The war discourse on Twitter. How your hashtags can characterize you

Giulia Diamanti
g.diamanti2@studenti.unipi.it
Student ID: 551186

Davide Perra
d.perra@studenti.unipi.it
Student ID: 616686

Fabio Michele Russo
f.russo11@studenti.unipi.it
Student ID: 429892

1 ABSTRACT

The launch of the Russian-Ukrainian conflict in February 2022 was an important topic of discussion on social networks. In this report we propose a study conducted through Network Science tools regarding the relationships that have developed between users who use Twitter during the first weeks since the beginning of the war in Ukraine in February 2022. Within our network, **nodes represent users** and **links represent interactions** between them. We have divided the users / nodes into 4 categories of opinion and analysed the network. We then studied trends with the goal of identifying the potential of opinion changes following interactions with users of a different category.

2 DATA COLLECTION

2.1 Data sources

We chose Twitter as a source for mining of data because we believe it is one of the most used social networks to discuss the war in question as well as being, more generally, one of the most popular platforms worldwide. The war broke out on February 24 but we started analyzing tweets from **February 15, 2022** to capture users' opinions as soon as the tensions started to increase, until March 15, 2022, thus considering a moment that historically has more fragmented people's opinions and in which news have noted an important spread of propaganda through Twitter. For most of the work, therefore, we use only **one month of tweets** for memory management and performance reasons, which were crucial considerations throughout the entire project due to the gigantic amount of tweets and related data and metadata. Tweets during this month were collected and a **snapshot of user interactions** was extracted. For the final phase of the open question, it was necessary to widen the temporal view to more than a single period.

2.2 Extraction methodology

Interfacing with Twitter through the official *academic API* and the `Twarc2` library we implemented several custom classes (in the `extract` module) that allowed us to homogenize the collection of tweets according to our needs, so that all members of the group could then quickly get a `pandas.DataFrame` and/or a `csv` with tweets according to the requested query, and that any dataframe obtained would be consistent to the same project specifications (e.g. start / end date of the search, language). In addition, due to the amount of data, memory management was crucial; thus, all efficiency operations have also been centralized in the `extract` module. Here is a description of the main classes implemented:

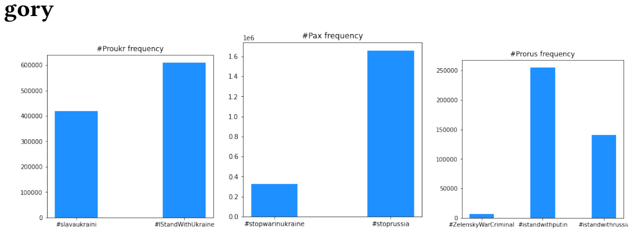
- `SocialETL` allows us to extract all tweets that contain a specified query;
- `Count` allows us to identify how often a given keyword appears by counting tweets that cointain it;
- `UserETL` accesses all tweets of a specified user, according to the time periods established by us;
- `DownloadTweets`, for a specified set of tweet IDs, collects only the attributes we need and builds with them a standardized database.

2.3 Hashtag categorization

At this stage we have categorized the hashtags collected into 3 categories, namely **pro Ukraine**, **pro Russia**, **Pax Romana**. To do this, we first manually identified 1 "parent" hashtag for each category and then extracted the "children" hashtags to be included in the category.

2.3.1 Parent hashtag selection. To choose the parent hashtag for each category, we identified possible candidates through manual searches on the main social networks and search engines; we then observed the frequency of each candidate in the time period of interest.

Figure 1: frequencies of candidate hashtags for each category



Finally, among the most popular hashtags we selected the three, one for each category, so that their frequencies would be as similar as possible. The table below sums up our choice with the respective frequency counts:

Figure 2: parent hashtag selected for each category and its count

category	parent hashtag	count
Pro Ukraine	#slavaukraini	419065
Pro Russia	#istandwithputin	255117
Pax Romana	# stopwarinukraine	322045

2.3.2 Extracting child hashtags. Once identified the parent hashtags, we extracted a list of child hashtags to assign to each of the 3 parent hashtags. These sets of **children + parent** make up each of our 3 categories. To extract the tweets we made a single query to the Twitter API containing the 3 parent hashtags: `#slavaukraini OR #istandwithputin OR #stopwarinukraine`. We collected a fixed maximum number of tweets: after several attempts we found that the maximum that our PCs could work on was **800,000**, which is the number of tweets used for this phase. To ensure the relevance of each hashtag worked we have created 3 thresholds:

- *threshold_support*,
- *threshold_certainty* and
- *threshold_ratio*.

Support.
We are only interested in hashtags that appear in a minimum number of tweets in our database. If a hashtag, for example, appears only in one tweet out of 800,000, we want to cross that hashtag out because with only one tweet we do not believe we can assign an opinion category to the hashtag with high confidence. In general, we define support = fraction of tweets in which the hashtag appears calculated as

$$\text{support} = \frac{\text{(number of tweets in which the hashtag appears)}}{\text{(total number of tweets obtained by our query)}}$$

and we define *threshold_support* = 0.9/10000, a number manually set as the minimum reasonable fraction of tweets in which a hashtag must appear to be included in the categorization phase. For a hashtag to be considered it must have

$$support \geq threshold\_support$$

We did several tests regarding the *threshold_support* and decided on the number reported after observing results and any reported idiosyncrasies. For example: set to 0.5 it turned out that the hashtag `#istandwithzelenskyy` was categorized as prorus, a semantic error. By doing this, we try and strengthen our work in regards to attempts by twitter users to *inject hashtags* into semantically wrong opinion categories, thereby "taking over" that hashtag search feed with malicious, opposing-view tweets.

Certainty.
Once we decided on the inclusion or exclusion of hashtags based on their *support*, we went on to calculate, for each hashtag, its **scores**. A hashtag's *score* for a given category represents that hashtag's relevance to that category, i.e. the relative frequency with which that hashtag was tweeted along with the parent hashtag for that category. An example:

```
#stophatingrussians the scores are
proukr    0.0696
prorus    0.4696
pax       0.4609
```

These 3 scores represent the fraction of the times `#stophatingrussians` was tweeted along with the parent hashtag of its category. For example, only 6.96% of the time it was tweeted along with `#slavaukraini`. Based on the higher score we categorize each hashtag as *proukr*, *prorus* or *pax*. However, we want there to be a minimum of certainty in categorization. Technically, a hashtag could be assigned to a category with a maximum relative score of 33.34% (the lowest possible maximum among three categories that add up to 100%). This does not satisfy us, so we have introduced *threshold_certainty* = 0.4 which represents the minimum that the score of a hashtag must be for the hashtag to be assigned to the category.

Ratio.
Finally, again to ensure a minimum level of certainty in categorization, we have introduced a third treshold. We note that in the example above, surely the hashtag does not belong to the *proukr* category. Among the other two categories, however, *prorus* and *pax*, the difference in score is minimal: 46.96% and 46.09%. We therefore introduced another threshold, *threshold_ratio=1.2*, which allowed us to avoid a hashtag being assigned to one category rather than another with a low difference in maximum scores. The formula used is:

$$max(scores) \geq 1.2 * median(scores)$$

With this threshold we therefore want to make sure that the hashtag is included in the most appropriate list: if a hashtag's maximum score reaches less than 20% more than its median score, we believe that there is not enough difference and therefore we discard it. Continuing the example above, we can see how the hashtag `#stophatingrussians` is then discarded and not placed in any of the three categories of hashtags. In order not to violate the *threshold_ratio* a hashtag with a maximum score of 0.47 its median score should have been **no more than** 0.47 / 1.2 = **0.39**.

Semantical cleaning.
A final processing step was to remove what looked to us like "non-representative" hashtags, which pass the *threshold_support* as they are widely used, but which we believe can't represent opinions about the war. Among these we divided between *"neutral"*, or **geographical places** that are semantically difficult to categorize, and *"trends"*, those hashtags used by users to increase views but that did not indicate a position of thought of the tweet contents. This processing, the "semantical cleaning", is the only one that has been carried out manually by us. We show the final

result of categorizing hashtags as the number of hashtags in each category and as wordclouds.

Figure 3: number of hashtags in each opinion category

category	number of hashtags
Pro Ukraine	781
Pax Romana	621
Pro Russia	152

Figure 4: wordclouds for each opinion category

2.4 User categorization

We have now categorized users, via what they tweeted, into one of four categories. Three categories are as discussed *proukr*, *prorus*, *pax* and the fourth is a residual category, *"don't care"*, in which are categorized users that don't tweet frequently about the war compared to their activity in posting tweets.

category	description	abv
Pro Ukraine	in support of Ukraine	proukr
Pro Russia	in support of Russia	prorus
Pax Romana	"peace at any cost"	pax
I don't care	they don't comment on the war on social media	dontcare

To get the tweets we used the same query we used in the hashtag phase: *(#slavaukraini OR #stopwarinukraine OR #istandwithputin)* and obtained a dataframe of 44812 lines, of which each line is formed by a tweet. We have carefully chosen the parameters (*pages* and *max_results*) in order to have the resulting number of nodes falls within the range of 10,000 to 15,000, following the preprocessing of the network. The authors of these tweets are 32,382 distinct users.

2.4.1 Tweet Categorization. Each tweet was categorized into one of the top 3 classes based on the hashtags it contained, with the simple majority rule. In the event that a tweet had a tie between two or more classes (example, in the same tweet, 2 hashtags *proukr*, 1 *prorus* and 2 hashtags *pax*), that tweet is deleted.

2.4.2 User Categorization. Once we categorized the tweets, we moved on to the categorization of the user themselves. At this stage we have introduced the categorization of the *dontcare* category through a 10% *indifference threshold.* If less than 10% of tweets belonging to the user are related to war (*prorus*, *proukr* or *pax*), the user is categorized as *dontcare*; otherwise, the simple majority is used to categorize the user as *prorus*, *proukr* or *pax*. In case of a tie, again, the user is discarded.

2.5 Data cleaning and network building

Our network has been created with:

- Nodes: twitter users who have posted at least one of the tweets included in our database;
- Links: interactions between users, for which we have chosen **retweets**. There exists a link between two users if one has retweeted a tweet of the other. Links are undirected.

To each node we have assigned the category of the user as a property in `networkx`. We deleted self loops as well as isolated nodes and got the following network:

- number of nodes = 11,197
- number of edges = 12,767
- number of components = 225

Among the components we had a giant component and several smaller ones having a size in a range of [2-39] nodes. We decided to work only on the **giant component**, therefore obtaining the following final network:

Figure 5: final composition of the giant component

Total number of nodes	10184
Total number of edges	11968

proukr	6922
pax	2165
nocare	416
prorus	681

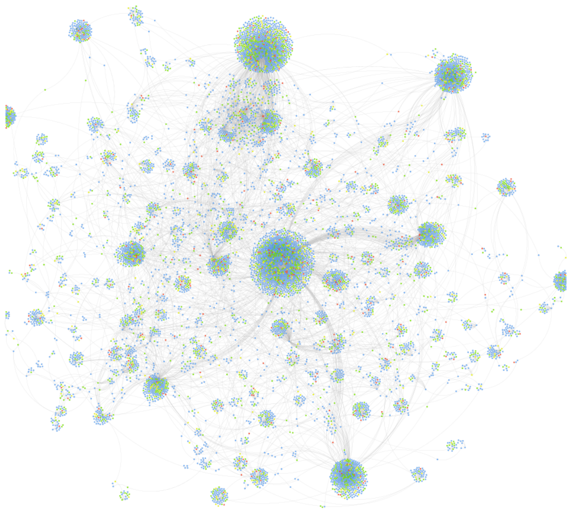
3 NETWORK CHARACTERIZATION

We will start with the analysis of the basic measures of our network while comparing them with those obtained on 3 synthethic nets: Erdos-Renyi, Barabasi-Albert and Watts-Strogatz model.
Following the previous data cleaning phase, for the purposes of our analysis we decided to focus only on the giant component present in our network,therefore our analyzed network consists of 10184 nodes and 11968 links, and it is an *undirected* and *unweighted* network.
We used `networkX` to model our network as a Real-World Network (also named "RW"). Then we used the graph visualizer *Gephi* to assign a different color to each node based on its category (see figure 6).
For an initial visualization of out network we used the *Force Atlas 2* layout which provided us an initial idea of the network structure (see figure 7).

Figure 6: percentage of nodes for each category

	proukr	67.97%
	pax	21.26%
	dontcare	6.69%
	prorus	4.09%

Figure 7: graph visualization using the ForceAtlas2 layout on Gephi



3.1 Synthetic graphs

To compare the results of our analysis on the Real World network created by us, we have built 3 different types of synthetic models:

- Erdos-Renyi model (ER): to make it similar to our network we set an ER with a number of nodes equal to 10184 (equal to the number of nodes in our network) and a value of p equal to 0.00025, thus obtaining an ER with a number of links similar to that of our network.
- Barabasi-Albert (BA): in this case we set a model that had 10184 nodes and a parameter m equal to 1, in order to obtain a synthetic model similar to our RW.
- Watts-Strogatz (WS): for this model we have defined the same number of nodes as our RW, parameter p equal to 0.05 and parameter k equal to 3 in order to obtain a model that had a number of links similar to ours.

In this way these models were useful to compare the results obtained for the measurements on our network with the results obtained on synthetic models.
Below is a table with the main metrics calculated on the different models:

measure	RW	ER	BA	WS
number of nodes	10184	10184	10184	10184
number of links	11968	11727	10183	10184
average degree	2.35	2.30	1.99	2
min degree	1	0	1	1
max degree	1956	10	171	5
diameter	18	26*	25	995*
average shortest path	4.97	10.69*	8.28	351.05*
density	0.0002308	0.0002261	0.0001963	0.0001964
average clustering coefficient	0.03828	0.0001309	0.0	0.0

*measures were computed on the largest connected component of the reference model

In the following phases we will analyze specifically the results obtained for the most interesting measurements on our RW network and the various similarities / differences with the results of the same metrics obtained on synthetic models.

3.2 Degree distribution analysis

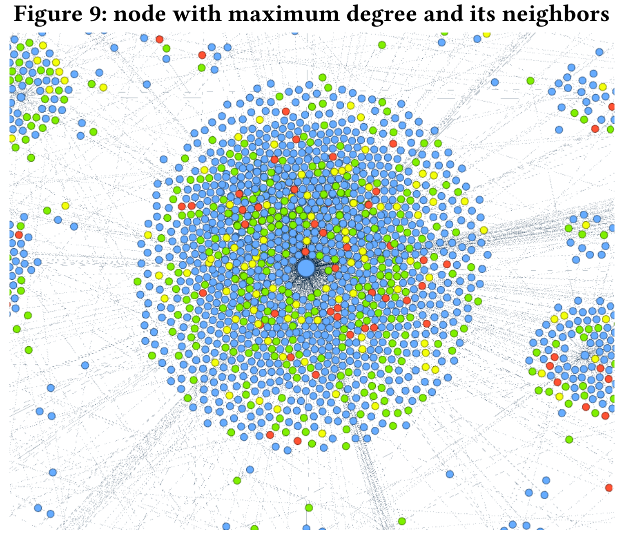
In the first phase of the analysis of our network we examined the basic measures focusing on the analysis of degree distribution.

Figure 8: average degree, minimun degree and maximum degree of the Real World network

measure	RW
average degree	2.35
min degree	1
max degree	1956

We have identified the node with the maximum degree (which corresponds to the value 1956 in the table), that is the node '27493883' (shown in figure 9) which is a *proukr* node with 1956 links divided between the different categories. In the following paragraphs we will deepen the analysis of other characteristics of this node.

Figure 9: node with maximum degree and its neighbors



3.2.1 Degree distribution. The degree distribution of our Real World network (figure 10), compared with that of the synthetic models analyzed, is similar to the distribution that we find in the synthetic Barabasi-Albert model (figure 11), while it presents significant differences with the distributions of the other models. In fact, the distribution in our model is in line with *power-law distribution* (similarly to what happens for the Barabasi-Albert model), this is due to the *preferential attachment* property. On the contrary, the distribution of our Real World Network turns out to be very different from the distribution of the Erdos-Renyi and Watts-Strogatz models that follow a *poissonian distribution.*

Figure 10: Real World network degree distribution

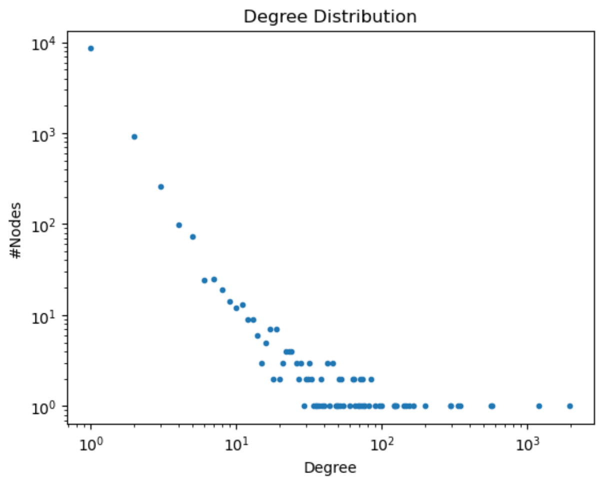
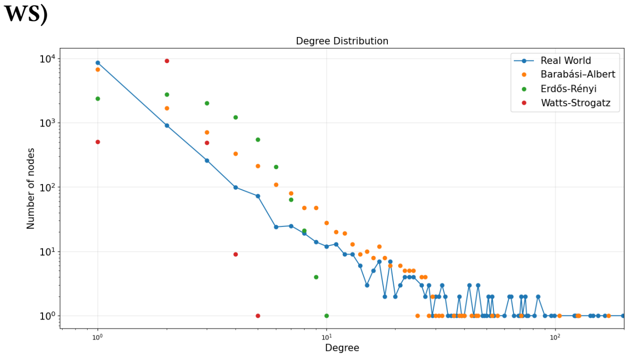


Figure 11: degree distributions comparison (RW, BA, ER, WS)



3.3 Connected components analysis and Path analysis

Continuing with the analysis of our Real World network, we moved on to the analysis of connected components. It emerged that in our current network, after the data cleaning phase seen above, there is a single giant connected component that coincides with the total network itself (N = 10184 number of nodes and L = 11968 number of links). Performing the same analysis on synthetic models, it emerged that two of the synthetic models (the ER model and the WS model) do not have a single connected component. In the following table we see what emerged from the analysis on the various models (figure 12):
(Remember that before the data cleaning phase, our network had larger dimensions: a number of nodes equal to 11197, number of links equal to 12767 and the network had 225 different components.)
Regarding the Path analysis our network has a diameter of 18 and an average shortest path of 4.97, a fairly modest value even compared to those calculated on synthetic models. The model that diverges most from our Real World in terms of path analysis

Figure 12: connected component analysis and largest connected component characteristics for each network

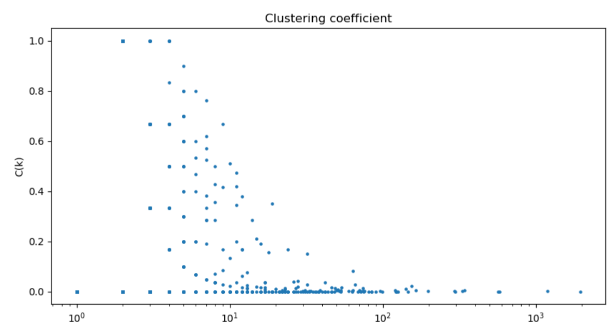
	number of connected components	largest connected component			
		number of nodes	number of links	diameter	average shortes path
RW	1	10184	11968	18	4.97
ER	1136	8835	11513	26	10.69
BA	1	10184	10183	25	8.28
WS	2	10112	10112	995	351.05

is certainly the Watts-Strogats model with a diameter of 995 and an average shortest path of about 351.05.

3.4 Clustering & Density

In our network the **Average clustering coeffcient** is 0.03828,tending to be low as it is a real-world network. By calculating the local clustering coefficient, we found several nodes that have a maximum local clustering coefficient value, namely equal to 1: most of them are pro-Ukraine nodes.
The distribution of the measures calculated for the clustering coefficent of our network is shown in the following graph (figure 13).

Figure 13: clustering coefficient distribution of Real World network



Turning now to the density analysis of our network compared with the measurements emerged in the synthetic models, we can say that all the models presented fairly low values for this metric. In the following table we can see the density values for each model analyzed:

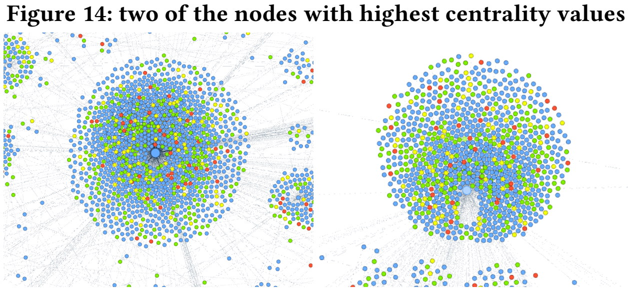
measure	RW	ER	BA	WS
density	0.0002308	0.0002261	0.0001963	0.0001964
number of links	11968	11727	10183	10184
potential number of links	71610528	68755401	51841653	51851836

This value for the density of our Real Network was predictable given the number of total links (compared to the number of potential links which is $N(N\text{-}1)/2$ for undirected graphs).

3.5 Centrality

For the centrality analysis we used different types of measures: Degree centrality, 3 types of Geometric centralities (Closeness centrality, Harmonic centrality and Betweenness centrality), Page Rank (which falls into the class of Connectivity-based centralities) and finally we conducted an analysis on Assortativity.
For the first measurements we investigated which nodes had the most significant values. In particular, we found 2 nodes that are more "central": both are nodes that represent pro-Ukraine users. We analysed one of these users semantically by looking at his Twitter profile and his Twitter activity. It turns out that this very central user (node '27493883' shown in Figure 14 on the left) is a not so popular American local politician. However, his high centrality is explained by his massive Twitter activity, especially retweeting others, and by the fact that he is heading a *follow and retweet circle*. He's inviting people who oppose a particular politician to retweet some of his tweets and to follow each other. Because of this he has a very high centrality although his real life popularity is not enormous.

Figure 14: two of the nodes with highest centrality values



3.6 Assortativity

To investigate the Homophily that characterizes our network, we chose the assortativity measure which can be considered as a quantitative measure of the correlation between nodes in the network and in particular we chose the Degree Assortativity (r). In our network this value is r= -0.2379, therefore our network is slightly disassortive.
The correlation between nodes degrees can be observed in Figure 15.

4 TASK: STATIC COMMUNITY DISCOVERY

In trying to discover communities we employed the *Label Propagation*, *Leiden* and *Louvain* methods (non-overlapping) and *Angel* and *K-Clique* (overlapping). In Figure 16 we report the average and standard deviation of the evaluation metrics.
According to **Newman-Girvan modularity** and **conductance** Louvain, Leiden and Label Propagation performed better than other in both the metrics.
In regards to external evaluation, for the **normalised F1 score** again were best Louvain and Leiden with scores of 0.23 while k_clique and angel reached 0.14 and Label Propagation scored less than 0.01. The **Mutual Information (MI) score normalized** - which can only be applied to non-overlapping partitionings - highlighted as best again the Louvain and Leiden with

Figure 15: assortativity distribution of Real World network

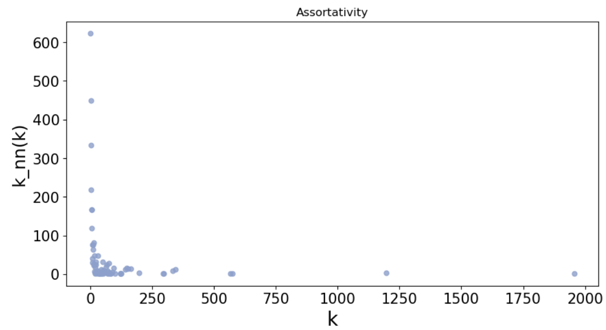
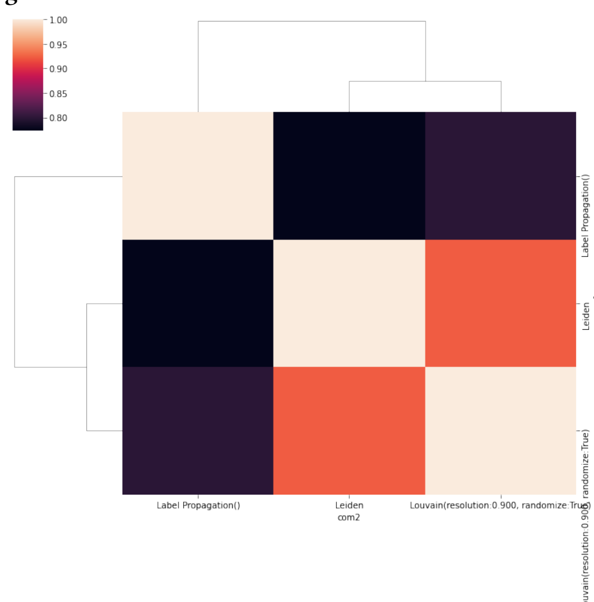


Figure 16: summary of parameters, results and performance of each community discovery algorithm

	freq	size	average node degree	modularity	conductance
Louvain	46	[221.4 , 343.0]	[2.0294 , 0.1785]	0.84100	[0.08671 , 0.05184]
Leiden	50	[203.7 , 281.1]	[2.0031 , 0.2515]	0.84457	[0.08624 , 0.05011]
Label_prop	360	[28.3 , 138.2]	[1.5424 , 0.3645]	0.79333	[0.25272 , 0.15463]
K_clique	39	[16.1 , 46.9]	[2.5334 , 0.9416]	0.01845	[0.88608 , 0.13878]
Angel	10	[52.8 ,106.6]	[3.1043 , 1.5313]	0.00185	[0.75228 , 0.1891]

scores of 0.93 and Label Propagation as worst. The matrix below shows this similarity.

Figure 17: heatmap of different Community Discovery algorithms



In reference to the formation of communities, our team initially attempted to identify communities in which users of only one category were present, but this was not found to be the case. Rather, the majority of communities we found presented an equal distribution of the four types of users we had identified. In only a few small communities were there at most two types of users. Thus, based on our categorization, these communities do not appear to represent communities of opinion. Our subsequent investigation focused on identifying patterns of geolocation in the formation of communities through the localization of each individual tweet and its respective user. While we did not find any common geographical location in the larger communities, we did observe a geographical pattern in some of the smaller communities, such as retweet communities of Italian users or users between Russia and Ukraine. It is important to note a limitation in our investigation regarding the presence of null values in the geolocations of many tweets. Additionally, some users utilized fictitious or overly general locations (e.g. "World"), which precluded their inclusion in our analysis.

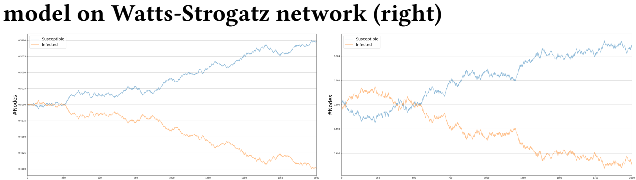
5 TASK: OPINION DYNAMICS

In this phase of our analysis we focused on how the opinions/positions of users in our network can change following the active diffusion models of Opinion Dynamics. In particular, we implemented *Voter*, *Majority rule* and *Sznajd* models. We applied these diffusion models to our network and then we compared them with the results obtained on the Barabasi-Albert and Watts-Strogatz synthetic networks. With the application of these models we analyze the trend of the diffusion of an opinion, inserted within a network, which can take only 2 possible values: +1 (positive opinion) or -1 (negative opinion), which can be displayed in our graphs respectively with the colors orange ("infected") and blue ("susceptible"). Our analysis for all 3 models starts from the hypothesis of an initial balancing situation between "infected" and "susceptible" and a number of iterations equal to 2000.

5.1 Voter model

The Voter model applied to our network shows a divergent trend between the two opinions, already starting from the 250th iteration. In this case, nodes with negative opinion tend to increase with increasing iterations, and the number of nodes with positive opinion tends to decrease more and more. This result can be compared to the trend of the same model applied to a Watts-Strogatz synthetic network with similar characteristics: in fact, even in this case, although only after the 500th iteration, the model shows this type of divergent trend between the two opinions.

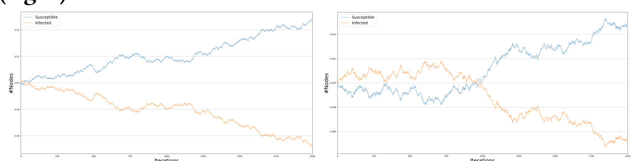
Figure 18: Voter model on Real World network (left), Voter model on Watts-Strogatz network (right)



5.2 Majority rule model

This diffusion model shows a very rapid reversal of trend: in fact, in the first iterations, the "positive" nodes are superior to the "negative" ones, while starting from the 30th iteration, we see the positive ones decrease and the negative ones increase.
We find a similar trend on the Barabasi-Albert model: here too, nodes with positive opinions are greater than the negative ones, but unlike what we observed in our network, the trend reversal occurs much later, starting from the 1000th iteration.

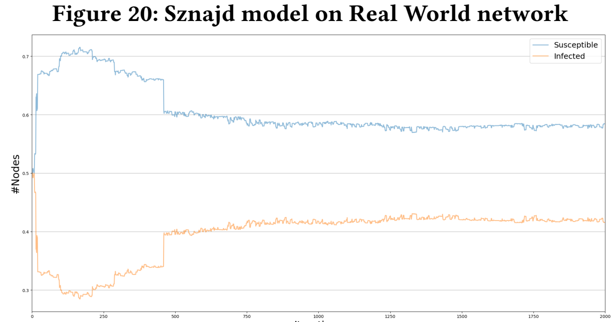
Figure 19: Majority Rule model on Real world network (left), Majority Rule model on Barabasi-Albert network (right)



5.3 Sznajd model

In our network the trend of the two opinions tends to stabilize around the 750th iteration with a prevalence of the negative opinion (Figure 20).

Figure 20: Sznajd model on Real World network



In synthetic models, however, opinions tend to diverge and one of the two tends to disappear, with a clear dominance of negative opinion observable especially on Barabasi-Albert (Figure 21).
In Watts-Strogatz, the dominance of negative opinion undergoes a reversal around the 250th iteration and then reverses again around the 500th iteration, then finally asserts itself as the dominant opinion (Figure 22).

6 OPEN QUESTION

We have investigated how users of different categories change their opinions two months after the beginning of the war. To achieve this, Markov Chains[1] were employed to calculate the probability of users maintaining the same opinion versus changing to one of the other three categories.

[1]https://github.com/NaysanSaran/markov-chain

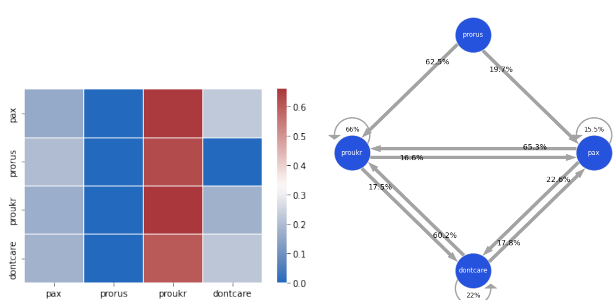
Figure 21: Sznajd model on Barabasi-Albert network

Figure 22: Sznajd model on Watts-Strogatz network

6.1 Behaviour observed in $t_0 \rightarrow t_2$

Two different time intervals were defined:
- t_0: From 15/02 to 15/03 we observe the opinions in the month straddling the date of the outbreak of the war. This is the time period spent so far in this work and the network generated previously.
- t_2: From 15/04 to 15/05 we observe opinions about two months after the start of the war. We observe the same users we observed in t_0 and re-categorize them in t_2 according to their opinions in this later time period. Categorization is done in the same way as for t_0, as already reported in chapter 2.

Figure 23: transition matrix heatmap (left) and Markov chain $t_0 \rightarrow t_2$ (right)



It was observed that two months after the outbreak of war:

- there is a 66% probability of those supporting Ukraine maintaining their opinion;
- the probability that users not being still actively interested in Twitter news about the war is 22%;
- the probability of those who want peace continuing to do so regardless of the outcome is 15.5%;
- the probability that a supporter of Russia will continue to favor its victory is equal to 0.

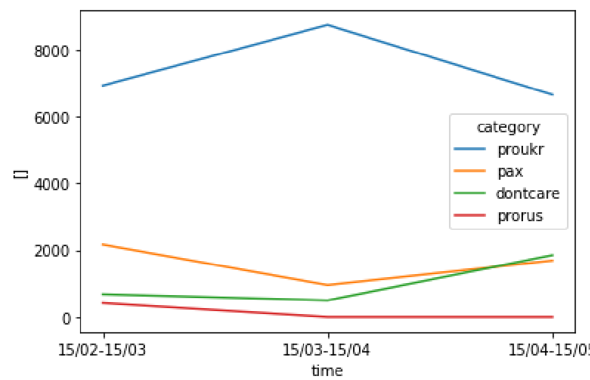
It is noteworthy that there is a high probability that the other three opinion categories converge to support Ukraine. Moreover, users who were categorized as pro-Russian were found to side with Ukraine or become peaceful by the end of the study.

6.2 Insights gained by adding t_1

Finally, an additional timestamp was used to understand the changes between t_0 and t_2:
- t_1 : From 15/03 to 15/04, a time interval captured to understand what happens between the beginning of the war (t_0) and two months later (t_2)

The line chart in Figure 24 shows how the total sum of all opinions for each category changes from t_0 to t_2.

Figure 24: line chart of change categories from t_0 to t_2



Initially, there was a notable rise in the number of individuals expressing support for Ukraine. However, this trend experienced a decline in the subsequent month, accompanied by an increase in individuals who expressed a neutral stance or indifference (*dontcare*) and those advocating for peace (*pax*). It is worth noting that the proportion of peace supporters, although showing an increase, remained lower than the level observed at the starting point (t_0). Contrarily, the aforementioned observation does not hold true for the *dontcare* category, as it experienced a substantial surge in numbers.

It is also noted that pro-Russian users were only 416 initially and disappeared completely in the subsequent timestamps due to various reasons:

- Twitter deleted tweets in favor of Russia, which affects the categorization, based on hashtags (ansa.it[2], voxukraine.org[3]);

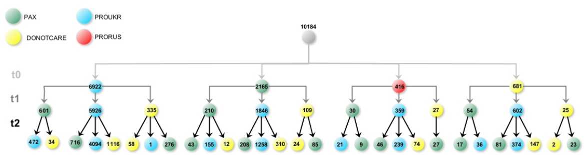
[2]https://www.ansa.it/sito/notizie/tecnologia/hitech/2022/03/17/ucraina-twitter-rimuove-50-mila-contenuti-fake-sulla-guerra_c02d3ff1-16d9-4650-87c4-21f0fe809d5e.html

[3]https://voxukraine.org/longreads/twitter-database/index-en.html

- Russians tend to use other social platforms, such as VKontakte, as Russia has blocked access to platforms such as Twitter and Facebook (theguardian.com[4])
- Additionally, nine days preceding the war were included in the analysis where users could have been in an initial phase of taking a position.

Through a tree graph, it is possible to observe how user categorization varies from one timestamp to another, labeling each category with a different colored node.

Figure 25: tree representation of change categories from t_0 to t_2



It can be noted that approximately 85% of the users in t_0 will be *proukr* category in t_1, while the remaining seem to support peace more. However, this is not true for the users who showed support for Russia in t_0, as half of them will side with peace while the other half will lose interest in the topic.
As also observed in the Markov chain, the majority (4094) of proukr tend to remain in the same opinion in all considered timestamps. While it is very difficult for a user belonging to *prorus*, *pax*, and *dontcare* to tend to stay in the same category.

6.3 Conclusions and limitations

This project shows that the prevailing opinion on Twitter during the initial months of the conflict is *proukr*, as Twitter users support Ukraine winning the war. One of the relevant results is that the more time passes since the outbreak of war, the more users tend to tweet less about war-related topics. Only 2 users remained in the dontcare category in all time steps, thus demonstrating how, in the face of such a strong issue, it is almost impossible not to be involved on Twitter.
One limitation of this project was categorizing pro-Russia tweets for which, as previously reported, the adopted classification method was not very efficient. Another limitation may be due to RAM, as it was not possible to collect too many tweets in creating the hashtag lists, having to refer to a limit of 80,000 tweets. Therefore, some hashtags may not have been correctly categorized, negatively influencing user categorization. However, based on what has been demonstrated, it is considered that the proposed method can be considered efficient and can be further refined.

7 REFERENCES

- `twarc2`: Documenting the Now. (2021). twarc2: A Python library for accessing Twitter data. Retrieved from github.com/DocNow/twarc;
- `networkX`: Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart, "Exploring network structure, dynamics, and function using NetworkX", in Proceedings of the 7th Python in Science Conference (SciPy2008), Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), pp. 11–15, Aug 2008, https://networkx.org
- `CDlib`: Rossetti G., Milli L., and Cazabet R. 2019. CDLIB: a python library to extract, compare and evaluate communities from complex networks, https://doi.org/10.1007/S41109-019-0165-9. Applied network science 4 (2019);
- `NDlib`: G. Rossetti, L. Milli, S. Rinzivillo, A. Sirbu, D. Pedreschi, F. Giannotti. NDlib: a Python Library to Model and Analyze Diffusion Processes Over Complex Networks. Journal of Data Science and Analytics. 2017. DOI:0.1007/s41060-017-0086-6 (pre-print available on arXiv); G. Rossetti, L. Milli, S. Rinzivillo, A. Sirbu, D. Pedreschi, F. Giannotti. "NDlib: Studying Network Diffusion Dynamics", IEEE International Conference on Data Science and Advanced Analytics, DSAA. 2017;
- `Gephi`: https://gephi.org/.

[4]https://www.theguardian.com/world/2022/mar/04/russia-completely-blocks-access-to-facebook-and-twitter