Who was deplatformed where?

Master's Semester Project

Marie Reignier Tayar

Manoel Horta Ribiero
Prof. Bob West

June 11, 2021

1 Introduction

With the rise of hate speech and incitement to violence on social media, many platforms take the decision to ban (temporarily or permanently) the user(s) behind such behaviour, as a way to prevent them to communicate and share their ideas with the public. Recently, the deplatforming of former U.S. president Donald Trump after the end of his term brought this subject into the spotlight.

Deplatforming raises important ethical, legal and societal questions, such as "is deplatforming effective, impacting the targeted user(s), or does it give them more visibility instead?" The objective of this project is to create a precise and large-scale dataset related to deplatforming for various platforms, in order to then be able to study the impact of deplatforming.

We start by presenting the data we use to generate such dataset. Then, we describe the algorithm that we apply to the data, followed by the results it produced. Afterwards, we present a preliminary analysis of the effect of a ban on the online attention that the entity receives, and discuss its limits. Finally, we suggest further work on this dataset and the deplatforming effect analysis.

2 Data

We collect the information of who was deplatformed where from Reddit, a website of social news aggregation, web content rating, and discussion. We use a corpus consisting of all Reddit posts from 2010 to 2020, which corresponds to approximately 1.2 billion entries [1]. These posts can be scraped using the pushshift API as demonstrates in the code. For each post, we are only interested in the following attributes:

— *title*: the title of the post. Often short, it is the only thing users see before clicking on the post. It has at most 300 characters and doesn't contain hyperlinks.
— *date*: date of the post's first submission.
— *ID*: unique identifier of the post, generated by Reddit.
— *score*: number of upvotes (positive votes by Reddit members) minus the number of downvotes (negative votes).
— *nb_ comments*: number of comments under the post.

With this data and the algorithm described in Section , we produce a dataset containing a 'deplatforming table', associating an entity, a platform from which this entity was banned, the date of the deplatforming and additional attributes concerning the posts from which the information was extracted.

3 Method

We use a **bootstrapping for entity extraction** algorithm inspired by the 'Quootstrap' algorithm [1]. We now explain how this algorithm, applied to the titles of the Reddit posts in the corpus, allows us to retrieve a set of entity-platform pairs (ent, plat). It is an iterative algorithm, and at each iteration $i \geq 0$, we define $\mathcal{S}_i$ as the set of currently retrieved entity-platform pairs, and $\mathcal{P}_i$ as the current set of known patterns used to extract these pairs. We start with a set of patterns $\mathcal{P}_0$. One iteration of the bootstrapping algorithm consists of:

1. **Pair extraction: (Section 3.1)** Using the current set of known patterns $\mathcal{P}_i$, we extract all pairs that match at least one of the patterns in the corpus to form $\mathcal{S}_i$. This step is divided in the following sub-steps: **pattern matching, platform extraction, entity extraction, pair filtering**

2. **Pattern inference: (Section 3.2)** All occurrences of the newly found set $\mathcal{S}_i$ of entity-platform pairs are searched in the corpus to infer the new set of patterns $\mathcal{P}_{i+1}$. We have the following sub-steps: **pattern extraction, pattern filtering**.

After n iterations of the bootstrapping algorithm, we link all the retrieved entities in $\mathcal{S}_n$ with a Google Knowledge Graph identifier (Section 3.3) and perform a final aggregation and filtering (Section 3.4).

3.1 Pair extraction

Pattern matching: The patterns in set $\mathcal{P}_i$ have at their extremity a placeholder for the entity `<ent>` and one for the platform `<plat>`. The platform can either be first: `[<plat> * <ent>]` or last: `[<ent> * <plat>]`. For example, patterns can be `[<plat> suspends <ent>]` or `[<ent>`

[1]Value computed with the Reddit unique ids, assuming that they are always continuous and consecutive

`was banned from <plat>]`. The first step for the pair extraction is to extract all elements from the corpus which match at least one of the patterns in $\mathcal{P}_i$ (case-insensitive).

Platform extraction: For a title t that matches a pattern $p \in \mathcal{P}_i$, we extract the word *plat* at the position of the placeholder <plat>. If we continue with our previous example, we have a pattern p: `[<plat> suspends <ent>]` and the title $t1$ = 'Twitter suspends Donald Trump' and $t2$ = 'The school suspends all basketball games' we would extract $plat1$ = 'Twitter' and $plat2$ = 'school'.

Entity extraction: For a text t that matches a pattern $p \in \mathcal{P}$, we extract a **group of words** *ent* at the position of the placeholder <ent> such that they all consecutively start with a capital letter. If <ent> is at the beginning of the pattern, we take all the consecutive words that start with a capital letter, starting from the rightmost word. If <ent> is at the end, we do the same, but with the leftmost word. If the rightmost (in the first case) or the leftmost (in the second) word doesn't start with a capital letter, we discard t. For example, with the same pattern as above, the text $t1$ = 'Twitter suspends Donald Trump', $t2$ = 'The school suspends all basketball games' and $t3$ = 'Facebook suspends Mr. John Doe', we would have $ent1$ = 'Donald Trump', $ent3$ = 'Mr. John Doe' and we would discard $t2$.

Pair filtering: For a text t that matches a pattern $p \in \mathcal{P}_i$, and a pair entity-platform (ent, plat) extracted at the two previous steps, we apply automatic and manual selection criteria to decide if we keep the pair in our dataset or discard it. For the automatic filtering, we first pre-process the entity group of words by only keeping nouns that are not pronouns. If the entity is not empty after this step, we apply another filter. We keep the pair (ent, plat) if the platform plat is in a predefined list of social media platform, extracted from a Wikipedia list of social networking services (case-insensitive). If a pair passes the automatic filtering step, we then perform a manual filtering step:

— We remove all entities that are not *specific* (not a proper noun, for example: 'The Player').

— By looking at the text t and the extracted pair, we remove all pairs for which the correct meaning wasn't captured. For example, with t = 'Twitter suspends New York Times magazine writer', the captured pair will be (*Twitter, New York Times*) which is semantically incorrect and should be removed.

After passing the filtering steps, the pairs are added to the set of discovered pairs $\mathcal{S}_i$.

3.2 Pattern inference

Pattern extraction: For all entity-platform pairs in the set $\mathcal{S}$, we extract from our corpus all posts that contain both the entity and the platform (case-insensitive). For each title t in the selected posts, we extract a new pattern by replacing the entity by the placeholder <ent> and the platform by the placeholder <plat>.

Pattern filtering: We start by grouping the patterns after changing them to lower case. We remove the patterns that have less than α occurrences and compute the *precision* of the remaining patterns. The precision is computed by starting the next iteration of **Pair extraction** with the candidate patterns, then computing the pattern precision π as:

$$\pi(p) := \frac{hits(p, \mathcal{S}_i)}{retrieved(p)} \tag{1}$$

with $hits(p, \mathcal{S}_i)$ being the number of pairs in the set $\mathcal{S}_i$ that can be extracted using pattern p, and $retrieved(p)$ the total number of pairs extracted with pattern p. Using this metric, we manually filter the pairs that have low precision. Finally, we perform a last manual filtering step to remove all patterns that don't have the correct semantic meaning, in terms of an entity being banned from a platform.

3.3 Linking with Google Knowledge Graph identifiers

After n iterations of the two steps described above, we have a set $\mathcal{S}$ of entities and platforms. We wish to link the entities to their Google Knowledge Graph identifiers. This step serves two purposes. First, it allows us to filter the entities that are not 'public' or well known enough to appear in the Knowledge Graph. Second, it allows us to retrieve the Google trends and Wikipedia page views (both of which can be done with the Google IDs). This step is followed by a manual check ensuring that the ID retrieved corresponds to the correct entity.

3.4 Final aggregation and filtering

Finally, we aggregate the pairs using their Google IDs and manually filter the resulting dataset again. We also add the ban date (with month precision) taking the months at which the posts talking about the bans were posted.

The manual filtering is done according to these requirements:

— The banning information is correct: this is checked by looking at the source of the original Reddit posts from which the pair is extracted and by cross-checking with other sources.

— The date of the ban is correct: as for the previous point, we check with different sources and correct the date if it isn't correct.

The filtering is done severely as to ensure that every entry of the final dataset is correct: the entity was banned from the platform at ban_dates.

The different steps and results, as well as implementation details, are detailed in section 4 and shown on the diagram below, figure 1.

4 Implementation

All the source code of our implementation can be found on this GitHub repository. We perform only two iterations of the algorithm described in section 3, as the size of the results grows exponentially, and we get a satisfactory number of pairs at the end of the two iterations. At each iteration, we add to the set $\mathcal{S}_i$: the entity-platform pair, as well as information about the posts from which the pair was extracted (text, date, ID, score and nb_comments).

4.1 First iteration

Pattern matching: We use a simple seed pattern for $\mathcal{P}_0$ in the first iteration: `[<ent> banned from <plat>]`. We also add the following pattern variations to increase the number of base pairs (as we are only doing two iterations): `[<ent>( am | was | is | are | were | got | gets | will be | getting | being | has been | have been | been )banned from <plat>]`. We get approximately 9'500'000 posts that match at least one of these patterns.

Platform and entity extraction, filtering: Performing these two steps, we reduce the number of posts to approximately 2000 and the number of pairs in $\mathcal{S}_0$ to 440. We use regex (regular expressions) to extract the entity and platforms, and the entity filtering is executed with the NLP library Spacy [2] for noun recognition.

Pattern extraction: We first select all the posts in the corpus containing one of the base pairs in the set $\mathcal{S}_0$ and extract the candidate patterns.

Pattern filtering: After transforming all the patterns to lower case and grouping them, we get 80 000 patterns. We choose $\alpha = 14$ for the occurrence cut-off, which reduces the number of patterns to 184. We then extract all the posts matching any of the 184 patterns to compute the precision. We filter manually and using the precision to arrive at a set $\mathcal{P}_1$ of 17 patterns. These patterns range from very general (e.g: `[<plat> banned <ent>]`) to the more specific (e.g:`[<plat> employee 'deactivated' <ent>]`).

4.2 Second iteration

Pattern matching: We extract all the posts containing one of the 17 patterns in $\mathcal{P}_1$, which make a total of approximately 40 million posts.

Pair extraction and filtering: We get approximately 3000 pairs in $\mathcal{S}_1$ after performing this step.

4.3 Google Knowledge Graph identifiers and final filtering

After linking the entities with their Google IDs (using the Google Knowledge Graph search API), aggregating them and manually filtering them, we have a final dataset of 293 entries.

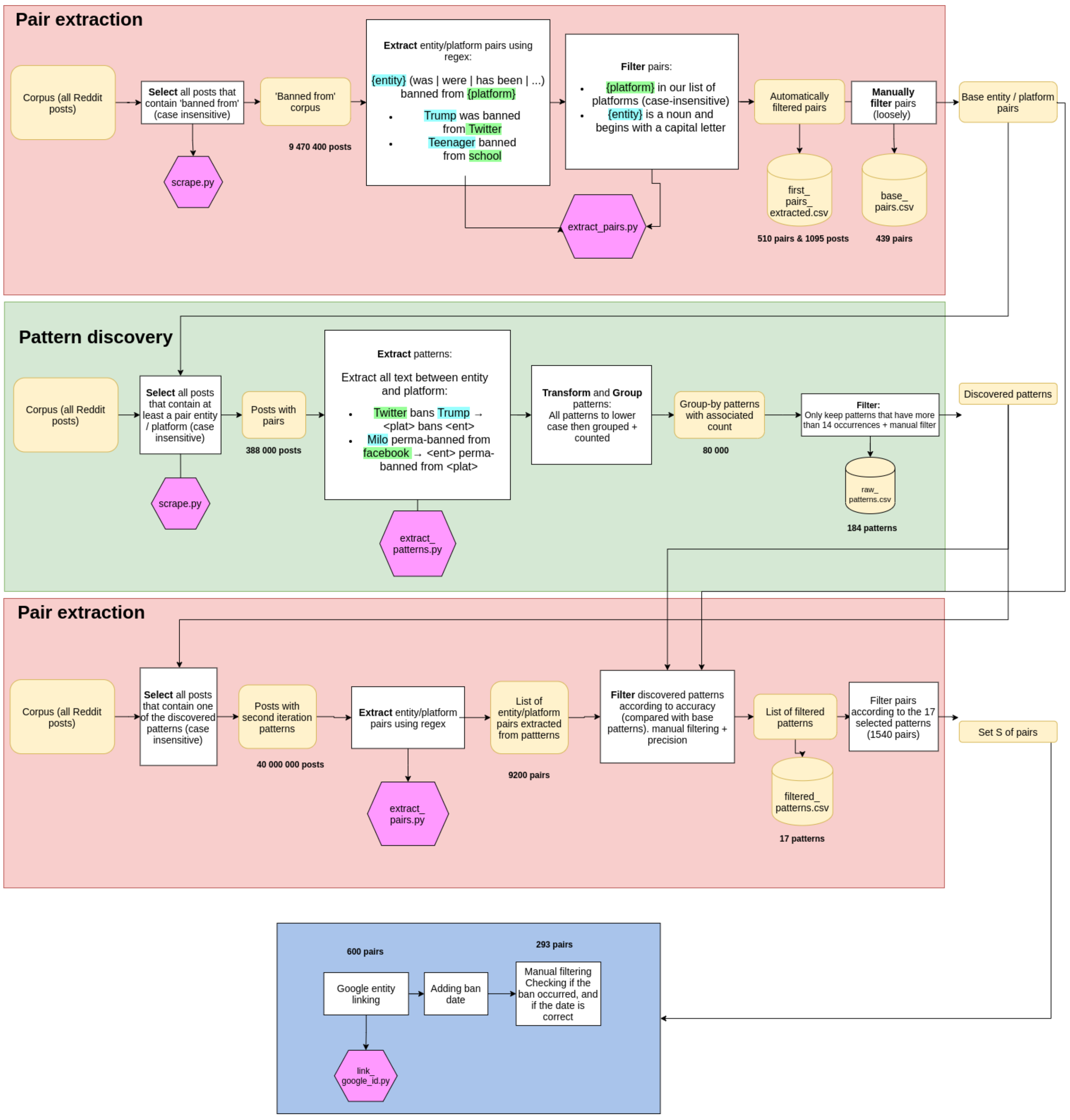
Pair extraction
Corpus (all Reddit posts)
Select all posts that contain 'banned from' (case insensitive)
scrape.py
'Banned from' corpus
9 470 400 posts
Extract entity/platform pairs using regex:
{entity} (was | were | has been | ...) banned from {platform}
Trump was banned from Twitter
Teenager banned from school
Filter pairs:
{platform} in our list of platforms (case-insensitive)
{entity} is a noun and begins with a capital letter
extract_pairs.py
Automatically filtered pairs
first_pairs_extracted.csv
510 pairs & 1095 posts
Manually filter pairs (loosely)
base_pairs.csv
439 pairs
Base entity / platform pairs
Pattern discovery
Corpus (all Reddit posts)
Select all posts that contain at least a pair entity / platform (case insensitive)
scrape.py
Posts with pairs
388 000 posts
Extract patterns:
Extract all text between entity and platform:
Twitter bans Trump → <plat> bans <ent>
Milo perma-banned from facebook → <ent> perma-banned from <plat>
extract_patterns.py
Transform and Group patterns:
All patterns to lower case then grouped + counted
Group-by patterns with associated count
80 000
Filter:
Only keep patterns that have more than 14 occurrences + manual filter
raw_patterns.csv
184 patterns
Discovered patterns
Pair extraction
Corpus (all Reddit posts)
Select all posts that contain one of the discovered patterns (case insensitive)
Posts with second iteration patterns
40 000 000 posts
Extract entity/platform pairs using regex
extract_pairs.py
List of entity/platform pairs extracted from patterns
9200 pairs
Filter discovered patterns according to accuracy (compared with base patterns). manual filtering + precision
List of filtered patterns
filtered_patterns.csv
17 patterns
Filter pairs according to the 17 selected patterns (1540 pairs)
Set S of pairs
600 pairs
Google entity linking
link_google_id.py
Adding ban date
293 pairs
Manual filtering Checking if the ban occurred, and if the date is correct

Figure 1 – Diagram describing the steps and results of the algorithm

5 Results

5.1 Description of the dataset

The final dataset contains the following information:

— *entity*: name of the banned entity
— *platform*: platform from which the entity is banned
— *g_ id*: google identifier corresponding to the entity
— *IDs*: list of post IDs from which the pair was extracted
— *score*: list of scores of the posts from which the pair was extracted
— *nb_ comments* : list of number of comments on the Reddit posts from which the information was extracted
— *ban_ date*: list with the date(s) of the ban(s) of *entity* from *platform*

We have 293 rows, with 255 unique entities. The distribution of the platforms, entity types (according to google knowledge graph classification) and ban dates can be found in the appendix (Figure 4).

5.2 Preliminary analysis

We wish to analyse the effect of deplatforming on the attention an entity receives online. As a proxy for online attention, we use Google Trends. These trends are extracted using the Google Knowledge Graph identifier retrieved in earlier steps, as well as the Google Trends Anchor Bank library [3] which converts the trends to a universal, absolute scale, giving an estimate of the search volume for a certain query. The Google Trends we extract have a monthly precision.

To analyse if banning an entity has an impact on its google search volume, we construct an interrupted time series analysis, to see if the trends change after the ban, both on the whole dataset and on specific entities.

5.2.1 First approach

The first approach consists of creating an interrupted time series for *all* the entries of the dataset at the *same time*. We try this approach for different window sizes (the number of months that we study before and after the ban), going from 3 to 25.

For each window size, we run Algorithm 1 for all the Google trends of each entity and their respective ban dates. We then build a model on the resulting dataset. This linear model can be described as:

$$ratio = \text{intercept} + \alpha * \text{month} + \beta * \text{banned} + \gamma * \text{month} : \text{banned} \tag{2}$$

We can interpret coefficient β as the immediate decrease/increase of the time series at the moment of the ban, and γ at the difference of angle between the trends before and after the ban.

Algorithm 1: Pre-processing the Google trend time series

```
Result: Centered and normalized time series
For a window size $w$, a time series $t$ and a ban date $b$ ;
for i in t do
    if i.month $\in$ [b-w, b+w] then
        i.month $\leftarrow$ i.month - b ;
        if i.month $\leq 0$ then
            i.banned $\leftarrow$ 0
        else
            i.banned $\leftarrow$ 1
        Add i to result ;
for i in result do
    i.ratio $\leftarrow$ (i.ratio - result.mean)/result.std
```

With this approach, we get a model for which the coefficients are all significant. The parameters beta and gamma evolve as follows when we change the window size of the model (Figure 2). We can see that as we increase the window size, both parameters converge to 0 (no effect), suggesting that the overall effect dissipates over time.

This analysis looks at all the time series at the same time, making individual behaviours disappear. To check whether the 'deplatforming effect' is the same for all entities, we zoom in and build a model for each individual entity.

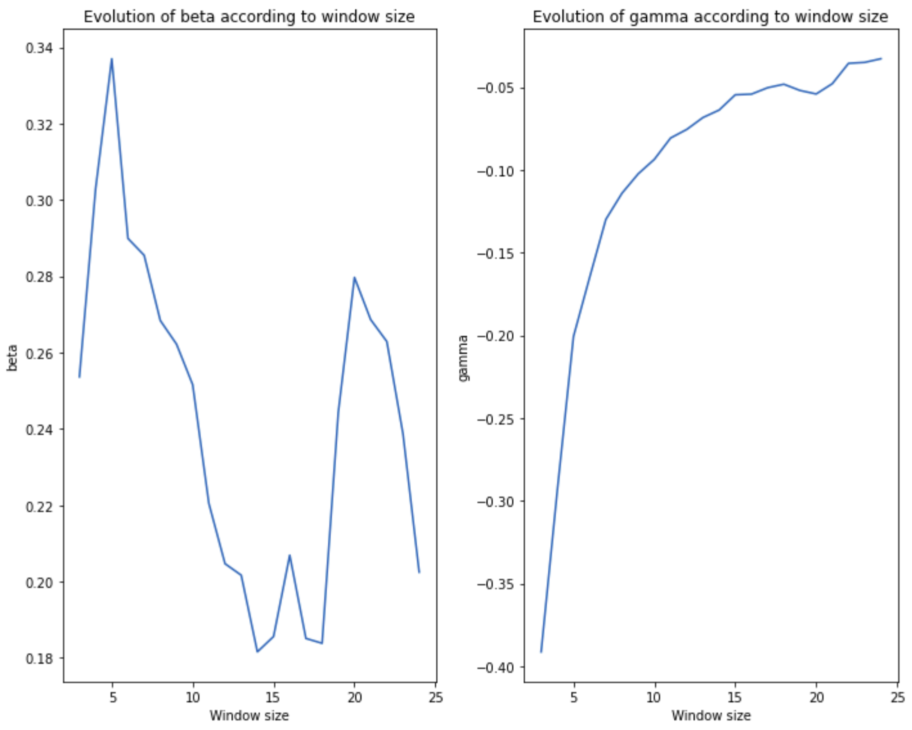

Figure 2 – Evolution of model's parameter according to the window size

5.2.2 Second approach

This is where our analysis becomes short-handed, because only approximately 10% of entities produce models with significant parameters when we run them individually. For some of them (which are significant), we can see a clear 'deplatforming effect' which is different from entity to entity. For example, we can see on Figure 3 (left) that the ban caused a clear drop in the attention, which then continuously decreased afterward. However, on the right we can see a totally different effect, with a gain of popularity after the ban that then slowly decreases but stays above the level before the ban.

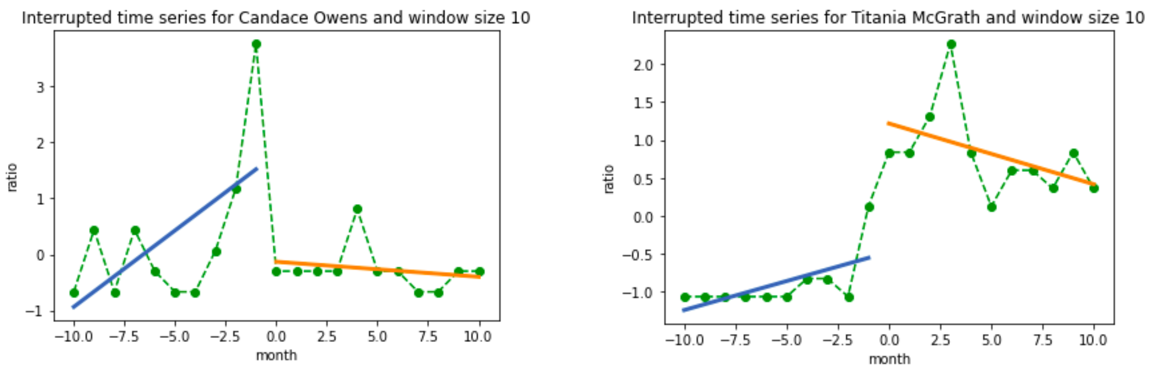

Figure 3 – Examples of 'deplatforming effect'

6 Challenges and further work

This project was challenging on many aspects. First, the volume of Reddit data – and the fact that our results were manually filtered multiple times – constrained us to reduce the number of iterations of the algorithm and, at each step, to be very severe on the filtering to reduce the quantity to manually filter in the next steps. The diversity of spellings in the posts also made the filtering (manual as well as automatic) quite tedious.

6.1 Improving the dataset

The dataset can still be improved by making the ban dates more precise and by adding more attributes to each entry. The ban dates are in monthly precision, and currently the ban date corresponds to the month of the ban, and will be the same if the banned occurred on the first day or on the last day of the month. This might need to be corrected to allow for a more precise ITS model. Additionally, it could be useful to add new attributes to the dataset as they might have an influence on the deplatforming effect. Such attributes can be: duration of the ban (some bans are temporary while others are permanent), cause of the ban (some are mistakes, some are mandated by governmental institutions and others are because of a disrespect of the platform's rules).

6.2 Improving the effect analysis

As presented in section 5.2, our analysis is not currently conclusive, and we fail at creating good models that describe the attention and the effect (or not) of the ban. To achieve better results and be able to draw conclusions on the effects of deplatforming, it would require creating a methodological toolbox that fits our data and what we want to show. Additionally, the attention proxy can be improved by adding Wikipedia or news coverage data to the Google trends analysis.

7 Conclusion

Our adaptation of the 'Quootstrap' algorithm applied to a corpus of 10 years of Reddit posts allowed us to create a dataset with 293 instances of some entity being banned from some platform. Our preliminary analysis shows that we can sometimes observe a statistically significant effect of the ban on the attention of an entity, but the analysis needs to be complemented with a more suitable model and extended dataset to draw more significant conclusions.

We hope that the creation of such dataset (covering a decade of deplatforming on many platforms) is the preliminary step in such analysis, and will allow us to better understand the effects of deplatforming.

A Appendix

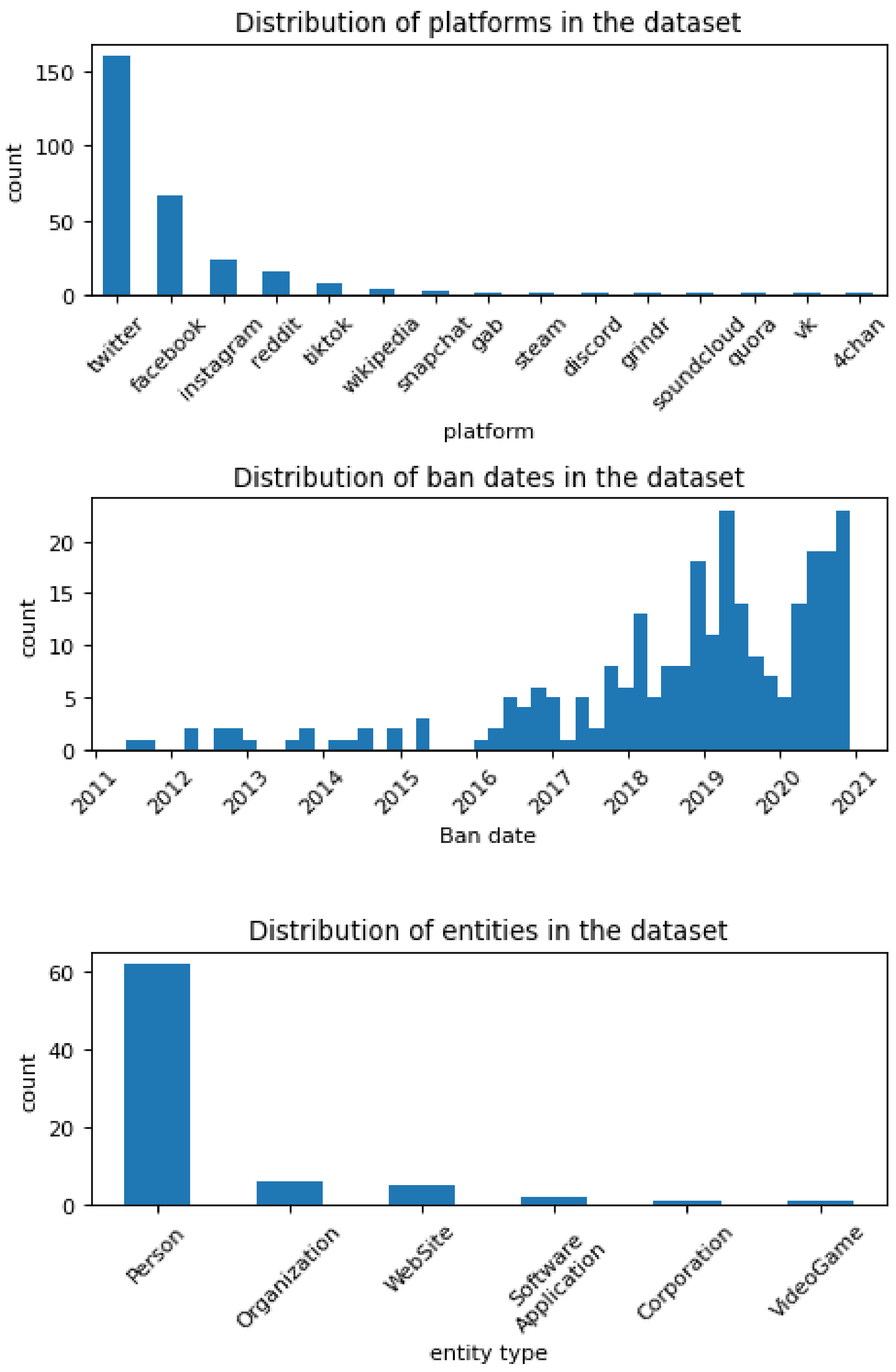

Figure 4 – Distribution of the final dataset

References

[1] Dario Pavllo, Tiziano Piccardi, and Robert West. "Quootstrap: Scalable Unsupervised Extraction of Quotation-Speaker Pairs from Large News Corpora via Bootstrapping". In: CoRR abs/1804.02525 (2018). arXiv: 1804.02525. URL: http://arxiv.org/abs/1804.02525.

[2] Matthew Honnibal et al. spaCy: Industrial-strength Natural Language Processing in Python. 2020. DOI: 10.5281/zenodo.1212303. URL: https://doi.org/10.5281/zenodo.1212303.

[3] Robert West. "Calibration of Google Trends Time Series". In: CoRR abs/2007.13861 (2020). arXiv: 2007.13861. URL: https://arxiv.org/abs/2007.13861.