below is our review of prior work on how definitions of "political" vary across platforms (2.1), the influence of political advertisements on voters (2.2), and an examination of prior work on developing machine learning models in the context of text classification (2.3).

2.1 Political advertisement definitions

The Federal Election Commission (FEC), Google, Meta, and Twitter all have different definitions of a political ad. The FEC classifies a political ad as any ad that needs a disclaimer [5]. The types of ads that need disclaimers are "any public communication made by a political committee", and while there is a lot of information readily available from the FEC regarding television, radio, and broadcast advertising, there is very limited information regarding digital advertising. This leads to each platform being responsible for deciding what constitutes a political ad.

Meta defines political ads as "made by, on behalf of, or about a candidate running for public office, a political figure, a political party or advocates for the outcome of an election to public office" or "about any election referendum, or ballot initiative, including 'go out and vote' or election campaigns" [3]. From this definition, we see that Meta focuses both on the overall content of the advertisement as well as who runs the ad, while the FEC definition focuses on who is running the advertisement.

Google refers to ads that could be described as political as "election ads". These ads are defined to include ads for "current officeholder or candidate for an elected federal or state office, federal or state political party, or state ballot measure, initiative, or proposition that qualifies for the ballot in a state", which is a fairly limited definition [11]. Election ads do not include "ads for products or services, including promotional political merchandise like t-shirts, or ads run by news organizations to promote their coverage of federal or state-level election campaigns, political parties, candidates, or current elected officeholders or state-level ballot measures, initiatives, or propositions" [13]. In Google's case, there is no mention of local campaign ads being considered "election ads" which differentiates this definition from Meta's and the FEC's. Twitter defines political ads "as content that references a candidate, political party, elected or appointed government official, election, referendum, ballot measure, legislation, regulation, directive, or judicial outcome." [27]

While platforms define methods to classify the content on their platforms, they are also responsible for enforcing their definitions by reviewing the content on their platform and ensuring that issuers of content on their platform are following appropriate guidelines. New York University's Cybersecurity for Democracy group released a study that found Facebook overcounted 55% of the ads run on their platform from July 2021 to February 2021 in the US when they should not have been considered political [17].

2.2 How do political advertisements change voter behavior?

Prior research demonstrates that advertisements can influence political behavior in voters. In a 2005 study, Brader demonstrated that fear-triggering ads with ominous music and bleak images of violence caused viewers to seek more information and remember it better [2]. Additionally, they found ads that sparked enthusiasm with upbeat music and images of flags could actually reduce a viewer's desire to investigate the candidate's platform. Depending on a campaign's goals, political advertisements can be tuned to elicit desired behavior in voters and increase or decrease vote shares for a particular candidate.

However, the influence of political ads does not always translate to increased voter share [6]. Throughout the 2016 election, a comprehensive study that adjusted for primary vs. general election, battleground vs. non-battleground states, viewer partisanship, and whether the ads were attacking or promotional, conducted 59 unique experiments on 34,000 people and found that while ads could shift candidate favorability, viewers decisions on who they would ultimately vote for did not change.

Social media platforms also have the potential to significantly influence political campaigns. Advertisements by malicious actors can take the form of bots–users with fake profiles typically managed in bulk–that leverage the power of word-of-mouth to spread information and disinformation without the oversight and context that binds traditional advertisements. A New York Times article notes that bots were prevalent during the 2016 election and linked to Russian influence by the CIA, NSA, and FBI "with high confidence" [23]. On Meta (known as Facebook at the time), these bots were reported to have purchased over $100,000 in ads [25]. While platforms like Meta and Twitter may enjoy inflated usage metrics and resulting profits, they also have an incentive to shut down bot accounts as they erode user trust. This can be a difficult challenge as intelligent bot profiles are hard to distinguish from actual users. Monitoring such online behavior is important work as both advertisements and word-of-mouth can have significant effects on political outcomes. Relatedly, Scheffler et al. note that social media can be leveraged by state actors to “have a weakening effect on strong democratic regimes” and “a radicalizing effect on weak democratic regimes’ [21].

2.3 Ad classification techniques

Chakrabarti et al. discuss how the relevance and context of ads is linked to both user experience and the revenue that ads generate [4]. This “relevance” is calculated as a similarity score between the website where the ad appears and the ad being analyzed. Data on ad impressions and click-data can further improve the calculation of relevance scores, and this paper offers a means to extend our work by examining the number of times a political ad has appeared or was clicked by a user and including those factors into our model estimation.

A key problem in classifying online advertisements is standardizing the types of data collected from ads issued by different advertisers. Pera et al. created a centralized database of ads using a new machine learning-based software tool [16]. The study addresses challenges around (i) classifying the domains from which ads are served, (ii) collecting data from ads structured in a variety of ways, and (iii) analyzing data from ads that fall into different categories and tagging those ads with corresponding keywords or attributes. The authors use Support Vector Machines (SVMs) and Decision trees to tag ad domains and to categorize an ad based on ad attributes, respectively, to maintain a database of ads. This study serves as a useful guide for modeling relevant keywords and finding attributes to classify ads into categories.

In contrast to the type of methodology we hope to implement, Sukul et al. capture data at the browser and server levels as opposed to ad-level data. They created a unique web bot [24] to collect information about video and image ads that are shown when a user watches a video on YouTube. Information collected includes the source URL of the ad, the type of the ad derived from the title of the ad, and the unique computer ID that collects ad metadata when users watch videos on YouTube. The web bot is also tested using a variety of browsing profiles that are constructed based on differences in subject-matter interest and geographic location to collect data from ads that represent diverse browsing behaviors.

Sanzgiri et al. uses a pre-trained machine learning model to collect image ads and compare the category associated with the collected ad to a historical list of topic categories listed on a website [20]. The model assigns predefined categories generated by an advertisement auditing team to the ads. As some ads include more sensitive content than others, the model is tasked with ensuring that the content of an ad served to a website aligns with the types of content previously served on that same website. While our study does not focus on categorizing ads based on image attributes, many ads are image-based and it is useful to understand how ad classifications methods vary based on the type of ad content.

3 Methodology

Our study methodology is broken into two discrete portions which we detail here. The first phase (3.1) of our study was to build an ad corpus of text ads with their corresponding platform’s tags of political or generic (non-political). The second phase (3.2) was then using that data to train and test several different machine learning models and evaluate the models with a manually tagged data set (3.3).

The goal of our methodology was to build a machine learning model that could accurately represent what text content informed a platform’s decisions to label an ad as political. Going forward, these models could go beyond looking for certain keywords in the ad text and consider how words were strung together. Additionally, we could run our models on novel data sets to see how well they agreed with the labels of other platforms and users.

3.1 Building an Ad Corpus

We use web-scraping techniques to collect ads from Google and Meta. On each media platform, we collect a set of political ads and a set of generic (non-political) ads. We also focus on collecting text-based advertisements. This made them easier to collect and train on in a shorter period of time. It also meant that our models could see the full context of the advertisement when training. Automating collection of ads with the context of whether that ad has been labeled as political or not by the platform posed additional challenges. We detail below how we overcame these challenges for each platform and their quirks. In the end, we collected 3011 advertisements from Google and 3504 advertisements from Meta, which gave us a total of 6515 advertisements to divide into sets for training and testing our models.

Platform	Political	Generic	Total
Google	2105	906	3011
Meta	3039	465	3504
Total	5144	1371	6515

3.1.1 Meta (Facebook)

Meta provides a user interface called "Facebook Ads Library" [9] for searching ads on its various platforms including Facebook, Instagram, and Messenger. The Ad Library places content into four categories: (1) Issues, Elections, or Politics, (2) Housing, (3) Employment, and (4) Credit. It also allows for filtering ads on type: video, image, or text. This made it great for our purposes, but Meta requires identity verification before users can gain access to the API for the library. For two of our researchers, this meant providing images of a government ID, entering an authentication code physically mailed to a US address, and completing a notarized affidavit of identity.

However, gaining access to the Facebook Ads Library API was not enough. The API endpoints only allowed for collecting ads in the political category. This allowed us to collect our thousands of text ads from Meta tagged as political.

Since Meta made it difficult to collect non-political ads from the Facebook Ads Library, we decided to build a web scraper in Python using the Selenium library [22]. Selenium allows us to instance the HTML of a page in a Chromium browser which can then be snapshot and parsed. However, Meta likely uses ReactJS [19] in the back end for their Facebook Ads Library web page. This meant that instead of easy-to-parse HTML with detailed class names, identifiers, and HTML elements, the HTML of the page was largely a massive tree of div elements with dynamically assigned class names. By leveraging XPATH expressions, we were able to parse through the tree to extract individual ad texts from the page and tag them based on the parameters we set in the HTTP request of the page we requested. The exact code for this can be viewed in our appendix.

3.1.2 Google

Google provides a transparency report which includes a listing of political advertisements organized by country and advertiser campaign. These transparency reports provide individuals and governments with information on company practices and provide methods to examine the data these companies collect on a broad scale. Although transparency reports are not standard practice among technology companies, they are a useful means to both respond to government requirements to provide detail on companies' actions and provide the public with a resource to better understand the data collected on them [15].

Google provides no API for dynamically collecting these ads en masse. Thus, we build a web scraper to collect the ads. Unlike Meta, the HTML content was well formatted with specific HTML elements and thus fairly straightforward to collect. The code for this can be found in our Appendix.

To collect non-political ads from Google, we could not scrape from their transparency reports because it only includes political ads. To solve this, we scraped Google search results where US political ads must contain a "Paid for by" disclosure [12]. By searching for these disclosures, we could determine if the ad was considered political or not by Google. For these search results, we used words that we expected would give potentially–but not necessarily–misidentified political ads in the result. For example, we used the queries "can I vote?", "texas guns", "fox news", and "medicare". Our full list of search queries can be found in the appendix.

3.1.3 Data Aggregation

We aggregated all of our collected ads from all platforms into one comprehensive data set with the following rows: AdText, Category, Platform, and Keywords. We then split this data set into two child data sets, one for training the model and one for testing the model. We ensured that both children had the same ratios of political to non-political ads as well as the same ratios of Google to Meta ads. Within each sub-category (e.g. "political ads from Google"), each child randomly selected half of the ad entries without replacement.

Next, we semi-randomly sub-sampled from the testing data set 100 ad entries for manual classification by our researchers. The selection was controlled to have 25 of each of the sub-categories: Google political, Google non-political, Meta political, Meta non-political. However, the selection of the specific ads within those sub-categories were random. It was important we took this manual classification set from only the testing data so that when we ran the machine learning models on it, we could be certain the model had not seen this ad before and thus just regurgitating what Meta or Google had already assigned it. We wanted the labeling process to be independent for both the researchers and the models.

3.2 Machine Learning for Ad Text Classification

To process the data in order to run our machine learning models, we implement Bag-of-Words (BoW) [1] and Term Frequency-Inverse Document Frequency (TF-IDF) [18]. BoW involves tokenizing the words in the training set via one-hot-encoding to extract relevant features from the document. In BoW, we optimize for the *vocabSize* (vocabulary size) parameter to measure the frequency of words appearing in our ads data. However, the problem with such an approach is that high frequency words influence the score of the entire model, i.e. more common words are assigned a larger weight than other words, and these frequent words may not be of any substance. Hence, TF-IDF was proposed to rescale the frequencies based on how often the words appeared across all documents. TF scores the frequency of the word in that document while IDF scores how common or how rare that word appears across the documents. We calculate TF and IDF in the following fashion,

$$TF(t,d) = \frac{n(t)}{d} \tag{1}$$

$$IDF(t) = \frac{N}{n(t)} \tag{2}$$

	AdText	Category	Platform	Keywords
0	Pete Buttigieg 2020 I Vote for Pete Today 2/29 I Find Your Polling PlaceAd p...	political	Google	NaN
1	Empowering people with creative ideas to succeed. Start your free trial. Wit...	non-political	Google	NaN
2	Do you love working with people and educating them? Do you want to be a lead...	non-political	Facebook	environment
3	Brothers Auto Parts is looking for a person to help us sell used auto parts....	non-political	Facebook	environment

Figure 1: Example table of one of our training datasets.

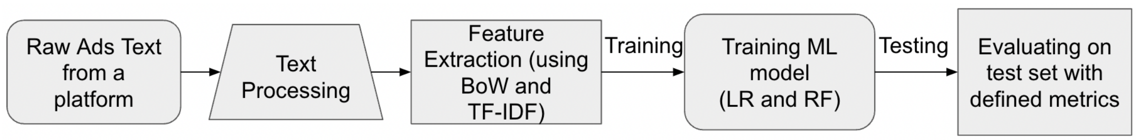


Figure 2: Our pipeline.

where $n(t)$ refers to the number of times the term t has appeared in document d, N refers to the total number of documents present in a set of documents U.

We then use two types of classification models: Logistic Regression (LR) and Random Forest (RF). Logistic Regression is a classification method where we compute the probability of the input with respect to a binary output i.e. 0 or 1 [7], and Random Forest is an ensemble for classification tasks which operates by utilizing multiple decision trees at a time [8]. For each of these models, we report F1-score and accuracy measures. However, as the data set we have is imbalanced, the F1-score (instead of accuracy) is a more suitable metric to evaluate our models. Additionally, we use our manually tagged ads data set and compare how the models predictions align with the manually tagged data set for each set of training data.

3.3 Manual Classification of Advertisements

We took a sub-sample of 100 advertisements from both Meta and Google. Two of our researchers then labeled whether they found the ad to be political, not political, or uncertain. This labeling was done independently of each other's labels, the preexisting platforms' labels, and knowledge of what platform the ad came from or what keywords were used to find it.

After independently labeling, the two researchers compared their responses and found they came to a consensus 93% of the time about whether an ad was political or non-political. From this classification, we learned that humans generally agree on whether a text ad is political or not. This high agreement also meant we could consider the manual classifications as an acceptable representative of general user definitions when comparing them to the platforms' and models' labels.

On average, our human labels agreed with Google's labels 95% of the time and with Facebook's labels 92% of the time.

4 Results

The accuracy of the six classifiers is presented in Table 3. First, we see that the models trained on a single platform's data are able to accurately predict the classifications on that same platform. Google's classifiers had 99% - 100% accuracy on categorizing advertisements from its own platform, and Meta's classifiers had an accuracy ranging from 95%-97% on classifying ads on its own platforms. Next, our results show that the Google classifiers retained 90% - 91% of accuracy when classifying the manually tagged data, but Meta's classifiers dropped to 51% - 58%, suggesting that the platforms' definitions align differently with the manual classifications from our researchers.

We also compute the confusion matrices of the many combinations displayed in Table 3. The true positives, true negatives, false positives and false negatives over the manual data set are our key results and are shown in in Appendix C. We see that the model trained on the mixed data (Google + Meta) yields the best results with 96% of classifications being true positives–predicted labels match manual labels.

We observe that the Google models performs the best across all the testing sets (highest accuracy and F1-score as shown in Tables 3 and 4), which suggests that their definition of a political ad aligns best with that of other platforms and the manually generated classifications. Conversely, models trained on Meta's data show decreased accuracy and F1-score on the manually tagged set. We also find that the Random Forest (RF) models consistently performs better than Logistic Regression (LR) models in terms of both accuracy and F1-scores.

5 Discussion

In our work, we have learned about the definitions media platforms employ to classify political ads as well as how they

Model	Google accuracy	Meta accuracy	Mixed accuracy	Manual accuracy
Google-LR	0.99	0.96	0.96	0.90
Google-RF	1.00	0.80	0.89	0.91
Meta-LR	0.70	0.95	0.81	0.51
Meta-RF	0.70	0.97	0.83	0.58
Mixed-LR	0.86	0.97	0.95	0.75
Mixed-RF	0.94	0.98	0.99	0.88

Figure 3: Accuracy of classifiers on the same platform, a different platform, and manually tagged ads

Model	Google f1	Meta f1	Mixed f1	Manual f1
Google-LR	1.00	0.98	0.97	0.90
Google-RF	1.00	0.87	0.92	0.90
Meta-LR	0.82	0.97	0.89	0.66
Meta-RF	0.82	0.98	0.90	0.69
Mixed-LR	0.91	0.98	0.97	0.79
Mixed-RF	0.96	0.99	0.99	0.88

Figure 4: F1 Scores of classifiers on various platforms

are enforced. We discuss the implications of these definitions (5.1) alongside the limitations of our work (5.2) and avenues for future work (5.3) in the remainder of this section.

5.1 Media platforms' definitions

Media platforms identify political ads according to the definitions they employ. As individuals may hold their own definitions for the content they consider political, definitions from media platforms, and the way they tag advertisements on their platforms, may not align with users' opinions. The benefit of a definition to categorize content on a platform is also only useful to the extent that media organizations take measures to enforce their definitions.

To the extent that media organizations do enforce their definitions, we use the ads they tag to train a machine learning model to gauge whether the model is able to identify the set of content that falls under their definitions. The fact that our models perform well at matching the labels given by the platforms they trained on suggests that the definitions used by these platforms are at least consistent.

While employing a common set of criteria is useful for media organizations, users may benefit from additional means to understand whether the content they are seeing could be political. For instance, platform content moderators could introduce a means to signal that the content on their platform may in some way be considered political even though it does not clearly meet the criteria in their definitions. Such content could have a message denoting that platforms have not yet identified the content as political while letting users know that what they may be viewing differs from other content on the platform. In the process, users would be more aware of the content they are being served, and media platforms could use this method as a means to cope with their inability to accurately classify the content posted on their platform.

5.2 Limitations

We had very few non-political advertisements compared to the number of political advertisements. In total, we collected 5144 political ads and and 1371 non-political ads from both Google and Meta. Since we have such a high disparity in the numbers for political and non-political advertisements, the bias in our data may influence the accuracy of the classifications our machine learning model produces.

Additionally, only two of our team members manually classified advertisements, and the variety of conceptions people hold with regard to the content they consider political is no different for this process. As our team members discussed the process of tagging political content, we too found it hard to employ a common set of criteria to classify advertisements. After tagging ads on our own, we reviewed them together to agree on a classification. In our discussion it was clear that both team members tended to focus on the general content of a political ad and did not focus on the mention of individuals in a political position or political organizations if the ad text did not otherwise involve political content. While our manual classification differs from media platforms processes in that we focused on the overarching message of the ads we reviewed instead of, for example, the issuer of an advertisement, other people may take a different approach to classifying ads.

By collecting only the ads that media platforms identify as political, our analysis does not include ads that may meet a platform's criteria but that they have not identified. By missing this set of content, our machine learning model may classify ads as political since it recognizes that the content of an ad matches a platforms definition, even though it may not be properly reflected in our data.

In dividing our data into training and testing data sets, we semi-randomly selected which ads went into which data set. This was good at improving our model because it would see a greater variety of ads. However, because the method we used for collecting ads from both Facebook Ads Library and Google leveraged search terms, this meant that our models were seeing a recurrence of words in the testing set that the had already seen in the training set even if the ads as a whole were different. Unfortunately, this means our models may have simply associated select words like "biden" and "america" as political without considering the context surrounding those independent words. Given more time, it may be interesting to see how well our models performed when data sets were divided on keyword as well as platform and political classification.

5.3 Future Work

Our work would benefit from obtaining manually classifications from a group of people from a diverse set of backgrounds that resemble the demographic makeup of the United States. Doing so would provide us with a more general understanding of what people consider a political advertisement, and we would be better equipped to evaluate the classifications obtained from our machine learning model to see how those classifications compare classifications from a broader set of individuals. An alternative approach to improving manual classifications could involve consulting politicians or political science researchers to identify a broad set of criteria that could be used to classify political advertisements. Reviewing ads with specific criteria in mind could improve the consistency with which individuals tag ads, but it is still an unanswered question as to whether criteria could bias individuals away from classifying content based on what they believe is political.

By conducting an ad review process similar to those used by media platforms, we could improve the validity of our data set and increase the accuracy of our results. Understanding the processes that media platforms employ would help us better understand the methods they use to classify political content and how they limit the ability for ads to provide political information without being categorized as such. We could then work to identify why ad review processes do not identify certain content even though it meets platforms' definitions and then use these ads to let the set of content in our data better align with the definitions media platforms use.

In addition to collecting ads that platforms do not identify, we may also be able to improve the accuracy of our results by obtaining a more representative set of advertisements in our data set. As our data contain a larger set of political than non-political ads, we could increase the number of non-political advertisements that we collect to improve our model's ability to recognize this content.

Additionally, we could extend our analysis framework to other media platforms and types of media content (e.g., image- or video-based ads). Doing this would help us better understand how media platforms political ad definitions differ and let us understand how types of political content vary in terms of media vehicle in which they are presented. Further, by comparing the results of our model trained on ads from other media platforms, we could provide more comprehensive results on how a platform's definitions align with our manual classifications.

6 Conclusion

Given the prevalence of political advertisements on media platforms, the large amount of funding driving them, and the influence they can have on voter opinions or behavior, they are an important part of the digital media ecosystem. In our work we show that it is possible to train a machine learning model to classify advertisements on a media platform with accuracy and that the definitions employed by media platforms may not align with individuals' understanding of political content.

Acknowledgments

We would like to thank Professor Lujo Bauer for his guidance throughout the project and Professor Aleecia M. McDonald for her initial discussions with us that gave us the idea for this project. We would also like to thank Laura Edelson and the researchers working on New York University's Cybersecurity for Democracy effort for their guidance and work that inspired us throughout our project.

References

[1] Wisam Abdulaziz, Musa M. Ameen, and Bilal Ahmed. An overview of bag of words; importance, implementation, applications, and challenges. pages 200–204, 06 2019.

[2] Ted Brader. Striking a responsive chord: How political ads motivate and persuade voters by appealing to emotions. *American Journal of Political Science*, 49:388 – 405, 04 2005.

[3] Facebook Business Help Center. About ads about social issues, elections or politics. https://www.facebook.com/business/help/167836590566506. [Online; accessed 11-October-2021].

[4] Deepayan Chakrabarti, Deepak Agarwal, and Vanja Josifovski. Contextual advertising by combining relevance

with click feedback. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, page 417–426, New York, NY, USA, 2008. Association for Computing Machinery.

[5] Federal Election Commission. Advertising and disclaimers. https://www.fec.gov/help-candidates-and-committees/advertising-and-disclaimers/. [Online; accessed 9-December-2021].

[6] Alexander Coppock, Seth J. Hill, and Lynn Vavreck. The small effects of political advertising are small regardless of context, message, sender, or receiver: Evidence from 59 real-time randomized experiments. *Science Advances*, 6(36):eabc4046, 2020.

[7] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.

[8] Adele Cutler, David Cutler, and John Stevens. *Random Forests*, volume 45, pages 157–176. 01 2011.

[9] Facebook. Ad library. https://www.facebook.com/ads/library/?active_status=all&ad_type=political_and_issue_ads&country=US&media_type=all. [Online; accessed 10-December-2021].

[10] Facebook. How are ads about social issues, elections and politics identified on facebook? https://www.facebook.com/help/180607332665293. [Online; accessed 9-December-2021].

[11] Google. Political advertising on google – google transparency report. https://transparencyreport.google.com/political-ads/region/US. [Online; accessed 11-October-2021].

[12] Google. Political content. https://support.google.com/adspolicy/answer/6014595.

[13] Google. Political content-advertising policies help. https://support.google.com/adspolicy/answer/6014595?hl=en#zippy=%2Celection-ads-in-the-united-states. [Online; accessed 11-October-2021].

[14] Tara Jacoby. Why are you seeing this digital political ad? no one knows! https://www.vox.com/recode/2020/9/29/21439824/online-digital-political-ads-facebook-google. [Online; accessed 10-December-2021].

[15] Rob Pegoraro. Tech companies are quietly phasing out a major privacy safeguard. https://www.theatlantic.com/technology/archive/2019/09/what-happened-transparency-reports/599035/.

[16] Maria S. Pera, Rani Qumsiyeh, and Yiu-Kai Ng. Web-based closed-domain data extraction on online advertisements. *Information Systems*, 38(2):183–197, 2013.

[17] Victor Le Pochat, Laura Edelson, Tom Van Goethem, Wouter Joosen, Damon McCoy, and Tobias Lauinger. An audit of facebook's political ad policy enforcement. 2021.

[18] Juan Enrique Ramos. Using tf-idf to determine word relevance in document queries. 2003.

[19] React. https://reactjs.org/docs/getting-started.html. A JavaScript library for building user interfaces.

[20] Ashutosh Sanzgiri, Daniel Austin, Kannan Sankaran, Ryan Woodard, Amit Lissack, and Sam Seljan. Classifying sensitive content in online advertisements with deep learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 434–441, 2018.

[21] Guy Schleffer and Benjamin Miller. The political effects of social media platforms on different regime types (summer 2021). http://dx.doi.org/10.26153/tsw/13987.

[22] Selenium. https://www.selenium.dev/documentation/. Selenium automates browsers. That's it!

[23] Scott Shane. The fake americans russia created to influence the election. https://www.nytimes.com/2017/09/07/us/politics/russia-facebook-twitter-election.html. [Online; accessed 9-December-2021].

[24] Adisak Sukul, Baskar Gopalakrishnan, Wallapak Tavanapong, and David A.M. Peterson. Online video ad measurement for political science research. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2132–2140, 2017.

[25] New York Times. Fake russian facebook accounts bought \$100,000 in political ads. https://www.nytimes.com/2017/09/06/technology/facebook-russian-political-ads.html. [Online; accessed 9-December-2021].

[26] New York Times. What ads are political? twitter struggles with a definition. https://www.nytimes.com/2019/11/15/technology/twitter-political-ad-policy.html. [Online; accessed 9-December-2021].

[27] Twitter. Political content. https://business.twitter.com/en/help/ads-policies/ads-content-policies/political-content.html. [Online; accessed 11-October-2021].

Appendix

(A) Web Scraping Code

Attached here are relevant snippets of our web-scraping code. The full code for the Google and Facebook Ads Library scrapers can be found at https://github.com/cchoy96/ads-scapers. Chris aims to cleanup the code and eventually port it into an open-source library for usage by other researchers frustrated by Google and Meta.

Google Transparency Report Scraper

Given a list of Google political advertiser IDs, which can be obtained from their transparency-report-bundle csvs, this code loads that report filled with ads for each of those advertisers and scrapes all the ads it can see from the page and outputs it to a csv of our design.

```
def scrape(advertiserId):
    def build_query(advertiserId):
        baseurl = "https://transparencyreport.google.com/political-ads/advertiser/{aId}?
            campaign_creatives=start:{start};end:{end};spend:;impressions:;type:3;sort:3
            &lu=campaign_creatives"
        start = 1527638400000
        end   = 1635897600000
        return baseurl.format(aId=advertiserId, start=start, end=end)

    options = Options()
    options.headless = True
    options.add_argument("--window-size=1920,1200")
    adTexts = []
    with webdriver.Chrome(options=options) as driver:
        wait = WebDriverWait(driver, 10)
        driver.get(build_query(advertiserId))
        for _ in range(10): # need to render several times to collect all elements
            html = driver.execute_script("return document.body.outerHTML;")
            soup = BeautifulSoup(html, "html.parser")
        ads = soup.find_all('text-ad')
        adTexts = [ad.getText().strip().encode("ascii","ignore").decode("ascii")
            .replace(',','') for ad in ads]
    return adTexts

def scrape_advertisers(advIds, newFile=False):
    count = 0
    mode = 'w+' if newFile else 'a'
    with open(outpath, mode) as f:
        writer = csv.writer(f)
        if newFile:
            writer.writerow(['AdText', 'Category', 'Platform'])
        for aId in advIds:
            adTexts = scrape(aId)
            print(aId, ':', len(adTexts))
            count += len(adTexts)
            for ad in adTexts:
                writer.writerow([ad, 'political', 'Google'])
```

Google Search Results Scraper

Given a text file filled with search queries, this scraper Googles each of those queries and scrapes all ads inserted into the search results. A new instance of Chromium is started for each query because we found that more ads would be shown for the first searches done in a browser session compared to any subsequent searches in that session. This was our anecdotal experience for getting around Google's ad rate limiting.

```
def scrape_search_ads(queries, newFile):
    options = Options()
    options.headless = True
    options.add_argument("--window-size=1920,1200")
    writeToDisk([], None, newFile) # initialize output file

    for query in queries:
        with webdriver.Chrome(options=options) as driver:
            wait = WebDriverWait(driver, 10)
            driver.get("https://google.com/ncr")

            ## Load Page ##
            driver.find_element(By.NAME, "q").send_keys(query + Keys.RETURN)
            wait.until(presence_of_element_located((By.CSS_SELECTOR, "h3")))
            driver.execute_script("window.scrollTo(0, 200);") # just adding some human behavior
            time.sleep(2)

            ## Scrape ads from page ##
            ads = [] # list of tuples (adText: String, isPolitical: boolean)
            adBlock = driver.find_elements(By.XPATH, "//*[@id='tads']//div[@data-text-ad='1']")
            print("Found {n} ads for query: {q}".format(n=len(adBlock), q=query))
            for ad in adBlock:
                political = "Paid for by" in ad.text
                adText = ad.find_element(By.XPATH, "./div/div[2]").text.strip()
                   .encode("ascii","ignore").decode("ascii").replace('\n',' ').replace(',','')
                ads.append((adText, political))
        writeToDisk(ads, query)

def writeToDisk(ads, query, initialize=False):
    mode = 'w+' if initialize else 'a'
    with open(outpath, mode) as f:
        writer = csv.writer(f, delimiter=',')
        if initialize: # write header
            writer.writerow(['AdText', 'Category', 'Platform', 'Keywords'])
        for adText, isPolitical in ads:
            category = 'political' if isPolitical else 'non-political'
            writer.writerow([adText, category, 'Google', query])
```

Facebook Ads Library Scraper

```
AD_TYPE = {
    'all': 'all',
    'political': 'political_and_issue_ads',
    'housing': 'housing_ads',
    'employment': 'employment_ads',
    'credit': 'credit_ads'
}

def build_query(q, type=AD_TYPE['political']):
    getVars = {'country':'US', 'sort_data[direction]':'desc',
        'sort_data[mode]':'relevancy_monthly_grouped'}
    getVars['active_status'] = 'all'
```

```
    getVars['search_type'] = 'keyword_unordered'
    getVars['media_type'] = 'none'
    getVars['ad_type'] = type
    getVars['q'] = q
    return "https://www.facebook.com/ads/library/?" +
        urllib.parse.urlencode(getVars, encoding='utf-8')

def fb_scrape(keyword, ad_type):
    ad_texts = set()
    options = Options()
    options.headless = True
    options.add_argument("--window-size=1920,1200")
    with webdriver.Chrome(options=options) as driver:
        WebDriverWait(driver, 10)
        driver.get(build_query(keyword, ad_type))

        # Scroll to the bottom a bunch to load older months
        for _ in range(10):
            time.sleep(3)
            driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")

        # Grab all ads on page
        months = driver.find_elements(By.XPATH,
            '//*[@id="content"]/div/div/div/div[3]/div[1]/div/div/div[position()>3]')
        for month in months:
            try:
                ads = month.find_elements(By.XPATH, './div[3]/div[1]/div')
                for ad in ads:
                    text = ad.find_element(By.XPATH, './div/div[3]/div/div/div[2]').text
                        .strip().encode("ascii","ignore").decode("ascii")
                        .replace("\n"," ").replace(",", '').strip()
                    if text:
                        ad_texts.add(text)
            except Exception as e:
                print("[WARN] Something went wrong")

            n = len(ad_texts)
            print("\tAds scraped: ", n)
            if n > 500: break  # just to avoid potential memory issues
    return ad_texts
```

Data aggregation

Here we list some portions of code used in aggregating our ads data and dividing it into testing and training sets. Our "clean duplicates" code ensured our model would only see each unique ad text once so that all ads in a category (political or non-political) would be equally weighted. The split data functions shows how we semi-randomly sub-sampled without replacement for the training and testing data sets.

```
def clean_duplicates():
    df = pd.read_csv(outpath)
    before = len(df.index)
    df.drop_duplicates(subset=['AdText'], keep='first', inplace=True)
    print("Dropped {n} duplicate rows.".format(n=before-len(df.index)))
    df.to_csv(outpath, index=False)
```

```
def split_data(df):
  '''Returns semi-randomly sampled subsets for training and labeling.'''
  def halflen(df):
    return int(len(df.index)/2)
  pdf = df[df.Category == 'political'].sample(frac=1).reset_index(drop=True)
  npdf = df[df.Category == 'non-political'].sample(frac=1).reset_index(drop=True)
  train = pdf.head(halflen(pdf)).append(npdf.head(halflen(npdf)))
  label = pdf.tail(halflen(pdf)).append(npdf.tail(halflen(npdf)))
  return (train, test)

# Split Google Data
g_train, g_test = split_data(all[all.Platform == 'Google'])
g_train.to_csv(train_goog_path)
g_test.to_csv(test_goog_path)

# Split Facebook Data
fb_train, fb_test = split_data(all[all.Platform == 'Facebook'])
fb_train.to_csv(train_fb_path)
fb_test.to_csv(test_fb_path)
```

(B) Search Queries and Keywords

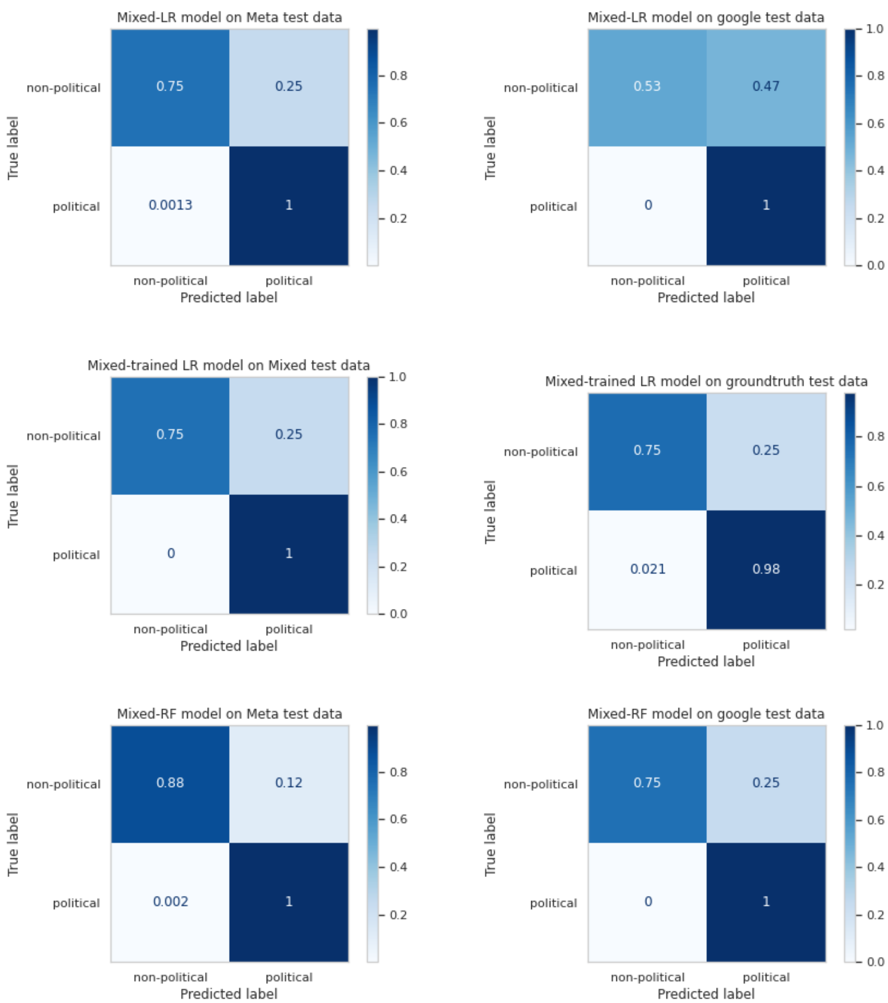
This is a comprehensive list of all search queries we input into Google and Facebook Ads Library for scraping ads.

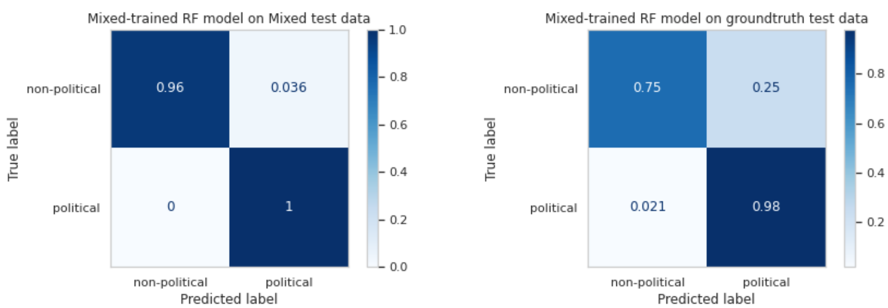
- abortion
- abortion options
- aetna providers near me
- america
- apartments for rent in pittsburgh
- bahamas vacation
- bernie sanders
- best food near me
- biden
- build a website
- campaigners for hire
- can I vote?
- cars
- carvana
- cheap apartments
- chef jobs nearby
- chinese food delivery
- cleaners for hire
- cooking
- coronavirus
- covid19 remedies
- dating websites
- disease specialists nearby
- doctors near me
- economy
- election ads
- environment
- fleshlights
- flower delivery nearby
- food delivery
- fox news
- google ads database
- green businesses
- guns
- gyms near by
- health care
- houses for sale
- how to learn react
- incel
- infrastructure
- jill biden
- job listings
- jobs
- jobs near me
- joe biden
- law advice
- lawyers near me
- learn angular
- loan sharks
- loaners near me
- match.com alternatives
- mcat study guide
- medicaid
- medicare
- nearby clinics
- nearby thanksgiving food
- need a car loan

- pizza near me
- political ads
- political campaign team for hire
- political consultant for hire
- polls near me
- poster printing
- privacy
- python courses online
- safest cars
- shelves for sale
- shooting classes texas
- show me ads
- single women near me
- student loans payment
- sushi delivery
- tax consultants near me
- tinder alternatives
- trump
- turbotax
- US politics
- vote today
- white supremacy
- yoga balls for sale

6.1 (C) Confusion Matrices

Confusion matrix plots for models trained with the mixed (Google + Meta) data. The top four plots display the Logistic Regression (LR) model results while the next four show the Random Forest (RF) results.





Mixed-trained RF model on Mixed test data
non-political
political
True label
0.96
0.036
0
1
non-political
political
Predicted label
1.0
0.8
0.6
0.4
0.2
0.0
Mixed-trained RF model on groundtruth test data
non-political
political
True label
0.75
0.25
0.021
0.98
non-political
political
Predicted label
0.8
0.6
0.4
0.2