/r/politics Network: An analysis of emerging behaviors during the 2020 US presidential election

Giuseppe Grieco
g.grieco6@studenti.unipi.it
Student ID: 606532

Mattia Sangermano
m.sangermano1@studenti.unipi.it
Student ID: 599826

ABSTRACT

In this paper is shown the analysis of a real-world political interaction network retrieved from the well known social network Reddit. The community taken into consideration is /r/politics and the month analysed is the one preceding the 2020 US presidential election. The analysis starts by showing some statistical information of the network and comparisons with synthetic graphs. Next, the implementation of an algorithm for counting connected graphlets of size 3 and 4 is shown. This is followed by a description of some of the results obtained through community discovery and opinion dynamics algorithms. The main results concern the analysis of the implicit interactions identified by the community discovery algorithms, the influence of the user's country on his behaviour and the impact of toxicity with respect to the interactions of a post. [1]

KEYWORDS

Social network analysis, US election, US presidental election

ACM Reference Format:
Giuseppe Grieco and Mattia Sangermano. 2021. /r/politics Network: An analysis of emerging behaviors during the 2020 US presidential election . In *Social Network Analysis '2021.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Nowadays, social networks like Reddit play a crucial role in the debate of political issues. In fact, many studies examine this social network to analyze different social phenomena e.g. [5], [10], [3].

Previous works on this topic focuses for example on the absence of echo chamber, in particular [7] analysed Trump and Clinton supporters and observes that there is a preference for cross-cutting political interactions between the two communities rather than within-group interactions, thus contradicting the echo chamber narrative. [9] analysis the impact of digital propaganda on users. An interesting work about social phenomena is [6], where the authors characterize the behavior of Trump supporters, looking at homophily, social influence, and social feedback. They use these information to predict the user participation in `r/The_Donald` and they found that homophily-based and social feedback-based features are the most predictive signals.

2 THE DATASET

The foundation of the presented work is the graph, since all the subsequent analysis relies on it. This paragraph describes the sources of information used and characterizes the graph.

2.1 Data collection

As mentioned in the previous paragraph, Reddit, more precisely the /r/politics subreddit, has been the source used to build the network on which the study is based. Other sources were then used to add information to the graph, namely:

- A BERT model used to classify comments and posts based on their political leaning [8].
- A set of geographical information for a subset of r/politics users for whom it was possible to infer such information. The approach used to retrieve this information is inspired by [2].
- A BERT model used to compute different toxicity measures on comments and posts. It has been trained on 3 Jigsaw challenges: Toxic comment classification, Unintended Bias in Toxic comments, Multilingual toxic comment classification.[4]

2.1.1 Crawling Methodology and Assumptions. The graph relies upon data collected partly by using the pushshift.io service which has been used to analyze the history of /r/politics in the entire month from 01-10-2020 to 31-10-2020 in purpose of obtaining all references to posts and comments. The resulting references were then supplemented with all the updated information retrieved using the Reddit API.

2.1.2 The crawled data-structure. A data-structure has been built for each complex object crawled - the non atomic ones, i.e. the post, the comment and the user. This section describes the data structures used and the fields they contain.

Post Data Structure. Each Post is characterized by 12 fields, 10 of which are atomic:

- `id`: a string that uniquely identifies the post.
- `title`: a string contains the content of the post.
- `ups`: an integer representing the number of upvote that the post received.
- `downs`: an integer representing the number of downvote that the post received.
- `upvote_ratio`: a float representing the ratio between `ups` and `downs`.

[1] **Repository**: https://github.com/sna-unipi/2021---final-project-bormioli_grieco_sangermano/

SNA '2021, 2020/2021, University of Pisa, Italy

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$0.00
https://doi.org/10.1145/1122445.1122456

- `is_original_content`: define if the content of the post is original or not.
- `score`: difference between ups and downs.
- `num_comments`: an integer representing the number of comments that the post received.
- `num_crossposts`: number of different communities in which the post has been shared.
- `created_at`: an integer representing the timestamp of the post creation time.

The remaining 2 are complex objects, in details:

- `author`: containing an instance of the *User Data Structure* related to the data crawled according to the identifier of the author.
- `comments`: is an hashmap which maps each comment identifier to the corresponding *Comment Data Structure*.

The hashmap that maps all the `id` of any *Post Data Structure* instance to its value is called Post Map.

Comment Data Structure. Each Comment is characterized by 12 fields, 11 of which are atomic:

- `id`: a string that uniquely identifies the comment.
- `downs`: an integer representing the number of downvote that the post received.
- `ups`: an integer representing the number of upvote that the comment received.
- `likes`: an integer representing the number of likes.
- `body`: a string that contains the content of the comment.
- `total_awards_received`: an integer representing the number of awards that the comment received.
- `score`: difference between ups and downs.
- `num_reports`: counts the number of reports since the last approval.
- `parent_id`: the identifier to the parent in the comment tree, i.e. the comment or post to which the comment responds to.
- `link_id`: the identifier of the item to the root of the comment tree, i.e. the post that started which induced the discussion.
- `created_at`: an integer representing the timestamp of the comment creation time.

The last field is the `author`: an instance of the *User Data Structure* related to the identifier of the author.

User Data Structure. Each User is characterized by 5 atomic fields:

- `id`: a string that uniquely identifies the user.
- `name`: a string that contains the user nickname.
- `link_karma`: is a measure of how valued your links are.
- `comment_karma`: is a measure of how valued your comments are by the community.
- `created_at`: an integer representing the timestamp of the user creation time.

2.2 Network Characterization

2.2.1 Definition. The network is defined by the pair $G = (V, E)$ where V is the hashmap of the nodes and E is a set of edges. An edge is a pair (u, v) where u and v are two keys of the hashmap V. When in the pair (u, v) the order is not considered we assume the graph as undirected and it is denoted by G; while if the order is relevant we consider the graph as directed and it is denoted by $\overrightarrow{G}$.

The nodes. The hashmap V contains as keys all the identifiers of the cralwed users, i.e a node exists if its identifiers appears as author in some comment or post. Each key is associated with a value which is an instance of *The Node Data Structure*:

- `data`: An instance of the *User Data Structure* related to the key, i.e. the `id` the of the user.
- `submission`: a set of instances of the *Post Data Structure*, which refer to all posts of which the user is the author.
- A set of toxicity labels that takes values in the continuous range [0, 1] and measures the different types of toxicity alignment of the user.
  - `labels_toxicity`;
  - `labels_severe_toxicity`;
  - `labels_obscene`;
  - `labels_threat`;
  - `labels_insult`;
  - `labels_identity_hate`.
- `labels_political_leaning`: it is an attribute that takes values in the continuous range [0, 1] and measures the political alignment of the user.
- `labels_political_leaning_cat`: it is the categorical version of `labels_political_leaning` provided by enforcing a threshold of 0.5 on the value, i.e. left for values < 0.5, right otherwise.
- `geo`: it is a categorical attribute which assigns to each node the region of the United States to which the corresponding user belongs.

Note that all the labels of the `The Node Data Structure` were obtained as the average of the results obtained by applying the respective models on each post and comment of a given user.

The edges. The existence of an edge indicates the presence of one or more comments between two users; if the graph is directed then the order in the pair of nodes (u, v) is such that: the first component is the author of the comment, while the second one is the recipient user.

It is possible that there are more than one comment between two users even considering the order, therefore considering the edges individually both G and $\overrightarrow{G}$ are multi-graphs. For simplicity the edges have been considered as super-edges weighted by the number of edges present in the starting multi-graph and whose information was given by the union of the data present on the edges. From now on G and $\overrightarrow{G}$ will be considered to be composed of only super-edges and we will therefore unambiguously refer to them as edges.

Each edge is associated with an instance of the *Edge Data Structure*:

- `comments`: is a set of instances of *Comment Data Structure*, one for each comment that the represents, i.e. the cardinality of this set is the weight.

- `weight`[2]: an integer that indicates the number of edges in the multi-graph.

3 NETWORK ANALYSIS

The graph described in section 2.2 is composed of 103968 nodes; if the undirected graph G is considered the number of edges is 461912; whereas if $\overrightarrow{G}$ is considered the number of edges is 505118. The weight average considering G is ≈ 1.138 with a standard deviation of ≈ 1.61; the distribution shown in Figure 1(a) highlights how not many users have more than one interaction with the same user, this may indicate that they do not react so much to a comment because of the author but rather for the content. The same weights behaviour described for G also applies to $\overrightarrow{G}$.

3.1 Connected Components

The analysis of connected components have been performed on G. There are only 6 different connected components sizes(Figure 1(b)), the giant one is composed of 95168 nodes whereas the others are made up of between 1 and 5 nodes. It is interesting to notice the big amount of isolated nodes, namely users that have published a post to which anybody has not responded to. Moreover those users did not respond to any posts or comments (in the time period analyzed).

3.2 Giant component

The analysis from now on focuses on the giant component, since it represents the most mature part of the evolution of the graph.

3.2.1 Node degree. The average node degree in the giant component is ≈ 9.6822 with a standard deviation of ≈ 54.5249. The degree distribution in Figure 1 (c) follows a power-law as is often the case in real networks.

3.2.2 Clustering coefficient. The average clustering coefficient in the giant component is ≈ 0.0347 with a standard deviation of ≈ 0.1293. The low value reported is a consequence of the absence of the concept of friendship on Reddit - the action of posting a comment by a user relies only on the content regardless who is its author. This property decreases the presence of the triadic closure within this social network.

3.2.3 Geodesic distances. Given the size of the graph, it is unfeasible to compute the distances of all possible combinations of two nodes. Therefore, an unbayased estimator - a Monte Carlo approximation method - was used to estimate the mean geodetic distance and diameter: each episode consists of the extraction of a pair of nodes between which the minimum distance is computed. The number of episodes δ, determines the accuracy of the estimate. As the number of draws increases the estimate goodness also increases, the same apply to the time complexity. This procedure has $O(\delta \cdot |V|)$ space-complexity and $O(\delta \cdot (|V| + |E|))$ time complexity. The procedure described above has been used on the network using $\delta = 1000000$:

- Mean geodetic distance $\langle d \rangle = 3.817$
- Diameter $d_{max} = 10$

[2]When the weight is considered as cost its value is the inverse.

3.2.4 Centrality. The analysis on centrality relies on two algorithms namely: the Eigenvector centrality and the Pagerank centrality. The distributions of both centralities (Figure 2) show a similar pattern, i.e., a distribution skewed toward very small long-tailed values. The observed numerical results are shown in Table 1

	mean	stdev	min / max
Eigenvector	$0,00032$	$0,0032$	$1.8e-19$ / 0.67957
Pagerank	$0,00001$	$0,00009$	$2.0e-6$ / 0.02383

Table 1: Centralities numerical results

Analyzing the top 5 scores obtained in the centrality analysis we obtained respectively:

- Eigenvector:
  0.680, 0.173, 0.155, 0.141, 0.127
- Pagerank:
  0.024, 0.007, 0.003, 0.002, 0.002

It is interesting how analyzing the intersection of these two sets, the first two users appear in both centralities as first. Investigating these two users we discovered they are moderators, as such they have many interactions but not necessarily relevant to content but only at administrative level. Proceeding with the analysis the other users, also in the ones in the intersection of top-5, appear to be standard users and not moderators.

Finally, we computed the homophily value applying the assortativity using as discrete attribute the user's country. This analysis aimed to measure how geographically close nodes tend to be connected with higher probability than expected. The result obtained using the state is 0.0077 hence very close to no assortative mixing.

3.3 Synthetic data

To be able to make fair comparisons the synthetic graphs are all indirected, we have changed our graph into a indirected one. The degree distributions of the synthetic graphs are shown in image 3. All synthetic graphs were generated using the same number of initial nodes as our graph.

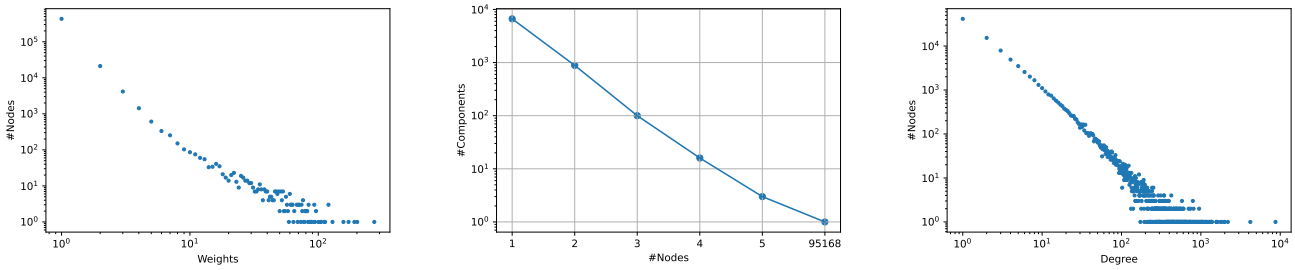
models	# nodes	density	<k>	<C>
rpolitics	95168	$1.0 \cdot 10^{-4}$	9.68	0.0346
ER	103950	$8.5 \cdot 10^{-5}$	8.87	$8.0 \cdot 10^{-5}$
WS	103968	$7.7 \cdot 10^{-5}$	8	0.473
BA	103968	$5.8 \cdot 10^{-5}$	6.0	$7.9 \cdot 10^{-4}$
configuration	95155	$8.29 \cdot 10^{-5}$	9.40	0.0281

Table 2: Statistics largest components

3.3.1 Comparison with ER model. The Erdos-Renyi graph has been sampled using the pair of paremeter (n, p): in order to make the comparison meaingfull with our graph

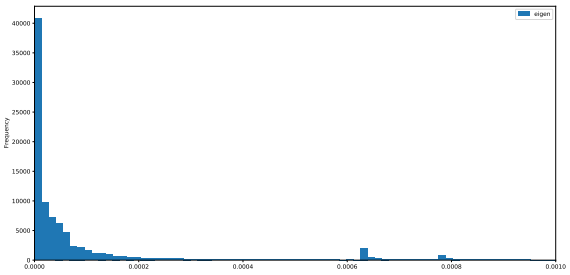
- n has been set as $|V|$
- the density of G has been used as value for p.

The obtained graph is very different, this is somehow expected since the Erdos-Renyi model does not reflects a lot of the real networks characteristics, e.g. the small world assumption. In particular:

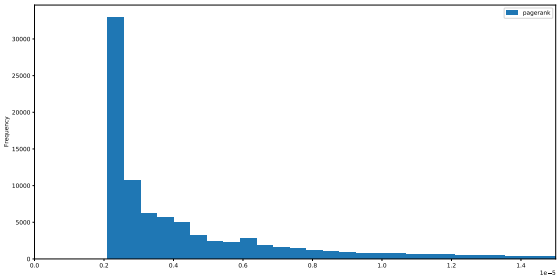


(a) Weight distribution considering G **(b) Connected components distribution considering G** **(c) Node degree distribution considering G**

Figure 1: Network analysis plots



(a) Eigenvector



(b) Pagerank

Figure 2: Centralities distribution G

- Since the distribution is a Poisson distribution (n is very large with respect to p) the largest connected component is made up of 99.9 of the initial nodes
- The clustering coefficient is three order of magnitude smaller (as well-known in theory) than the one in G

This ending up in having only the imposed measure similar, i.e. the number of nodes and the density.

3.3.2 Comparison with Watts-Strogatz model. The Watts-Strogatz graph has been sampled using the pair of paremeter (n, k, p): in order to make the comparison meaingfull with our graph

- n has been set as $|V|$;
- k has been set as the avarage clustering coefficient of G.
- p has been tuned into the range 0.1 - 0.001 in order to guarantee the small world property.

The value chosen for p has been 0.1 as it is the one that guarantees a value for the clustering coefficient closer to that of our graph.

In this case the similarity with respect to G is on the presence of the small world property and the obvious ones induced by the choice of parameters. The fundamental difference with respect to G (as expected from the theory) is in the clustering coefficient - the triadic closure is strongly present.

3.3.3 Comparison with Barabasi-Albert model. The Barabasi-Albert graph has been sampled using the pair of parameter (n, m): in order to make the comparison meaingfull with our graph

- n has been set as $|V|$;
- m - the value of the number of edges to add to each node once inserted in the graph, have been tuned into the set $\{1, 3, 4, 5, 7\}$ in order to guarantee the small world property.

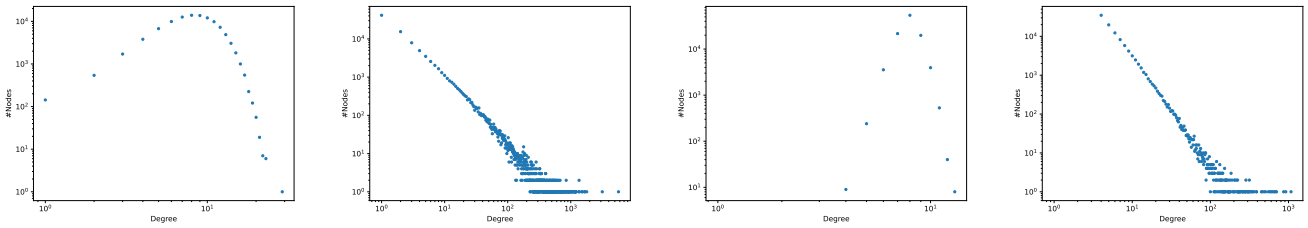
The value of m doesn't affect the degree distribution shape, but affects the average degree and thus shift the distribution on the x axis, the chosen value is 3.

The Barabasi-Albert graph is the synthetic graph with statistics most similar to G, this due to the preferential attachment, indeed it is also present in G. Hence the probability that two nodes are connected increases as their level of activity increases, if the two users post or comment often it is likely that an interaction between them can take place. For this reason it can be said that the probability that two nodes are connected is given by the degree of the two nodes. The degree distribution is very similar to the one of G, although the Barabasi-Albert graph has a smaller average degree, the same holds for the clustering coefficient.

3.3.4 Configuration Model. The configuration model is among the sampled synthetic graphs the one that is closest to the graph G ,since it is generated by imposing the list of degrees that the nodes must have.

4 COMMUNITY DISCOVERY

In order to perform the community discovery task we used the undirected version of our graph. The reference library is `cdlib` and



(a) Erdős–Rényi (b) Configuration Model (c) Watts-Strogatz (d) Barabasi-Albert

Figure 3: Degree distribution of synthetic networks

the selected algorithms have been chosen also taking into consideration the computational cost. For this reason, even if it would have been interesting, it was not possible to use the algorithms based on node attributes (i.e `eva` and `ilouvain`). The chosen algorithms are: `leiden`, `label propagation`, `infoMap`, `SCAN` and `ANGEL`. To evaluate the goodness of the found communities and to compare the results of the different algorithms, both internal and external comparisons were analysed. Internal comparisons were made through the use of fitness functions while external comparisons were carried out through partition comparisons.

4.1 Internal Evaluation

The fitness scores are shown in Table 3 and the used fitness function are:

- Internal Edge Density (IED): The internal density of the community set.
- Average Internal Degree (AID): The average internal degree of the community set.
- Newan-Girvan modularity (NGM): Difference between the fraction of intra community edges of a partition and the expected number of such edges
- Fraction over Median Degree (FMD): Fraction of community nodes of having internal degree higher than the median degree value.
- Conductance (C): Fraction of total edge volume that points outside the community.
- Triangle Participation Ratio (TPR): Fraction of community nodes that belong to a triad.
- Number of communities (n_c): Number of found communities
- Max community size (max_c): Size of the found community

	Leiden	Label Prop.	InfoMap	SCAN	ANGEL
IED	0.793	0.852	0.042	0.646	0.713
AID	1.294	1.170	5.800	1.631	1.577
NGM	0.276	0.0180	8.770e-05	3.592e-04	-0.219
FMD	0.102	0.090	0.389	0.223	0.287
C	0.353	0.361	0.066	0.327	0.962
TPR	0.009	0.003	0.121	0.011	0.028
n_c	318	2633	2	21	409
max_c	15423	88482	95144	93544	18607

Table 3: Fitness scores

In addition to this fitness function we have analysed also how the communities are affected by the political leaning of the users, for this reason three additional statistics have been computed:

- Average Political Score (APS): It's the average political score of the communities.
- Right Communities (RC): It's the number of communities with an higher number of right-wing users then left-wings.
- Left Communities (LC): It's the number of communities with an higher number of left-wing users then right-wings.

The results of this additional statistics are shown in Table 4. Unlike fitness functions, these attributes do not seem to be relevant for characterising the found communities. For this reason they will not be referred to in the analysis of the CD algorithms.

	Leiden	Label Prop.	InfoMap	SCAN	ANGEL
APS	0.253	0.246	0.221	0.182	0.233
RC	31	202	0	1	8
LC	275	2431	2	20	421

Table 4: Political leaning statistic

4.1.1 Leiden. Leiden as it is designed with the aim to create a set of communities that maximize the modularity - maximize the number of edges between the nodes within the same community and minimize the number of edges between nodes in different communities. This is clearly visible in the reported numerical results, as each modularity-dependent metric (NGM, FMD) is optimized. Moreover, the found communities show an high edge density, as result they are very cohesive and self-contained.

4.1.2 Label propagation. Label propagation is an algorithm that find communities using only network information. It start assign a unique label to each node and than through an iterative process propagate those labels. In our network label propagation find a huge community (88482 nodes) highly connected and which shows statistical properties very similar to those of the starting network. Moreover, it finds a larger number of small communities very uncohesive. This behavior is due to the low value of the clustering coefficient which plays an important role in label propagation.

4.1.3 InfoMap. InfoMap is designed to create a set of communities that minimize the conductance - the fraction of edges volume that points outside the community. Due to its nature InfoMap create a

huge communities and a set of small communities that contain almost all the node of the graph. This is explained by the fact that the network has a relatively low clustering coefficient, i.e. considering a node v already in the community and the addition of its neighbor u, it is very likely that most of u's neighbors are not already in the community. Therefore, each time a node is added to a community, in order to minimise the conductance, many of its neighbours must also be inserted. Iterating this process leads to the insertion of many nodes and explains the the presence of single large community.

The numerical results follow what has been said in fact the largest community has 95144 nodes and shows statistical properties very similar to those of the starting network.

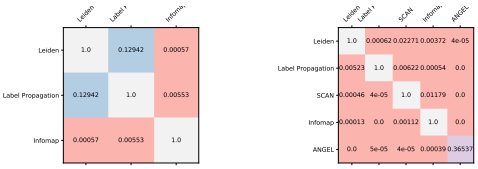
4.1.4 SCAN. SCAN is an algorithm that determines communities based on a structural similarity measure - nodes are grouped into the communities by how they share neighbors. Since the clustering coefficient of the network has an high variance, it is difficult to find the right value for the parameter ϵ. Using an high value of ϵ SCAN struggles to find core nodes, when an hub occurs a lot of nodes are added into the community. On the other using lower value of ϵ, many nodes are nominated as core, this results in many non-significant communities. This behavior, as the numerical results show, leads to the creation of a huge community which reflects the statistical property of the prior network.

4.1.5 Angel. The idea behind the design of Angel is that real networks are often complex object, hence the intuition is then to work on smaller local components, the ego-networks. For each node v its ego network E_v is extracted and it is used to identify micro-scale communities, in the end those communities are merged in mesoscale ones. This design really fit well in our scenario, because the assumption upon which the algorithm relies is that locally each node is able to identify its communities. In our network if we consider a generic node v it is a user who either wrote a comment to a post or wrote a post that someone else commented on. In both the case the assumption is that the node v shares with each of its neighbors an interaction on a certain post. Considering this property for a generic node v and one of its neighbors j, under the assumption that users comment more on the basis of topics than on the basis of who wrote them, the neighborhood of j probably had indirect interactions with v, e.g. replied to a comment to which v also replied to, or replied to a comment on a post of v. Considering the above and using the concept of ego-networks, can be said that ANGEL implicitly consider indirect interactions.

The numerical results find explanation in the above. NGM and the conductance have respectively a low and an high value. Presumably this due to the fact that the indirected interactions are not present in the graph, hence the ratio tends to the external communities. On the other hand the density is pretty high considering the comparison with other algorithms, this suggests that nodes within the communities found have a high number of direct interactions.

4.2 External Evaluation

Regarding the external evaluation the community discovery algorithms were compared using the F1 score, furthermore all those algorithms having reached full coverage were compared via Normalized Mutual Information (NMI), the results are shown in Figures 4. As expected and highlighted in the internal evaluation section, there are no particular similarities between the found communities. The algorithms that from this evaluation seem to be more similar are Label Propagation and Leiden with a value of Mutual Information equal to 0.0129.



(a) Mutual Information **(b) Normalized F1**

Figure 4: External evaluation scores

5 OPINION DYNAMICS

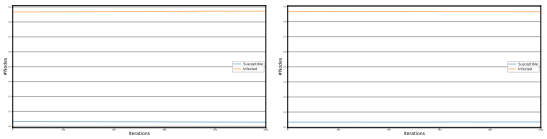
This section will go through an opinion dynamics analysis performed on the sub-graph induced by the set of node having the geo attribute.

5.1 Discrete analysis

The discrete algorithm used are: Voter/Qvoter model, MajorityRule and Sznajd.

5.1.1 Voter Model. The voter model is a discrete algorithm which assign randomly (or based on a specific distribution) a binary state which classify nodes in infected and susceptible. The algorithm at each iteration randomly select a node $v \in V$ and a node $u \in N(v)$ and assign to v the opinion of u. A generalization of this algorithm also applied on our graph is QVoter, which takes at random a node v and q of its neighbors: if all agree on a such opinion they influence one neighbour at random.

Note in the geo-graph 7710 nodes have 1 neighbors, 2887 have 2 neighbors and 1459 have 3 neighbors. Since the distribution of the size of the neighborhood is skewed towards 1, if the distribution of the label is skewed towards infected or susceptible than for both the algorithms is hard to have a relevant dynamic of opinions. This explains why when an initialization of the labels based on the political leaning is used than it is not possible to effectively propagate an opinion in the graph, since the political orientation is strongly unbalanced to the left. The results are shown in Figure 5.



(a) Political leaning distribution **(b) Random distribution**

Figure 5: Voter results with unbalanced distribution

For this purpose it is interesting to note that the percentage distribution of political leaning in the subset of vertices with 1

neighbor is very different from that apriori in the graph, in fact it shows a significant component of the the right-wing.

Using instead a more balanced representation, e.g. 50% infected and 50% susceptible, significant dynamics can be observed. The results are shown in Figure 6.

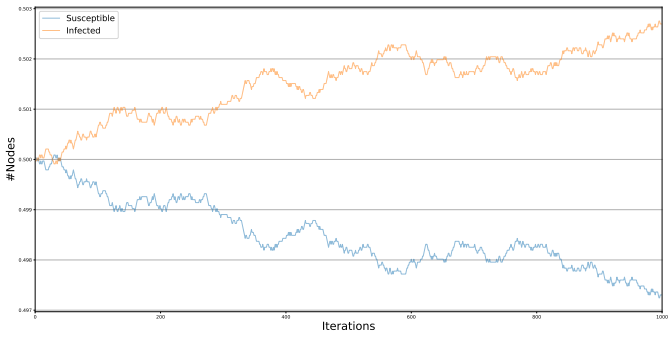
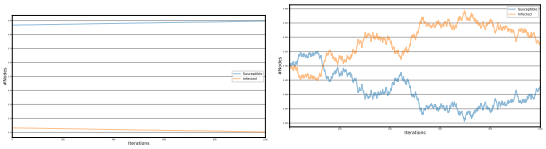


Figure 6: Voter results with balanced distribution

5.1.2 MajorityRule. MajorityRule is a discrete algorithm which select randomly set of r nodes and assign to each of them the opinion which occurs most in the set.

As in the case of Voter an unbalancing in the distribution can generates inefficiency. If we considered an unbalanced distribution towards infected (without loss of generality) than the probability of an infected node being chosen for comparison is very low - the probability of changing its opinion is low. The result is an inefficient process where even a considerable number of iterations is not sufficient to show relevant dynamics. The results are shown in Figure 7



(a) Political leanign distribution **(b) Random distribution (equal-proportion)**

Figure 7: Majority Rule results

5.1.3 Sznajd. Sznajd is a discrete algorithm which assign randomly (or based on a specific distribution) a binary opinion +1 or -1. At each timestep a pair of adjacent agents is randomly selected, if both have the same opinion then it is taken by all their neighborhood. The algorithm, as well known in theory, converges towards one of the two opinion depending on the initial condition. Using political leaning to distribute the two opinions in the graph (left-wing and right-wing) it is possible to see in Figure 8 (a), how the graph converges to the left opinion, being the distribution of the option strongly skewed towards the left-wing.

The same experiments were performed using two random distributions:

(1) a distribution that follows the same proportions as those of political leaning, but the assignment of the opinions is randomly performed.
(2) A distribution with equal proportions between the two opinions.

The resulting using such distributions are shown in Figures 8 (b) and (c). Interestingly, using the distribution of the political leaning, the algorithm converges slowly than a random distribution with the same proportions. Therefore, the topology imposed by the nodes of the two opinions disfavors the change of opinion. The behaviour is highlighted in Figure 8 (d)

5.1.4 Complete graph comparisons. For computational reasons, the comparisons between our graph and the complete graph were made using a subgraph of the geo-graph. The subgraph was created by including the nodes and edges encountered by running a BFS on a randomly chosen node. In the creation of the subgraph also the political leaning has been taken in consideration, namely the subgraph and the initial graph have the same percentage of right-wind and left-wind users.

The algorithms used for the comparisons are the same as those described above. When Voter and QVoter are used, the complete graph tends to be more stable and maintain the initial percentage of infected users. Using the Majority Rule algorithm, since it does not use the topology of the graph, very similar results are obtained between the subgraph and the complete subgraph. Instead, using Sznajd the behaviour is totally different, in fact on the complete graph, due to the way Sznajd is defined, it is straightforward that in one iteration the algorithm converges to one of the two opinions.

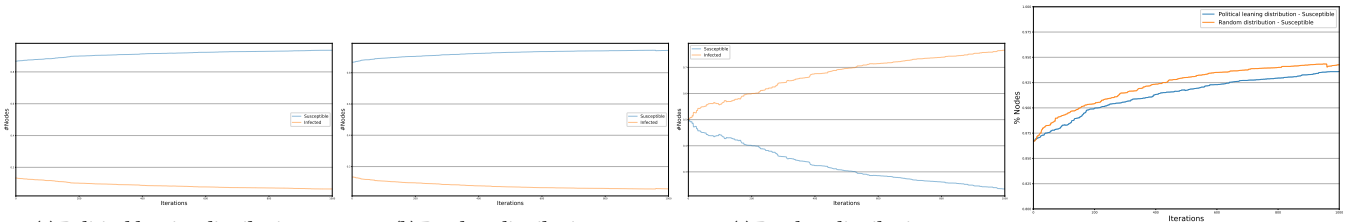
5.2 Continuous analysis: Algorithmic Bias

Algorithmic Bias is a continuous algorithm which assign randomly an opinion $\in [0, 1]$ to each node in the graph. At each time step a pair of node (u, v) is extracted s.t. u is randomly chosen, while v is chosen with a probability that decreases with the distance w.r.t. the opinion of u. The parameter that controls how this effect is significant is γ. After that if the distance between u and v opinions is under a certain threshold, namely the bounded confidence ϵ, then the two nodes take the average of their opinion. If $\gamma = 0$ the algorithm works as the Deffuant model - no bias.

Due to the computational cost we applied the algorithm to a subgraph of the geo-graph, built as described in the section 5.1.4. Different parameters were tested to verify in case of bias which situations occurred, both using the real graph and the complete one. As we expect from the theory in the absence of bias there is a convergence towards a common opinion, results are shown in figures 9 (a) and (b). Introducing bias in the algorithm greatly slows down the convergence process in the case of the real graph and shows opinion fragmentation (which intensifies as ϵ shrinks), results in figures 9 (c) and (d).

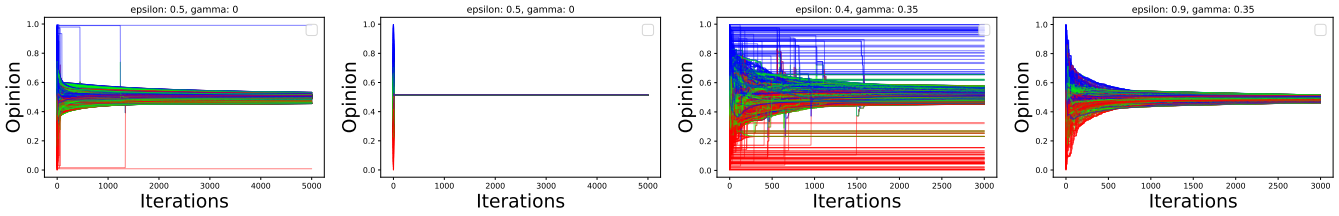
6 GRAPHLETS CENSUS

Graphlets are small non-isomorphic induced subgraphs of a large network. This section is intended to describe the approach used for counting connected graphlets of size 3 and 4. All the possible



(a) Political leaning distribution (b) Random distribution 1. (c) Random distribution 2. (d) Comparison between political leaning and random distribution 1

Figure 8: Sznajd results



(a) Deffuant $\epsilon = 0.5$ (b) Deffuant $\epsilon = 0.5$ **(Complete graph)** (c) $\epsilon = 0.4, \gamma = 0.35$ (d) $\epsilon = 0.9, \gamma = 0.35$

Figure 9: Algorithmic bias result

configurations of graphlets of size 3 and 4 are 16. We are interested only in those connected, namely those shown in Figure 10. In the rest of this chapter to refer to a particular configuration we will use the nomenclature present in Figure 10.

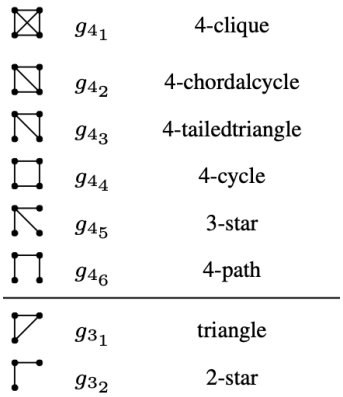


Figure 10: Connected Graphlets of size 3 and 4

The algorithm used is described in details in [1], which fits particularly well to our problem, because it is designed to be efficient in the case of sparse graphs.

The founding intuition is that by applying appropriate combinatorics properties on graphs it is possible to reduce the count of the number of graphlets of size 4 to that of the count of cliques of size 4 (G_{4_1}) and cycle of size 4 (G_{4_4}). The first step is to compute the number of connected graphlet of size 3, which strictly follow the definition:

- In order to compute G_{3_2} the algorithm goes through each edge (u, v) than visit both the neighborhood of u and the ones of v and count each triple.
- In order to compute G_{3_1} the algorithm goes through each edge (u, v) than visit first the neighborhood of u, then the ones of v and for each triple that respect G_{3_2} if the node considered is in a graphlet G_{3_2} also with u than it count a G_{3_1}.
- In order to compute G_{4_1} the algorithm start from the set of triangles found and try to complete it with edges in order to create a 4-clique. This procedure has time complexity $O(k_{max} \cdot T_{max})$, where T_{max} is the maximum number of triangles incident to an edge and k_{max} is the maximum node degree.
- In order to compute G_{4_4} the algorithm start from the set of 2-starts found and try to complete it with edges in order to create a 4-cycle. This procedure has time complexity $O(k_{max} \cdot S_{max})$, where S_{max} is the maximum number of stars incident to an edge.

The computational cost of an iteration is absorbed by the counting of the 4-cliques and 4-cycle. Since there are as many iterations as the number of arcs in the graph, the algorithm has a total complexity of $O(|E| \cdot max\{k_{max} \cdot T_{max}, k_{max} \cdot S_{max}\}) = O(|E| \cdot k_{max} \cdot max\{T_{max}, S_{max}\})$.

Given the number of 3-cycle, 3-stars, 4-clique and 4-cycle it is possible to derive the number of remaining graphlets:

$$G_{4_2} = \sum_{e \in E} \binom{G_{triangle(e)}}{2} - 6 \cdot G_{4-1} \tag{1}$$

$$G_{4_3} = \sum_{e=(u,v) \in E} G_{triangle(e)} \cdot (G_{2-star(u)} + G_{2-star(v)}) - 4 \cdot G_{4_2} \tag{2}$$

$$G_{4_5} = \sum_{e=(u,v)\in E} \binom{G_{2-star(u)}}{2} + \binom{G_{2-star(v)}}{2} - G_{4_3} \quad (3)$$

$$G_{4_6} = \sum_{e=(u,v)\in E} G_{2-star(u)} \cdot G_{2-star(v)} - 4 \cdot G_{4_4} \quad (4)$$

The execution on the network has generated the following results:

- $G_{4_1} = 1045823$
- $G_{4_2} = 519037725$
- $G_{4_3} = 43963129$
- $G_{4_4} = 25349648$
- $G_{4_5} = 369310606929$
- $G_{4_6} = 1512599487$
- $G_{3_1} = 265507$
- $G_{3_2} = 72889923$

7 OPEN QUESTION

This section discusses three topics of interest with respect to specific biases and behavioral phenomena emerging from the graph:

(1) **Implicit edges**: This question is intended to investigate the community detection algorithms, in particular Angel. We sought for evidence that the communities found by this algorithm have a significant number of implicit edges, so they are more connected than numerically appears.
(2) **Geographical bubble**: Using the user's country, we looked for behavioral biases related to it: specifically how much the variety of states it interacts with is related to other topological factors.
(3) **Toxicity analysis**: We analyzed the toxicity of a message and tried to understand how much this affected the popularity of a post.

7.1 Working with implicit edges

In section 4.1.5 were shown the results obtained by the community discover algorithm Angel. In this section is presented the following conjecture: the locality assumption and the mechanisms based on Angel's ego-networks create communities that are actually denser than what is shown by the numerical data. This happens because Angel implicitly adds many triples of nodes for which there is an implicit edge.

In order to proceed, it is necessary to give the definition of an implicit edge:

Definition 1. Given a 2-star graphlet composed of the triple (v, u, e) since it is a 2-star then $(v, u) \in E$ and $(u, e) \in E$. The triple (v, u, e) has an implicit edge if only if $(v, e) \notin E$ and both v and e interacted on the same post. From now on, we denote triples that respect this property as $\overline{(u, v, e)}$

For simplicity we denote the i-th Angel community as $C_i = (V_i, E_i)$ and we call $2star(C_i)$ the set of triples $(u, v, e)\ s.t.\ u, e \in V_i$ that are in a 2-star graphlet i.e $(u, v), (v, e) \in E_i$.

For each community we are interested in analysing the number of triples that satisfy the definition of implicit edge. The metric we have derived is the ratio between all the triples having an implicit edge and those that could had one, formally:

Definition 2. The score $\Omega(C_i)$ assigned to the community C_i is defined as:

$$\Omega(C_i) = \frac{|\{(u, v, e) \in 2star(C_i) \wedge \overline{(u, v, e)}\}|}{|\{(u, v, e) \in 2star(C_i) \wedge (u, e) \notin E\}|}$$

We computed the Ω score on all significant Angel communities. A community is considered significant if it is not the giant one found and if it is at least 6 nodes. To confirm what we expected the phenomenon is very frequent in these communities, the numerical data are present in table 5. The average Ω score is 0.6414.

	numerator	**denominator**	**Ω**
C_1	42	52	0.769
C_2	42	42	1.0
C_3	42	48	0.875
C_4	0	0	0
C_5	32	32	1.0
C_6	24	24	1.0
C_7	0	0	0
C_8	2	10	0.2
C_9	6	14	0.4
C_{10}	2	10	0.2

Table 5: Ω scores of the firsts Angel communities

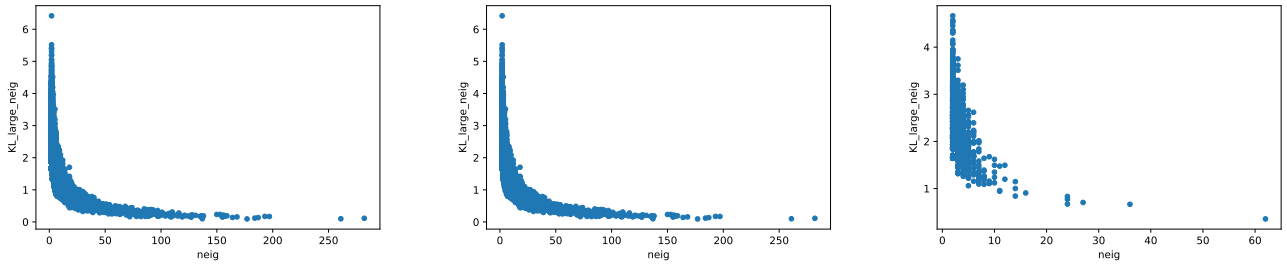
7.2 Geographical influence on topology

A question that was attempted to be addressed is the following: "is there a connection between the geographic position - the user's country and the way it is connected to the rest of the graph?" In this section an attempt to give a positive empirical answer to this question is presented.

7.2.1 The problem. The first sub-problem to face is to clearly define what you want to look for: we focused on analyzing for each user $v \in V$, its neighborhood $\mathcal{N}(v)$. In particular the interest was on observing particular behaviors in how the states were distributed in the neighborhood and at the same time how the neighborhood was constructed. The measure in which we are interested in is the size of the neighborhood which is indeed straightforward to compute. On the other hand, a measure that quantify how the states are distributed in the neighborhood requires a discussion, since there are multiple choices and each of them could potentially show different results.

In order to define a specific measure, it is therefore necessary to refine the initial question: the focus then shifted to answering the question, how is a user open to having interactions with different states? How this is related to the size of its neighborhood?

7.2.2 Proposed approach. To measure this phenomena we decided to exploit the information divergence or relative entropy applied to the following distributions:



(a) all the users **(b) only users classified as left-wing.** **(c) only users classified as right-wing.**

Figure 11: $D_{KL}(v)//|N(v)|$: on the y-axis the information divergence on the x-asis the dimension of the neighborhood

(1) **empirical states distribution in the neighborhood**: for each node $v \in V$ the empirical distribution of its neighborhood was extracted, counting the number of occurrences of each state in neighboring users dividing it by the number of nodes in the neighborhood. This distribution is denoted with $P(v)$.
(2) **empirical states distribution in the graph**: for the entire graph the empirical distribution of states was calculated by counting the number of occurrences of each state and dividing it by the number of nodes in the graph having the geo attribute. This distribution is denoted with $Q(G)$.

Definition 3. For each node v the relative entropy $D(v)_{KL}$ is:

$$D(v)_{KL} = D_{KL}(P(v)||Q(G)) = \sum_x P(x) log\Big(\frac{P(x)}{Q(G)}\Big)$$

The relative entropy has a very specific meaning, it tells us how much each state distribution of the neighborhood is similar to the states distribution in the graph.

7.2.3 Results. The analysis is reduced to the sub-graph for which it was possible to get the geographical information, it contains 30979 nodes and 54709 edges. We will refer to this sub-graph as the geo-graph. Thus, all the presented results are biased on the available data.

For each node v in the geo-graph having at least one node in the neighborhood we computed the $D(v)_{KL}$, the results are shown in Figure 11 (a). In Figures 11 (b) and (c) have been filtered out the two nodes having a neighborhood size grater than 1000 (which are only 2).

Several aspects already emerge from this first results:

- users tend on average to have direct interactions with few users and therefore have a very high relative entropy;
- users who tend to be "more open" - in the sense of writing to many people, have a low relative entropy $D(v)_{KL}$, namely they are less prone to close themselves in a geographical bubble.

Subsequently, the same analysis was made by discriminating users on the attribute *labels_political_leaning_cat*, obtaining the sets of right-wing and left-wing users.

The results shown in Figure 11 (b) and (c) highlight how the right-wing users tend to be more "closed" - have a restricted neighborhood and therefore they have an higher relative entropy. This could be caused by the fact that the graph is very unbalanced towards the left-wing, being /r/politics frequented by many users of that political wing.

It could be argued that choosing Q as the distribution for the relative entropy it disadvantages the nodes with the small neighborhood because the distribution takes into account states that not occurs in the neighborhood. To further validate our analysis, we defined a new distribution $\bar{Q}(v)$ specific to each node: instead of considering all the nodes of the graph we consider only the nodes that have as state a value that occurs in $N(v)$, namely $V(v)$. We then computed the empirical distribution by counting the number of occurrences of each state divided by the number elements in $V(v)$. The results obtained are analogous to those described, Figure 12.

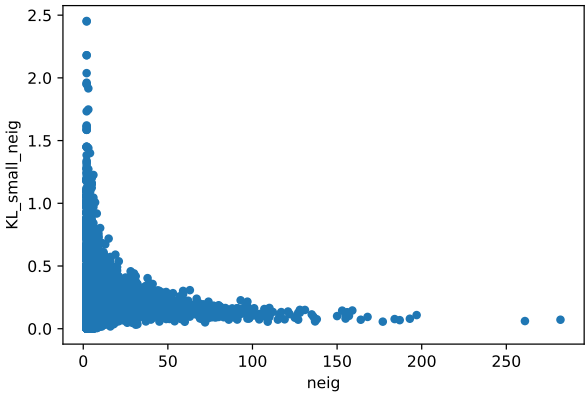


Figure 12: Result using $\bar{Q}(v)$

7.3 Characterizations of toxic interactions

Ultimately, in this section is presented an analysis made on the graph $\overrightarrow{G}$ having as a focus the toxicity attribute. We built a set of sub-graph of $\overrightarrow{G}$, the $t3 - partitioning$, defined as:

Definition 4. t3-partitioning: $G_{t3-partitioning}$ is a set of sub-graph of $\overrightarrow{G}$ s.t. $\forall G(t3) \subset G, \forall v \in G(t3)$ the following implication holds:

$G(t3) \in G_{t3-partitioning} \implies v$ is involved in the post with id t3

It follows from the definition that the number of subgraphs in $G_{t3-partitioning}$ is equal to the number of posts contained in $\overrightarrow{G}$

Subsequently, for each of the subgraph $G(t3) \in G_{t3-partitioning}$ the number of occurrences of toxic comments was counted - a comment is toxic iff its toxicity value is ≥ 0.5. The number of total comments was compared with the number of toxic comments in order to check if there was a correlation, the results are shown in Figure 13. The results show the hypothesized correlation, these results are obviously biased on the model used to compute the toxicity and the time period considered. We therefore sought further confirmation of this correlation to reduce the effect of the bias. In the first instance we observed whether the same correlation was shown in the case where instead of considering the number of comments we considered the number of users who participated in the post. Using this metric we get the same results (Figure 14) as the previous one with imperceptible variations, This due to the fact that often happens a user only one or a few interactions within the same post.

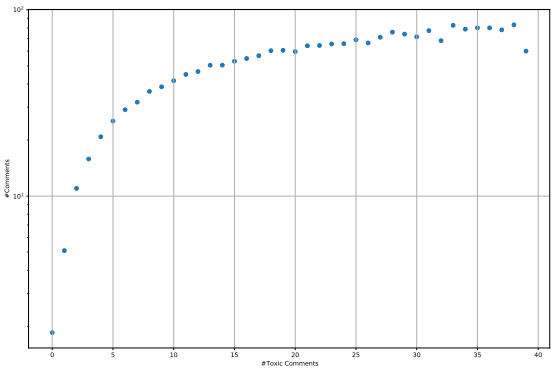


Figure 13: Number of toxic comments on x axis and number of comments (averaged over all the posts having such number of toxic comments) on y axis

Given what has been said before the focus has shifted to try to understand how toxicity could be in correlation with the increasing the posts activity. The definition of an active user is the following:

Definition 5. A user $v \in V$ is said to be active if and only if $\exists u \in V$:

$(v, u), (u, v) \in E \wedge (v, u)$ occurs two times within the same post

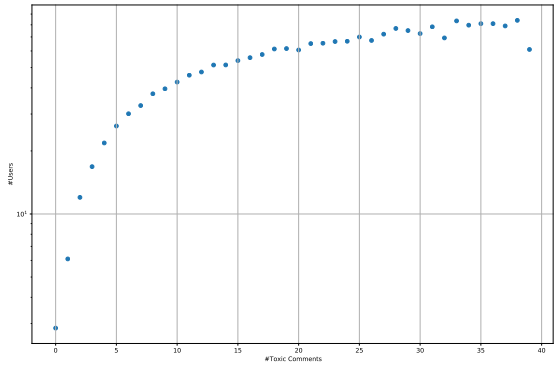


Figure 14: Number of toxic comments on x axis and number of users (averaged over all the posts having such number of toxic comments) on y axis

In practice a user is considered active if he/she answers a post or a comment and the author of that comment answers. Finally the user back to answer the new comment.

Given the definition of active user, we tried to analyze if there is a correlation between the number of active users and the number of toxic comments. The results are presented in Figure 15 and show the sought correlation.

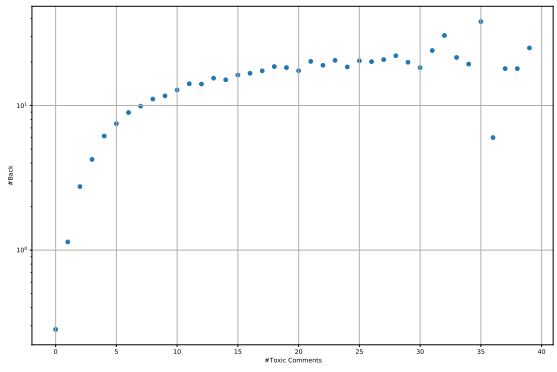


Figure 15: Number of toxic comments on x axis and number of active users (averaged over all the posts having such number of toxic comments) on y axis

It could be argued then that in our graph the presence of a strong component of toxic comments is correlated to the activity of the post.

7.3.1 Monopoly of discussions. As mentioned above our graph is strongly unbalanced towards the left wing, this could potentially disfavor discussions of the counterpart. In this section we will try

to briefly analyze this aspect.

We took for each topic in the graph the polarity of the topic: if the topic was classified as right-wing then it is contained in the set R, otherwise it is contained in the set L. Subsequently, the concept of monopoly has been defined as follows: a discussion is monopoly of the right wing if the average of political leaning is higher than 0.5, otherwise it is monopoly of the left. Finally, it was analyzed how many times the discussions in R ended up being a monopoly of the right-wing and how many times of the left-wing, the same has been done for the discussions in L. The results are shown in Table 6.

set	left-wing monopoly	right-wing monopoly
R	0.87	0.13
L	0.89	0.11

Table 6: Results of monopoly discussion analysis

A strong monopoly of discussion by the left-wing emerges from this metric even when discussions of departures are classified as right-wing.

Given the results about the monopoly of the discussion we sought to analyze how much this correlated with toxicity. It turns out that whenever the discussion is a right-wing monopoly there is about 50% of toxic comments, either the discussion is in R and in L. If instead a discussion is in L and it is a left-wing monopoly there is 17% of toxic comments. While, if initially a discussion was right-wing there there is 20% of toxic comments. Under the assumptions made then when a post goes to be right-wing monopoly, the percentage of toxic comments significantly increases.

8 CONCLUSIONS

The analysis presented above sought to reveal interesting patterns by primarily exploiting the topological information and specific attributes of each user. The network considered was built on a "hot" month, the one preceding the American elections in 2020. Therefore, some of the characteristics identified may be typical of that period, as it is close to a central event with respect to the discussed topics. Given the topic, the attributes of interest were political leaning, user's country and message toxicity. First a statistical analysis of the network was carried out, where some of the characteristics frequently occurring in real graphs were assessed. Three analytical tasks were then tackled separately, namely: the counting of connected graphlets of dimension 2 and 3, the identification of communities using community discovery algorithms and an analysis of the diffusion of opinions in the graph using opinion dynamics algorithms. The most important goal of this work was to investigate different aspects of the network that can be traced back to the behavioural biases of /r/politics users. This analysis dealt with three independent aspects: the frequent presence of implicit connections captured by community discovery algorithms, the geographical influence on the choice of people a user interacts with and finally the impact of toxicity on post activity. Regarding the first aspect, a significant number of communities with a considerable number of implicit edges was found by the Angel algorithm. These subgraphs are implicitly more connected than the density alone would lead to think. The analysis of the impact of the geographic features is focused on the attribute of the user's country. Using a metric based on local entropy it was possible to investigate a specific pattern that /r/politics users have followed: "socially closed" users - those who have few interactions, tend to have those interactions with people from the same countries, creating local geographic bubbles. "Socially open" users - those with a larger neighbourhood, tend to have interactions with a greater variety of countries. It was also observed that the first set included the majority of users classified as right-wing. Finally, we analysed toxicity with respect to the activity of the posts: a correlation was observed between the growth of users participating in a post and the amount of toxic comments. The same correlation was found with respect to the number of comments and the number of active users.

REFERENCES

[1] Nesreen K. Ahmed, Jennifer Neville, Ryan A. Rossi, and Nick Duffield. 2015. Efficient Graphlet Counting for Large Networks. In *2015 IEEE International Conference on Data Mining*. 1–10. https://doi.org/10.1109/ICDM.2015.141

[2] Duilio Balsamo, Paolo Bajardi, and André Panisson. 2019. Firsthand Opiates Abuse on Social Media: Monitoring Geospatial Patterns of Interest Through a Digital Cohort. *CoRR* abs/1904.00003 (2019). arXiv:1904.00003 http://arxiv.org/abs/1904.00003

[3] Srayan Datta and Eytan Adar. 2019. Extracting inter-community conflicts in reddit. In *Proceedings of the international AAAI conference on Web and Social Media*, Vol. 13. 146–157.

[4] Laura Hanu and Unitary team. 2020. Detoxify. Github. https://github.com/unitaryai/detoxify.

[5] Jasser Jasser, Ivan Garibay, Steve Scheinert, and Alexander V Mantzaris. 2021. Controversial information spreads faster and further than non-controversial information in Reddit. *Journal of Computational Social Science* (2021), 1–12.

[6] Joan Massachs, Corrado Monti, Gianmarco De Francisci Morales, and Francesco Bonchi. 2020. Roots of trumpism: Homophily and social feedback in donald trump support on reddit. In *12th ACM Conference on Web Science*. 49–58.

[7] Gianmarco De Francisci Morales, Corrado Monti, and Michele Starnini. 2021. No echo in the chambers of political interactions on Reddit. *Scientific Reports* 11, 1 (2021), 1–12.

[8] Virginia Morini, Laura Pollacci, and Giulio Rossetti. 2021. Toward a Standard Approach for Echo Chamber Detection: Reddit Case Study. *Applied Sciences* 11, 12 (2021). https://doi.org/10.3390/app11125390

[9] Alexander James Richardson. 2020. Estimating the Impact of Political Propaganda on Reddit Users' Political Opinions. *Georgetown University* (2020).

[10] Ahmed Soliman, Jan Hafer, and Florian Lemmerich. 2019. A Characterization of Political Communities on Reddit. In *Proceedings of the 30th ACM Conference on Hypertext and Social Media* (Hof, Germany) *(HT '19)*. Association for Computing Machinery, New York, NY, USA, 259–263. https://doi.org/10.1145/3342220.3343662