HW 6 - Analyzing Disinformation Domains

Swathi Venkatesh
11/21/2021

Q1

Datasets D1 and D2 include the number of tweets that each domain was shared in (found in the last column/field of the dataset).

Q: For each of D1 and D2, what were the top 10 domains in terms of tweets?

For each of the top 10 domains from the previous Q:

Q: Which ones are no longer live? *Q*: How would you classify the domain? Show this information in a table like the one below, sorted by number of tweets. You should have 2 tables, one for the top 10 in D1 and one for the top 10 in D2.

Answer

```python
#!/usr/local/bin/python3
import pandas as pd
import numpy as np
#read the files in pandas dataframe
file1= pd.read_csv("D1.csv")
file2 = pd.read_csv("D2.csv")

#sort them in order
file1 = file1.sort_values(by=’# Citations in our Alternative Narrative Tweets’,ascending=False)
file2 = file2.sort_values(by=’Tweet count’, ascending=False)

search = file1.copy()

#get only 10 items
file1 = file1.head(10)
file2 = file2.head(10)


#drop unwanted columns
file1.drop([’Primary Orientation (Determined through Content Analysis)’, ’How Cited in Alternative Narrative of Shooting Events’],axis= 1, inplace=True)
file2.drop([’URL count’],axis =1, inplace=True)
```

```python
#rename columns
file1.rename(columns={"# Citations in our Alternative Narrative Tweets":"Tweets","Media Type (Determined through Content Analysis)":"Website Type"},inplace=True)
file2.rename(columns={"Tweet count":"Tweet"},inplace=True)

#swap order for first d1
columns_swap = ["Domain","Tweets","Website Type"]
file1 = file1.reindex(columns=columns_swap)

#add new columns to the data with NAN values
file1['status']= np.nan
file2['Website Type']= np.nan
file2['status']= np.nan

#change column types to string
file2['Website Type'] = file2['Website Type'].astype(str)
file2['status'] = file2['status'].astype(str)

numCount = 0
temp =""
 #Match domains in top 10 D2 dataframe with D1 to obtain Website Media Type
for index, row in file2.iterrows():
    #find a match(es) and store as a dataframe
    temp = search[search['Domain'].str.contains(row['Domain'])]
    #check if data frame is empty
    if(len(temp) == 0):
        #assign NaN value
        final = np.nan
    else:
        #assigne Media Type to final value
        final = temp['Media Type (Determined through Content Analysis)'].iloc[0]
    #insert into file2 dataframe
    file2.at[index,"Website Type"] = final

file1.to_csv("D1_new.csv", index = False, header=True)
file2.to_csv("D2_new.csv", index = False, header=True)
```

Listing 1: one.py

Table 1: Top 10 High Number of Tweets Domains (d1_new.csv)

Domain	Tweets	Media	Status
therealstrategy.com	7113	Alternative Media	not live
infowars.com	1741	Alternative Media	not live
newsbusters.org	1217	Alternative Media	live
washingtonpost.com	1108	MSM	live
nodisinfo.com	774	Alternative Media	not live
nytimes.com	759	MSM	live
veteranstoday.com	586	Alternative Media	live
beforeitsnews.com	580	Alternative Media	live
rawstory.com	308	Alternative Media	live
hoax.trendolizer.com	299	fact checker	live

Table 2: Top 10 High Number of Tweets Domains(d2_new.csv)

Domain	Tweets	Media	Status
21stcenturywire.com	3088	Alternative Media	live
clarityofsignal.com	2352	Not found(Alternative Media)	live
rt.com	1598	Foreign Government Media	live
newsweek.com	1249	Not found(MSM)	live
alternet.org	1221	Not found(Alternative Media)	Live
sputniknews.com	1076	Foreign Government Media	live
mintpressnews.com	919	Not found(Alternative Media)	live
cnn.com	756	MSM	live
globalresearch.ca	724	Alternative Media	live
theantimedia.org	682	Alternative Media	live

Discussion

For D1_new.csv I read the D1.csv file in a pandas dataFrame and was able to easily to sort the data according to the number of Tweets in Highest to lowest. As for D2_new.csv I did the same thing but I read in D2.csv and also compared the data with D1.csv to obtain the Media Values. Those values that are not found shows not found and what i manually classified them. I also manual checked the website on the browser to check if the where active website.

The Table 1 shows Top 10 High Number of Tweets Domains for D1 and Table 2 shows Top 10 High Number of Tweets Domains for D2. The Domains therealstrategy.com, infowars.com, nodisinfo.com are not live and all the other domains are live in d1_new.csv. In d2_new.csv all the top 10 domains are live. In Table 1 the highest number of tweets is 7113 and the top domain is therealstrategy.com. In Table 2 the highest number of tweets is 3088 and the top domain is 21stcenturywire.com. Most of the websites that have the Top 10 tweets belong to either Alternative

Media or Main Stream Media. But in Table 2 two of the domains belong to Foreign Government Media.

Q: For each of D1 and D2, what were the top 10 domains in terms of tweets?

Ans. The top 10 domains for D1 and D2 are as shown in Table 1 and Table 2.

Q: Which ones are no longer live?

Ans. The Domains therealstrategy.com, infowars.com, nodisinfo.com are not live and all the other domains are live in d1_new.csv. In d2_new.csv all the top 10 domains are live.

Q: How would you classify the domain? Ans. The domain classification is as shown in Table 1 and Table 2.

Q: What can you say about the domains that have been frequently shared on Twitter?

Comparing both the tables for D1 and D2 files, the top domains either belong to Alternate Media or MSM. For D1 top 10 domains, the ones that are not live belong to Alternate Media. The highest number of tweets in both the tables belong to website type Alternative Media. And the lowest number of tweets in Table 1 belong to Media type fast checker and to Alternate Media in Table 2.

Q2

Compare the amount of overlap between the three datasets. Remember that:

- D1 - domains shared in tweets related to mass shootings
- D2 - domains shared in tweets related to the White Helmets in Syria
- D3 - domains found to be publishing false Coronavirus information

- a. domains that are present in both D1 and D2
- b. domains that are present in both D2 and D3
- c. domains that are present in both D1 and D3
- d. domains that are present in all three datasets

Create a table showing the number of domains in each of the datasets above. List out the domains in each of the datasets in your report. Upload each of the datasets to your GitHub repo.

Answer

```python
#!/usr/local/bin/python3
#import necessary libraries
import pandas as pd

#read the files in pandas dataframe
file1= pd.read_csv("D1.csv")
file2 = pd.read_csv("D2.csv")
file3 = pd.read_csv("D3.csv")

def compareThisB(lowerCase,upperCase):
    #create an empty final dataframe
    number = 0
    column_name = ["Domain"]
    final = pd.DataFrame(columns = column_name)
    for index, row in upperCase.iterrows():
        #find a match(es) and store as a dataframe
        #set Uppercases domain to lowercase so that it can propermatch
        temp = lowerCase[lowerCase[’Domain’] == row[’Domain’].lower()]
        #check if data frame is empty
        if(len(temp) == 0):
            pass
        else:
            final.at[number,"Domain"] = temp[’Domain’].iloc[0]
            number +=1
    return final

"""
   Q2 Part a compare D1 and D2
"""
a_final= compareThisB(file1,file2)

"""
   Q2 Part b compare D2 and D3
"""
b_final = compareThisB(file2,file3)
"""
   Q2 Part c compare D1 and D3
"""
c_final = compareThisB(file1,file3)

"""
   Q2 Part d compare a_final  and D3
"""
d_final = compareThisB(a_final,file3)

#convert to csv files
```

```python
a_final.to_csv("a_final.csv", index = False, header=True)
b_final.to_csv("b_final.csv", index = False, header=True)
c_final.to_csv("c_final.csv", index = False, header=True)
d_final.to_csv("d_final.csv", index = False, header=True)
```

Listing 2: two.py

Discussion

The function called compareThisB(lowerCase,upperCase) to handle all the process for comparisons, for a,b,c,and d. This functions takes the two parameters. The second parameters get converted in all lower case before comparisons. Table 3 shows the domains that are present in both D1 and D2. All the data sets that are common either belong to Alternative Media or MSM. Table 4 shows the domains that are present in both D2 and D3. All the data sets that are common in D2 and D3 belong to some or other Media type. Table 5 shows the domains that are present in both data sets D1 and D3. All the data sets that are common in D1 and D3 belong to either Alternative Media or Foreign Government Media. Table 6 shows the domains that are present in both data sets D1, D2 and D3. s. All the data sets that are common in D1, D2 and D3 belong to either Alternative Media or Foreign Government Media.

Q: Is there anything interesting about the domains that are present in multiple datasets?

D1 talks about related to mass shootings, D2 is related to work of White Helmets in Syria and D3 about the domains publishing false Coronavirus information. Since we are considering the data sets that are not related to each other so its interesting to see that they still have the common domains. For mass shootings these domains have been listed to be having the correct information but D3 is all about the domains that have spread the false news about the corona virus. Its interesting to see the domains that have reported the correct information to be reporting all the false information about the corona virus.

Table 3: Domains that are present in both D1 and D2 from a_final.csv

Domain
rt.com
breitbart.com
theeventchronicle.com
themillenniumreport.com
beforeitsnews.com
thefreethoughtproject.com
veteranstoday.com
theintercept.com
theguardian.com
21stcenturywire.com
infowars.com
thedailybeast.com
heavy.com
blacklistednews.com
presstv.com
dcclothsline.com
theantimedia.org
upi.com
investmentwatchblog.com
dailymail.co.uk
nydailynews.com
fellowshipoftheminds.com
thetruthseeker.co.uk
abovetopsecret.com
cnn.com
worldtruth.tv
sputniknews.com
lewrockwell.com
nytimes.com
intellihub.com
thedailysheeple.com
globalresearch.ca
foxnews.com
thestar.com
activistpost.com
nbcnews.com

Table 4: Domains that are present in both D2 and D3 from b_final.csv

Domain
activistpost.com
beforeitsnews.com
breitbart.com
collective-evolution.com
dcclothesline.com
gellerreport.com
humansarefree.com
infowars.com
intellihub.com
ronpaulinstitute.com
sott.net
thewashingtonstandard.com
worldtruth.com
21stcenturywire.com
davidicke.com
off-guardian.com
presstv.com
ukcolumn.org
rubikon.news
globalresearch.ca
theduran.com

Table 5: Domains that are present in both D1 and D3 from c_final.csv

Domain
activistpost.com
beforeitsnews.com
breitbart.com
dcclothesline.com
infowars.com
intellihub.com
wakingtimes.com
worldtruth.com
zerohedge.com
21stcenturywire.com
presstv.com
globalresearch.ca

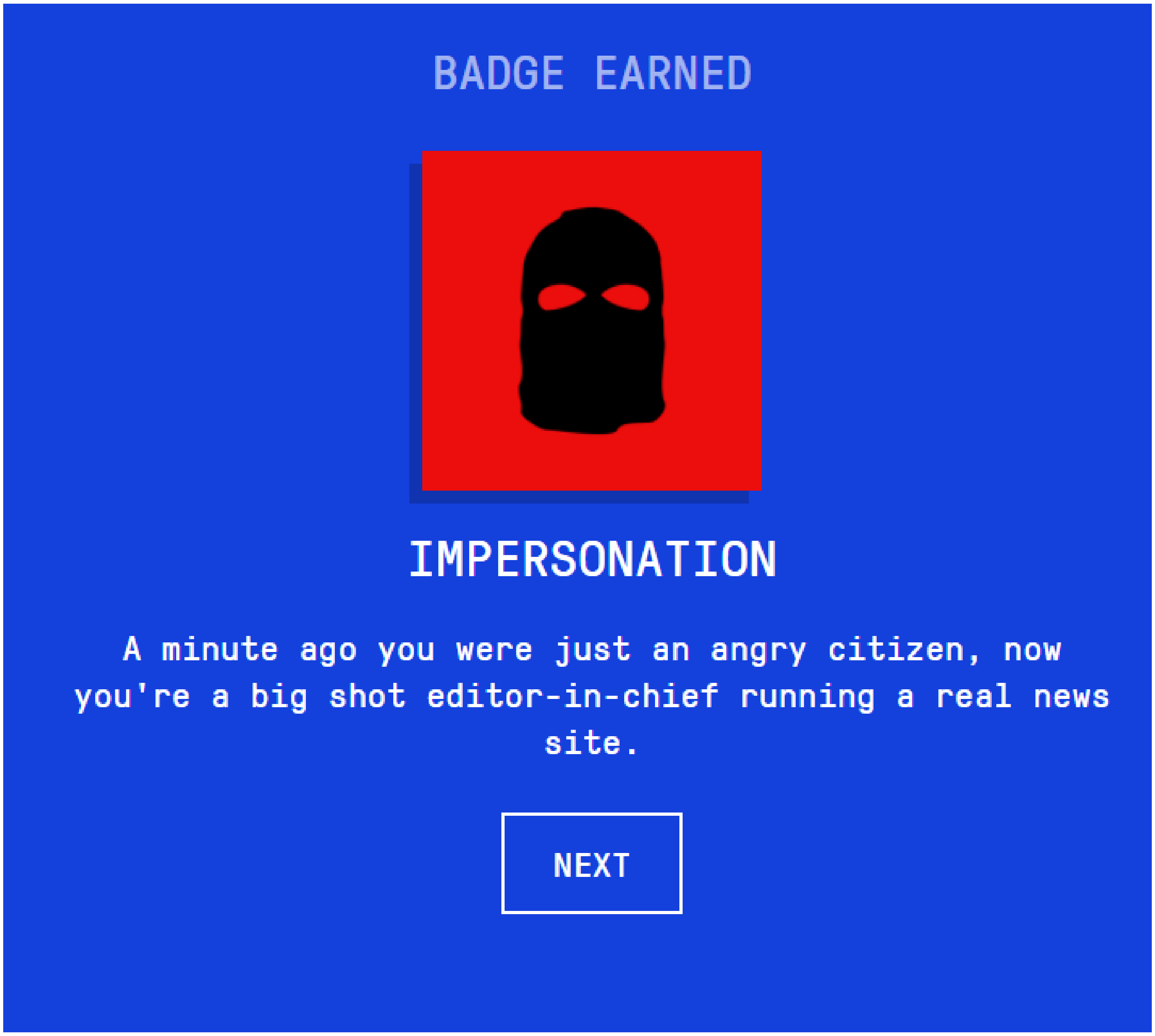
Table 6: Domains that are present in all three datasets from d_final.csv

Domain
activistpost.com
beforeitsnews.com
breitbart.com
dcclothesline.com
infowars.com
intellihub.com
worldtruth.com
21stcenturywire.com
presstv.com
globalresearch.ca

Q4

There have been several online games created to educate people about disinformation and how it spreads on social media. Play one of the games at https://fakey.osome.iu.edu/, https://www.getbadnews.com, or https://goviralgame.com. Write a paragraph about your experience and some lessons you learned by playing the game. Take some screenshots as you play to include in your report.

Answer

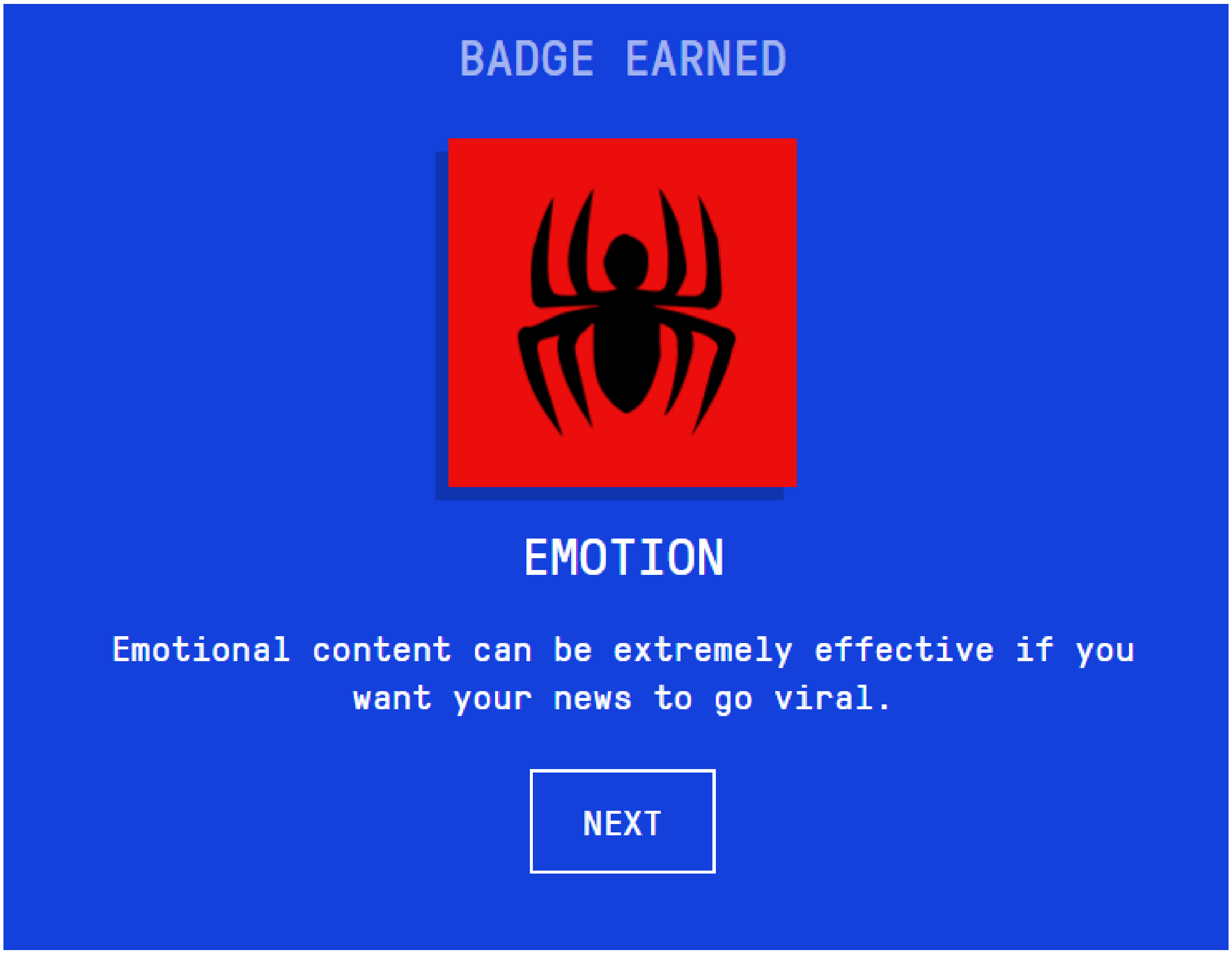


BADGE EARNED

IMPERSONATION

A minute ago you were just an angry citizen, now you're a big shot editor-in-chief running a real news site.

Figure 1: Badge1

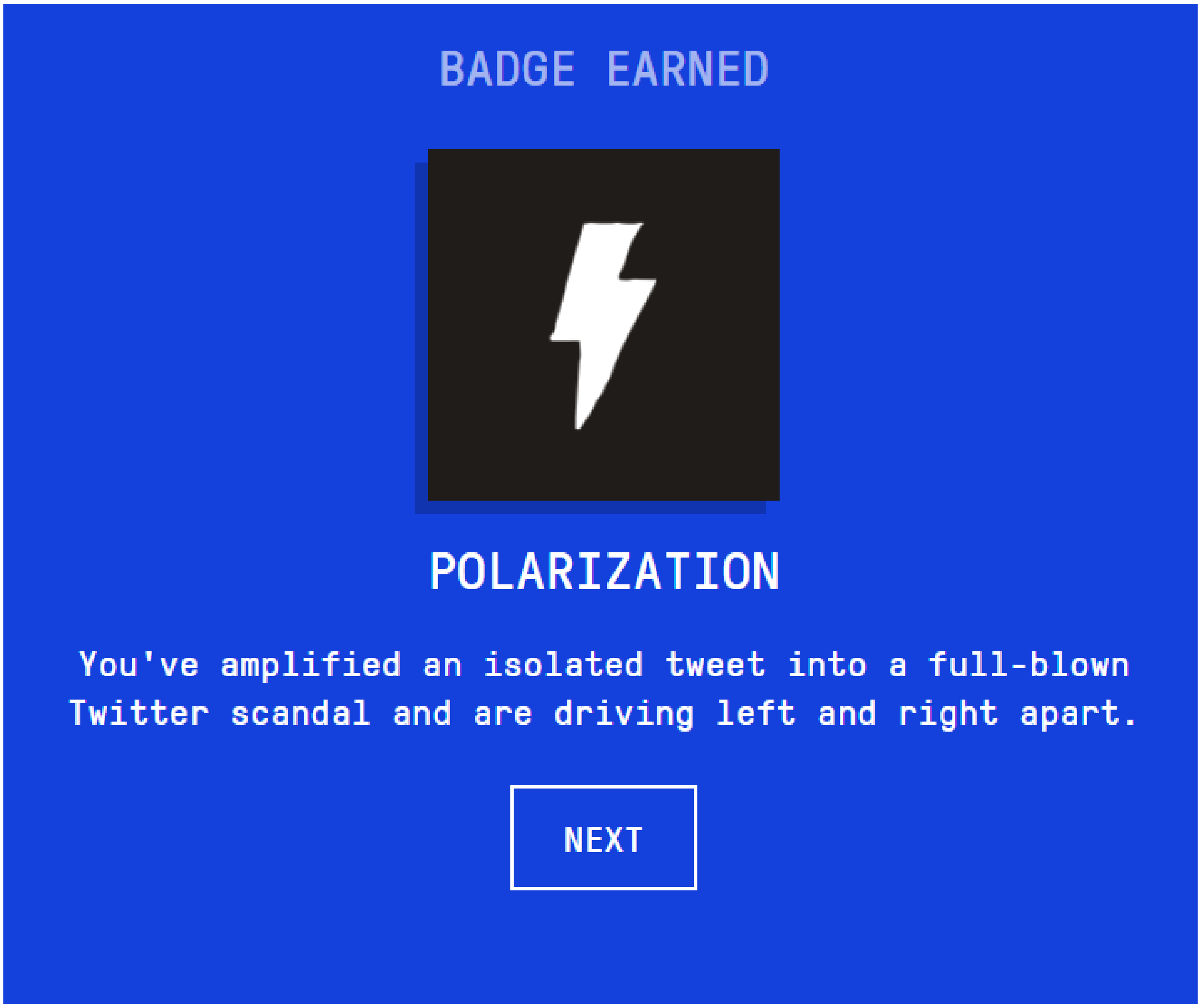


BADGE EARNED

EMOTION

Emotional content can be extremely effective if you want your news to go viral.

Figure 2: Badge2

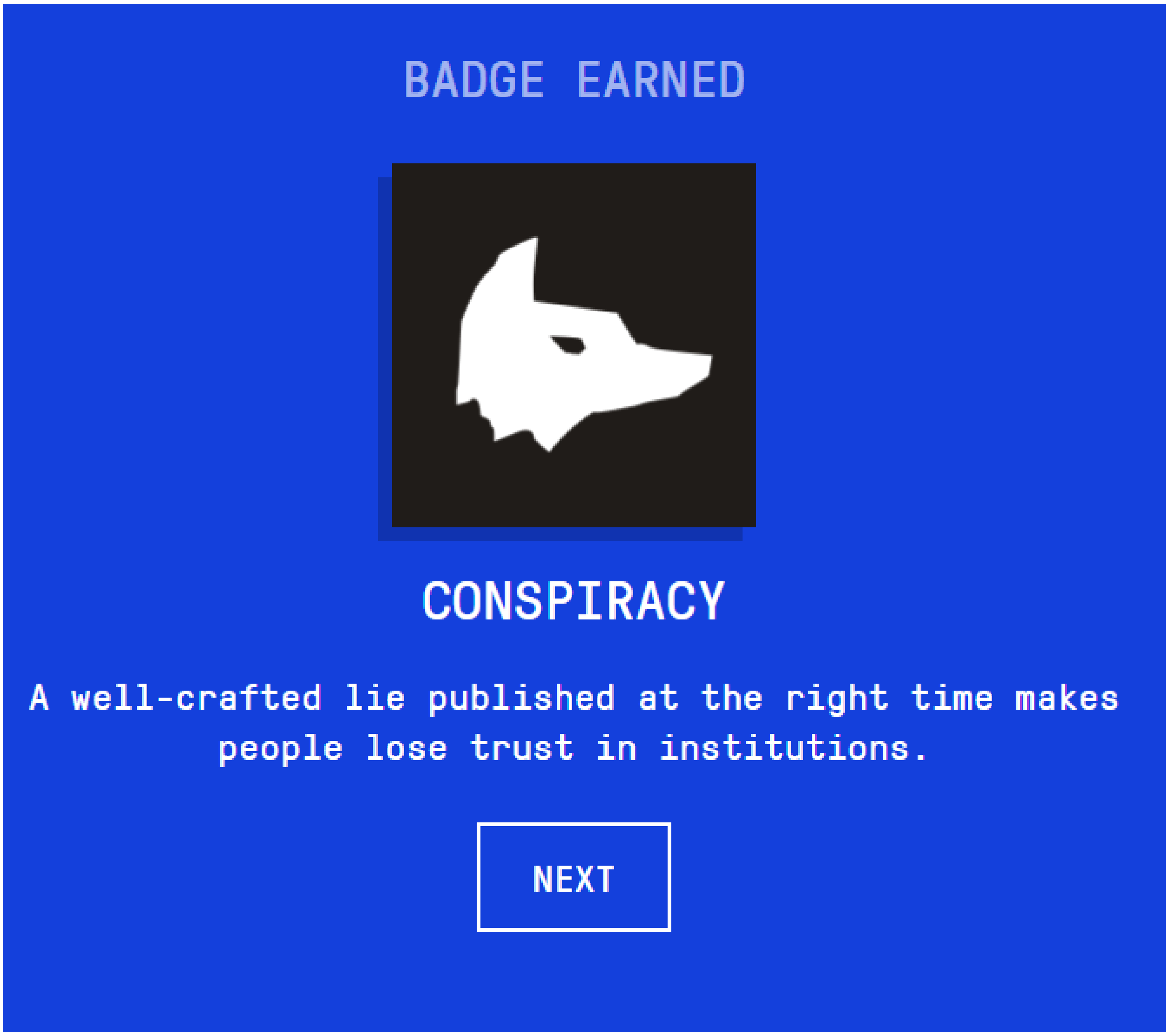


BADGE EARNED

POLARIZATION

You've amplified an isolated tweet into a full-blown Twitter scandal and are driving left and right apart.

Figure 3: Badge3

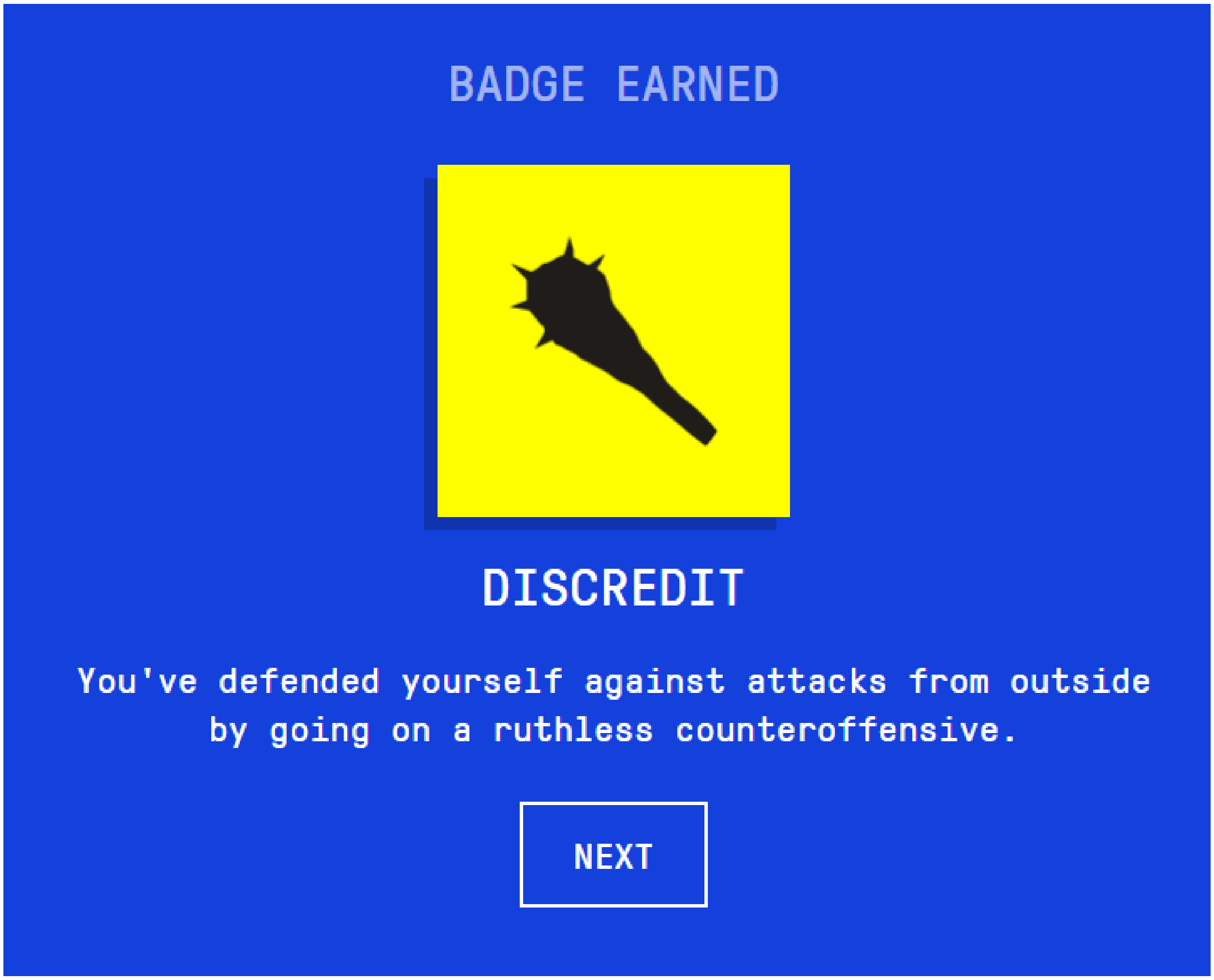


BADGE EARNED

CONSPIRACY

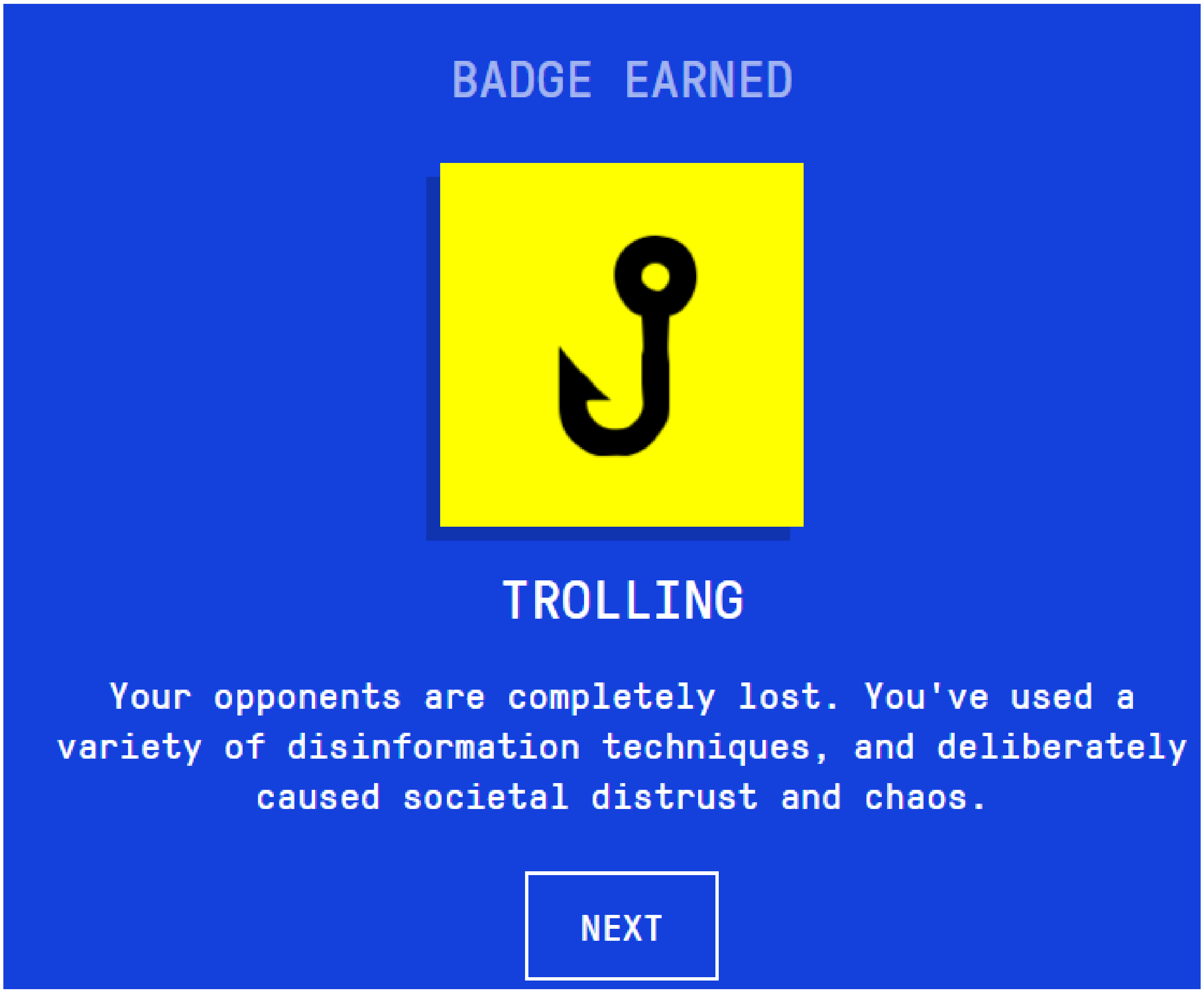
A well-crafted lie published at the right time makes people lose trust in institutions.

Figure 4: Badge4



You've defended yourself against attacks from outside by going on a ruthless counteroffensive.

Figure 5: Badge5

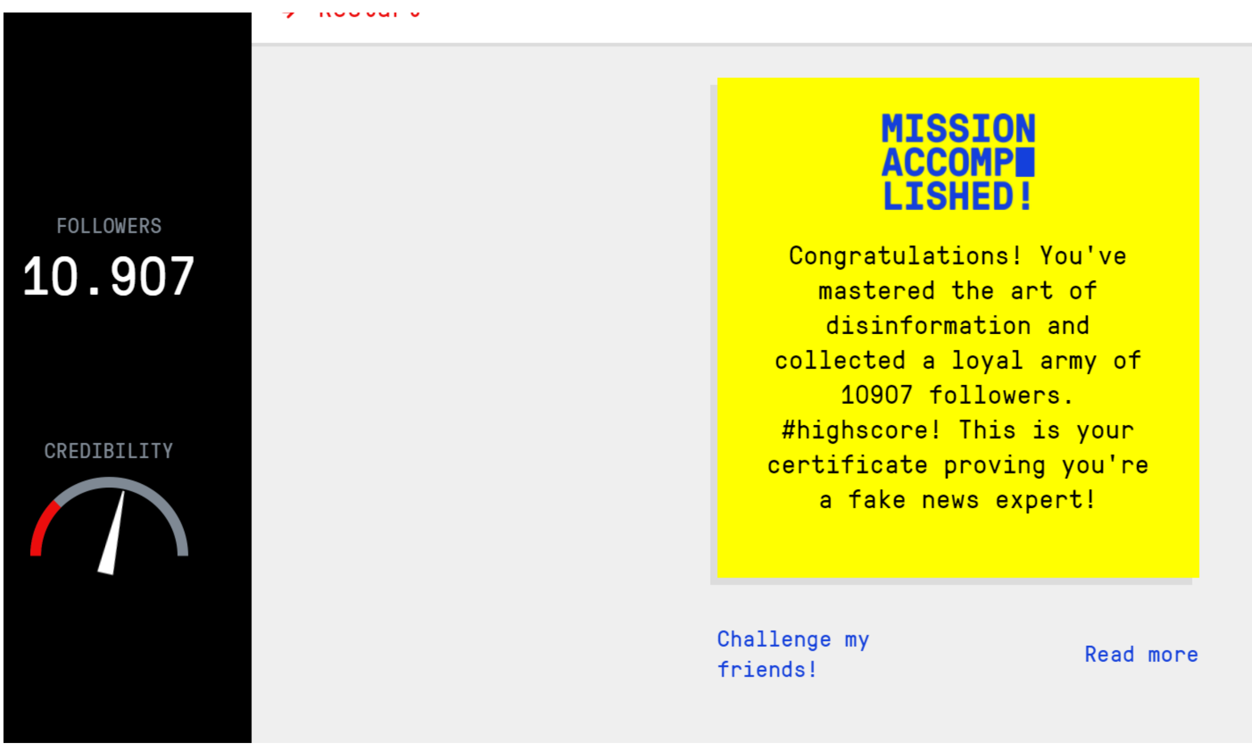


BADGE EARNED

TROLLING

Your opponents are completely lost. You've used a variety of disinformation techniques, and deliberately caused societal distrust and chaos.

Figure 6: Badge6



MISSION ACCOMPLISHED!

Congratulations! You've mastered the art of disinformation and collected a loyal army of 10907 followers. #highscore! This is your certificate proving you're a fake news expert!

Figure 7: Final Badge

Discussion

I used https://www.getbadnews.com/ to play the came and was able to earn the above badges below:

- Impersonation in Figure 1 (Impersonating someone else and disguising myself as a credible news source which was highly effective in increase my followers)
- Emotion in Figure 2 (Playing to people's emotion out of fear, anger or compassing was a great tool for spreading my messages)
- Polarization in Figure 3 (By finding existing grievance and blowing them out of proportion, drove people apart and made think a story is much more important that it really was.)
- Conspiracy in Figure 4(I can use people's desires for the 'truth' as a tool to lure them into my band of followers)
- Discredit in Figure 5(When someone is attacking my credibility i strike back. I do not apologize nor do I play nice and above all I do not retreat)
- Trolling in Figure 6 (Is a tool that evokes an emotional response such as anger, irritation or sadness.)
- The final score was 10907 followers in Figure 7.

References

- https://www.datacamp.com/community/tutorials/pandas-read-csv
- https://stackoverflow.com/questions/45164537/filter-pandas-data-frame-based-on-exact-string-match
- https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sort_values.html
- https://stackoverflow.com/questions/12021754/how-to-slice-a-pandas-data-frame-by-position
- https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.rename.html
- https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop.html
- https://stackoverflow.com/questions/38288372/unable-to-drop-a-column-from-pandas-dataframe
- https://stackoverflow.com/questions/45164537/filter-pandas-data-frame-based-on-exact-string-match
- https://stackoverflow.com/questions/39092067/pandas-dataframe-convert-column-type-to-string-or-categorical