Network Science - Interdisciplinary Project

HASHTAG ACTIVISM AND DISCRIMINATION IN GENDER CRIMES

CRIMES AND SOCIAL SOCIAL RESPONSE FOR MALES, FEMALES AND TRANSGENDERS IN PAKISTAN AND EGYPT

Mustafa Algun - 2049537

Selen Arslan -2004968

Umutcan Berk Hasret - 2041391

Dyutideepta Banerjee - 2005645

Noha Shohda - 2005638

Eieshah Mubasher - 2046859

Matyame Mouida - 2054681

Contents

Introduction

Different cultures across the world have taken different attitudes to gender diversity at different points in history. Unfortunately, even today, we still often hear about violence based on gender or the sexual orientation of each individual. We often hear about femicide, transgender people being attacked, or lesbian and gay people being beaten up just because they show their love in public. In many countries, required regulations and law enforcement are applied by judiciary authorities in remuneration for this type of violence acts. Nevertheless, this is not the case in Arab countries. Sharia law recognizes only two identities/genders: male and female, where dominance is usually asserted by men.

Starting from this point, gender-based violence by focusing on two Islamic countries, Pakistan and Egypt, is reflected in this paper. Pakistan and Egypt are two completely different nations that, however, base their justice on Sharia, the Islamic law. Sharia acts as a code for living that all Muslims should adhere to, including prayers, fasting, and donations to poor people. It aims to help Muslims understand how they should lead every aspect of their lives according to God's wishes. Sharia, therefore, aims to lay the foundations for leading a life in the right way, but very often these rules are misunderstood and anyone who breaks these rules is severely punished. Obviously, each country adapts to social progression, and this is what distinguishes conservative from progressive Muslim countries. Considering the fact that these are typically patriarchal societies, there is no lack of struggles by feminists and minority activists, and in many cases, it is difficult to remain silent in the face of crime and violence against these minorities.

1. Network Analysis

Many researchers used network analysis to deal with text corpus retrieved on social media. The basic assumption is that better explanations of social phenomena are yielded by analysis of the relations among entities, according to this, we tried to create different communities based on gender, starting with the most discriminated one in the two countries: the transgender community.
The main goal was to build a **Semantic Network**, then graphically represent the relationships between hashtags and words used, based on their co-occurrence, to express one's beliefs and feelings about a given topic. We analyzed the database according to two main techniques:

- Topic analysis
- Hashtag analysis

In this research, Twitter, a social media platform, is used as a means of information. In specific, the Sandbox tier of Twitter API was utilized throughout the whole data retrieval process. As a programming language, Python was used in the project. The general workflow was explained in steps below:

- Making queries and retrieving a predefined number of tweets according to the selected hashtags
- Converting obtained JSON files into Pandas data frames for future modifications.
- Filtering out unused parameters after parsing and creating a simpler table
- Applying necessary alterations to the data so that it is adequately articulate.
- Extracting words and hashtags from the data.
- Designing three different networks with only extracted words, only extracted hashtags, and the overall network with the complete data.
- Exporting node & edge lists to be used on Gephi.
- Adjusting node sizes according to PageRank and node colors according to modularity/communities.
- After importing node & edge lists at hand, choosing the right layout for that specific network.
- Calculating various network metrics and exporting them in CSV format.
- Tailoring the network design to our liking, exporting the final network picture.
- PDF and CDDF and gamma calculations and creating related plots based on the exported metrics

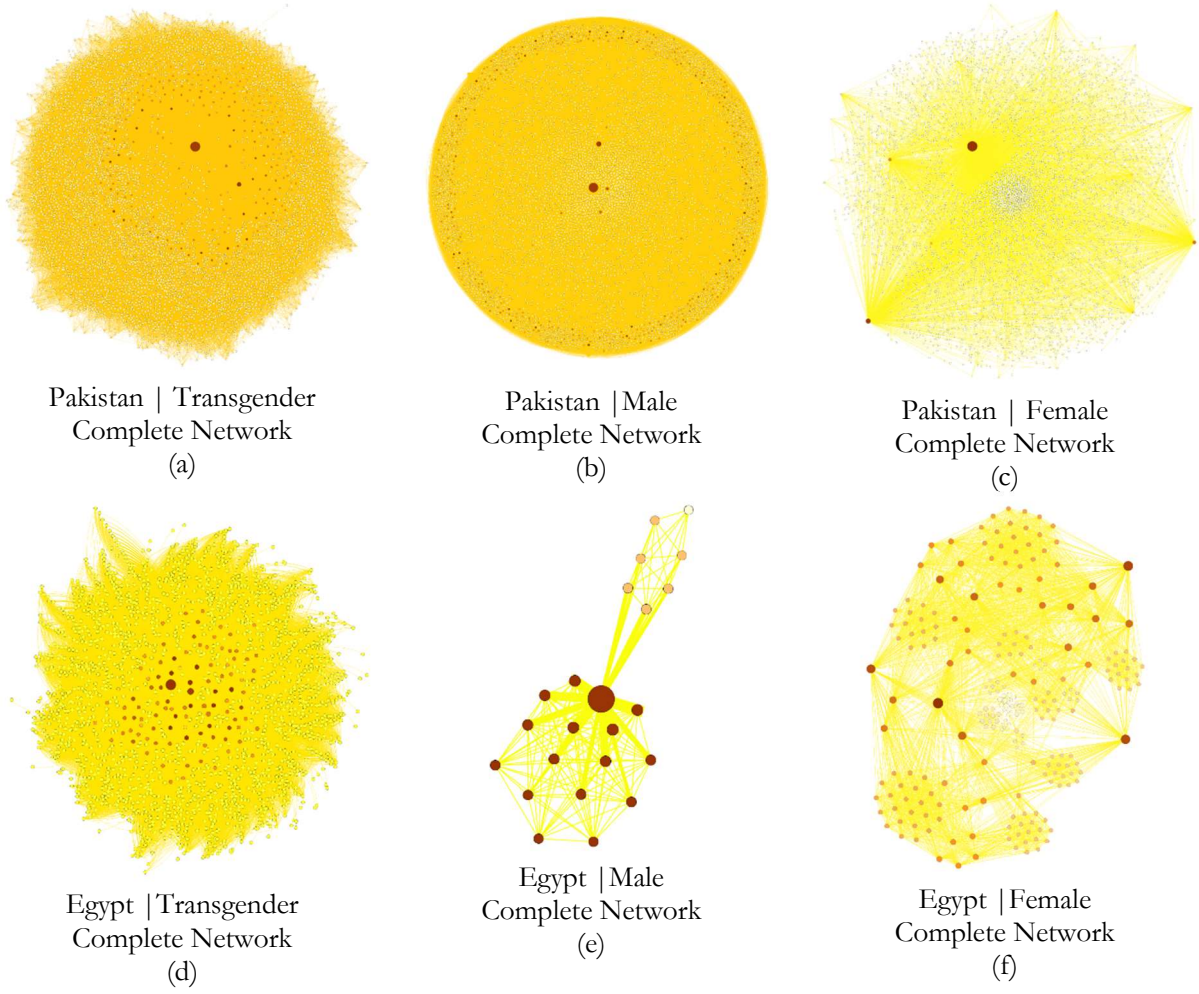


Figure 1: Complete networks designed in Gephi with Force Atlas layout. Dark nodes represent high degree nodes. Bigger nodes are ranked higher in PageRank. Hubs are pushed at the periphery and authorities are more central

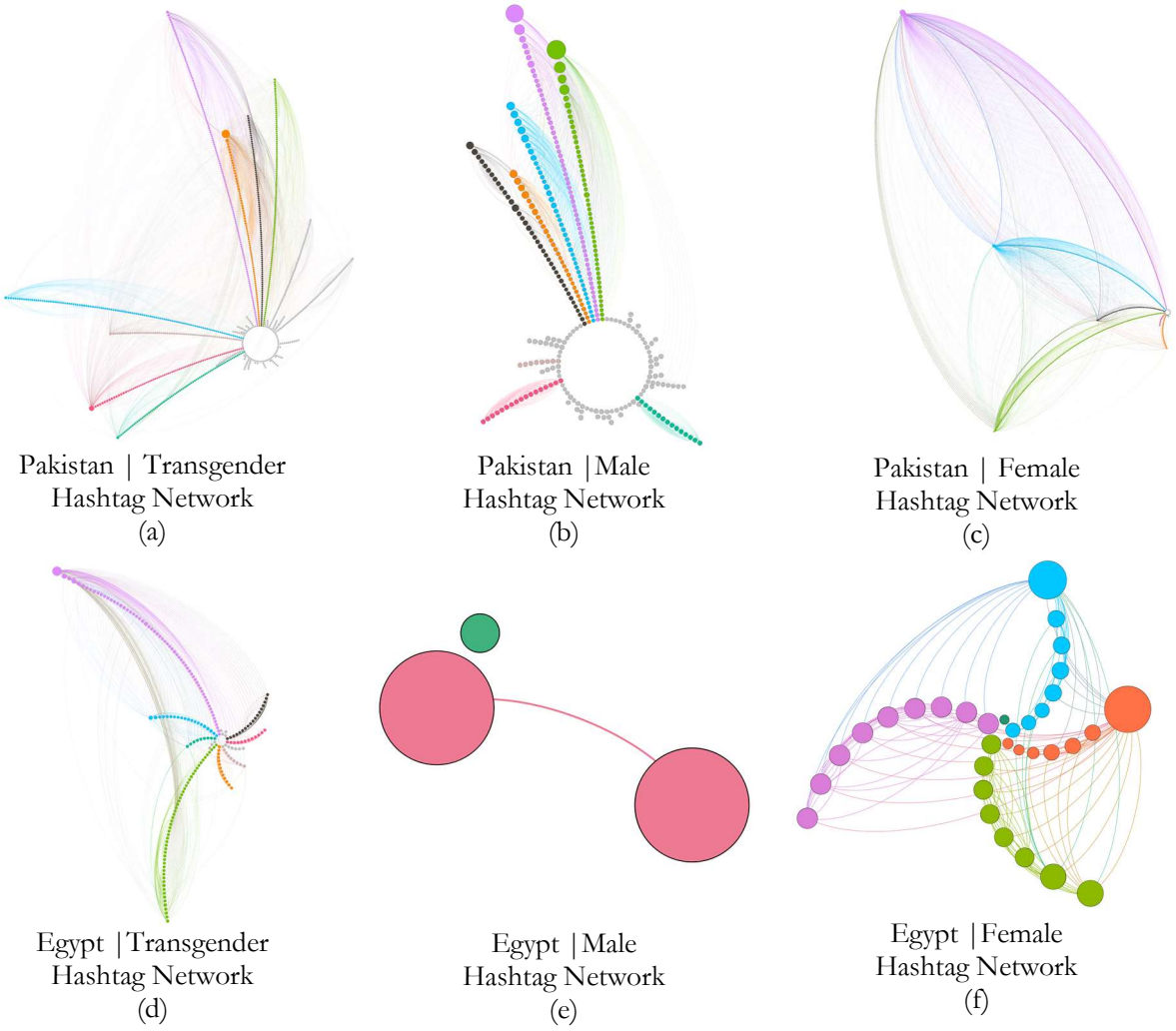


Figure 2: Hashtag networks designed in Gephi with Radial Axis layout. It allows studying homophily visually for small networks such as these ones.

For each gender role we have, we gathered several hashtags. In addition, we divided the study into two main parts: Pakistan and Egypt. Thus, the overall picture of the research can be seen below in Figure 3.

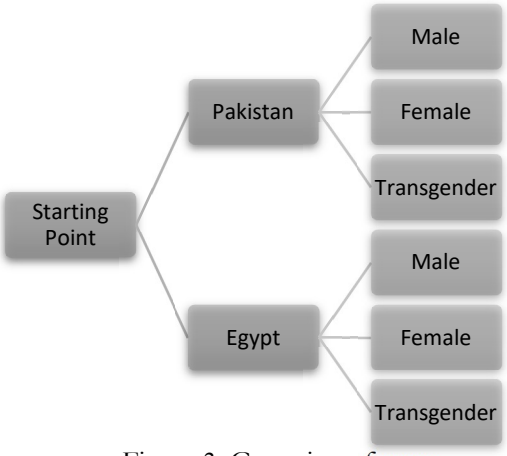


Figure 3: Grouping of cases

By analysing two culturally different countries, with different slang and two different languages, we found that the best way to identify the communities in our network was by using hashtags, cause the hashtags studied are all in English.

Local language-based hashtags could not be studied. As the initial idea was to identify for each country, which community suffers the most from social crimes and which community is most ignored by governments and society itself, and then compare the communities of the two countries considered for the research; gathering the information through hashtags was the best solution. Furthermore, not all cases get hashtags, so there are a lot more cases out there than there are hashtags. Social media algorithms can play a huge role in which sort of news gets the most amount of coverage. A parallel event more trending but less severe or completely irrelevant to gendered crimes could be isolating a hashtag. However, the topic network was not ignored, as it allowed us to create a connection between nodes by identifying who positively and actively support the social struggle against gender-based brutality of individuals.

To retrieve data were used major newspapers and several blogs were also used:

- BBC
- Aljazeera
- Internazionale
- The Diplomate
- Dawn (Pakistan's Newspaper)
- DailyPakistan.com
- Open Democracy
- Egypt Today
- Egypt Independent
- Global Fund for Women (Pakistani blog)
- Human Rights Watch (Egyptian blog)

For each case,3k tweets have been gathered.

1.1 Pakistan Cases

The starting point of our research was the recent rise in crimes against transgender people in Pakistan.

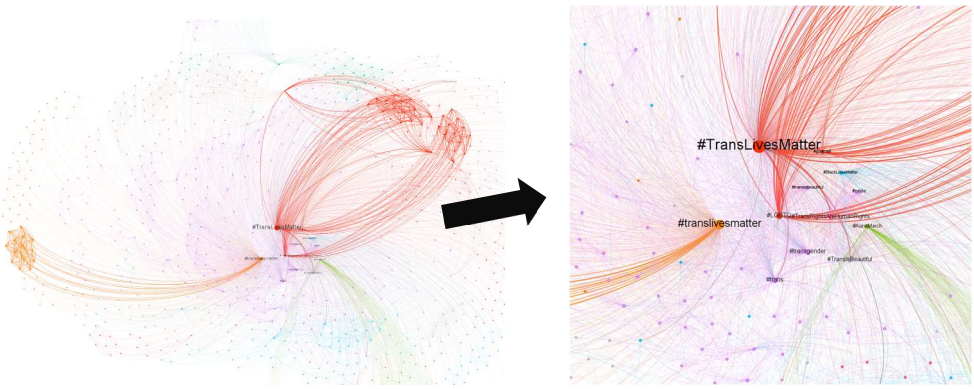
« In November 2021 Nayyab Ali, a prominent trans activist, was attacked in her Islamabad home by two men wielding knives. She was held hostage for three hours, during which time she was beaten and robbed ».

This news shocked the whole of Pakistan, becoming redundant even on social media, where the hashtag #NayyabAli went viral. However, this type of violence is not unfamiliar to the transgender community in Pakistan. Even though Pakistan has adopted laws recognizing gender reassignment, transgender people and other marginalized minorities suffer discrimination and violence in many spheres of their lives. Human rights violations and discrimination based on gender identity are still prevalent and mount a big challenge for Pakistan. The transgender community and other excluded minorities face stigma, discrimination, and violence much more than non-marginalized groups. Transgender people, and transgender women, in particular, face harassment, mistreatment, and exclusion from society, from the public health care system, education system, employment, and other institutions of government. They face different forms of abuse, ranging from exclusion from society to brutal murder. That means that the country has systematically failed to protect these minorities, but what is society's reaction to these brutalities? What is the government's response to this violence?

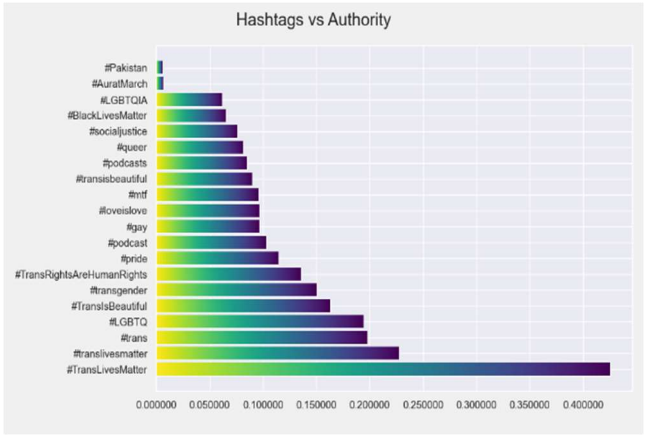
So, we wanted to investigate the social response to discrimination and violence based on gender, the main social network used for our research was Twitter, one of the most used social in Arab countries to raise the voice all over the world.
The first step was to find all news about crimes and brutality suffered by minorities in Pakistan; the keywords used in our research were: murder, transgender, feminicide, assault, harassment on transgender, transgender community, Khawaja Sira community (also known as the hijra, is a broad group that encompasses trans and intersex people, as well

as eunuchs), violence on trans community, shooting of trans people, rape on trans people. All these keywords help us to identify the first community on which we wanted to focus our attention. Thanks to the information collected by blogs, articles in newspapers we were able to identify the main hashtags #TransLivesMetter, #JusticeForGulPanra, #Auratazadimarch, #TransIsBeautiful, #LGBTQ.

These hashtags extrapolated from Twitter showed a network of nodes characterized by **homophily**, so an association of individuals that support the same idea: the transgender community must be protected and the government has to act to punish the cruelty against the community. Following this path, we then identified all the hashtags in posts with positive language that support the community and therefore condemn the injustices suffered by transgender people. The homophily present in that network led to a **collective action** identified by the hashtag #Auratazadimarch, organized by trans activists that take place every year, to these collective actions we don't have evidence of repressions and that demonstrate that authorities allow to people protesting peacefully to express their feelings and ideas, thanks to the laws that recognize those few rights to transgender people.



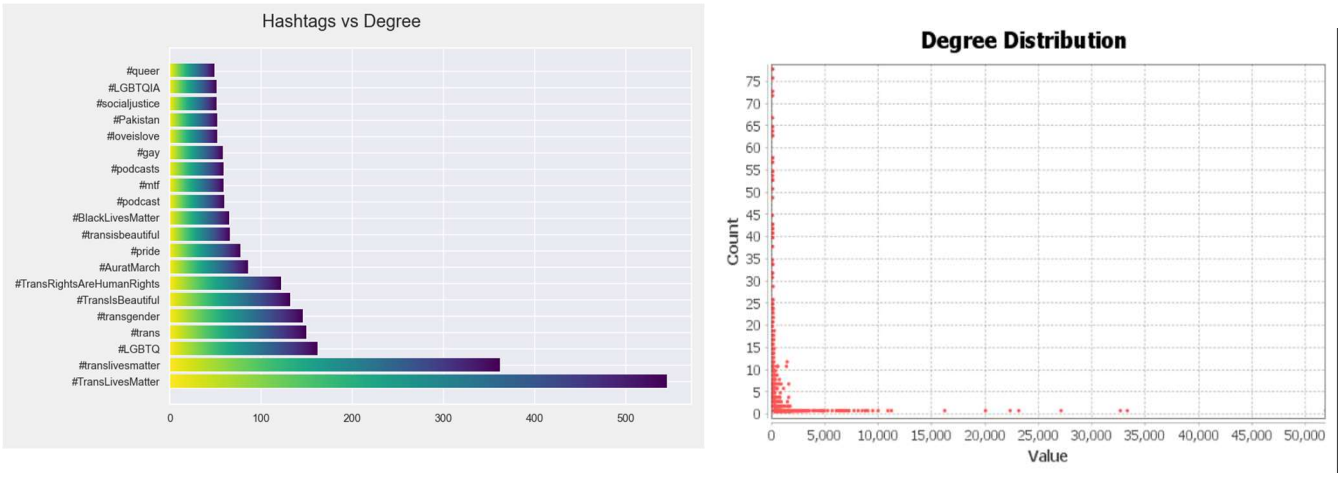
(a)



(b)

Figure 4: Network design with the hashtag of greatest degree: #TransLivesMatter

As Figure 4 demonstrate, the most used hashtag is #TransLivesMatter, followed by #LGBTQ, so users, in order to feel part of a group that supports transgender issue, will identify themselves with those hashtags, which are nodes with the highest degree value (demonstrated in Figure 5)



(a) (b)

Figure 5: Degree distribution in the bar graph and in scatter plot

1.1.1 Pakistan Female Cases

In order to make sure the data sets are equal and comparable for both Egypt and Pakistan, the strategy used was to employ three cases from each gender for each country. This means that three cases of men from Pakistan, three of women, and three of transgenders. The timespan of these cases are the latest possible and the qualifier is to use hashtags/cases that gathered a significant amount of coverage or matched across genders in the severity of the crime. This was our way to being sure the severity of the crime did not actually trump the results as the more heinous crimes are most likely to get the most amount of attention, thereby taking away from the difference of gender being our main cause of concern.

Date	Case	Hashtags
2021	Noor Mukaddam Murder	#zahirjaffer #justicefornoor #noormukadam #asmatadamjee #StandwithNoor #JusticeForNoormukadam #spreadnoor #NoorMuqaddam
2020	Motorway Gang-Rape	#motorwayincident #CCPOLahore #RemoveCCPOLahore #motorwaycase
2020	Qandeel Baloch Murder	#qandeelbaloch #QandeelBaloch #JusticeForQandeelBaloch

Table 1: Hashtags

With time, awareness, and continuous femicide activism in Pakistan, more cases came to light than in previous years. The amount of attention devoted to cases also picked up incrementally. This is the reason why 2 out of the 3 cases mentioned are from 2021. This was why these cases are more noteworthy as they reflect the recent most sentiment in the audience. The one case that has been selected from 2016 is still relevant to current day as the trial for it was still under process and the issue remains unresolved. The 2022 acquittal of the murdered in the case of Qandeel Baloch has made it still a topic being discussed, thereby still reflecting the current most audience sentiment.

In terms of intensity, the Noor Mukaddam murder case involved a woman being beheaded by a man belonging to a very rich family. The victim was kidnapped, abused, and then eventually killed. The victim as well as the accused belong to extremely affluent families of Pakistan with Noor being the daughter of an ex-foreign ambassador and the accused belonging to one of the richest businessmen families. The below case represents diversity in two different areas:

1. Class
2. Type of crime

The second selected hashtag is from the case of a middle-class woman who was driving late at night on the highway, then her car ran out of fuel and she was stranded. While she called for help, men from nearby settlements gathered and raped her in front of her family. While Noor's case represents the elite and category of crime as murder, the motorway case involves middle-class women with the crime category being rape.

The third case is of Qandeel Baloch, a social media influencer belonging to an extremely poor and backward area of Pakistan. Her brother killed her but what makes her murder different from that of Noor's is that it was an honor killing. The term describes a culture in the country where if a woman is considered notorious and bad for the reputation of her family, she is killed to protect the honour of the household. These cases are rampant across the country and their representation in the compilation of the cases we study is paramount.

In summary, these three hashtags are meaningfully picked to represent each financial/social class and popular most categories of kinds of crimes that take place against females.

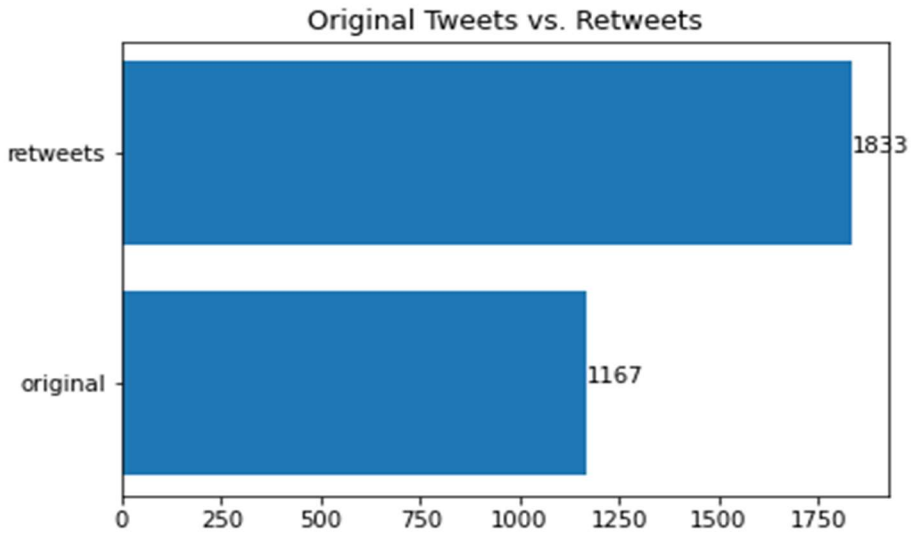


Figure 6: Original Tweets vs Retweets in Female Cases

From the data set analyzed, we see in Figure 6 that, on average, for all three cases combined, the number of tweets was 1167, whereas the number of retweets was 1833. The actual number could have been much higher if the query number hadn't been limited due to the tier used. Therefore, it's been tried to give a representative picture of the study with the current facilities (1000 queries per case and 3000 in total). However, it's observed that there was a significant amount of initiative in people starting the conversation around this hashtag and a great deal of support in sharing the sentiment of carrying it forward as a legitimate point of concern.

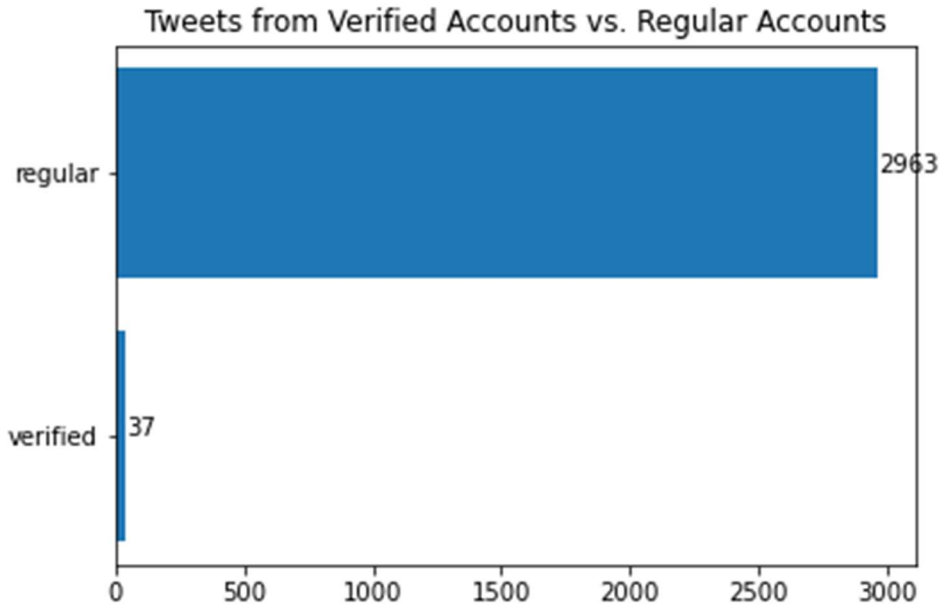


Figure 7: Tweets from Verified vs Regular Accounts in Male Cases

Considering Figure 7, for us to gauge how many of the influencer accounts or people who hold credibility in the society were there to tweet for gender crimes against women, the number is alarmingly low. It's been observed that very few verified accounts took part in the conversation. But on the brighter side, the nature of the cases drew out the common people as an issue worth talking about, regardless of whether social celebrities lead it. This can possibly suggest that people truly empathized with the occurrence rather than following a bandwagon of thought led by key personnel they may follow.

1.1.2 Pakistan Male Cases

Date	Case	Hashtags
2018 - Now	Ali Zafar Defamation	#justiceforalizafar #AliZafarisinnocent #boycottmeesha #banmeeshashafi #istandwithAliZafar
2021	Usman Mukhtar	#StayStrongUsmanMukhtar
2019 - 2020	Hameed Haroon	#Dawn #Jami #HaroonRapist #HameedHaroon

Table 2: Hashtags

Finding cases of gender crime against men was in itself an ordeal because the privilege lies with this particular gender, turning them into perpetrators rather than victims. In spaces where they do actually face criminal activity, they are much more likely to never report it as it would be considered shameful for a man to have been emasculated. The recent rise in the #MeToo movement initiated to uncover cases of abuse by men has been controversial in many instances, acting as a tool for defamation rather than justice. Although this is quite a grey area until an actual court ruling is out, it can be considered a gender crime against men.

The first case chosen is that of the absolute upper-class, high-end celebrity Ali Zafar who has been accused of harassment by multiple women of workplace harassment. He has been arguing through a very powerful legal team and media support group, and this is probably a ploy to harm his career and stems from jealousy rather than the actual crime. The ongoing court proceedings have kept this hashtag alive for years and are fueled by aggression and hate towards the women who reported him, labeling them as attention seekers.

The second case is that of Usman Mukhtar, a mid-tier actor in the Pakistani drama industry who claimed that the spat initiated after he worked on this colleague's music video as a director, but due to creative differences, he didn't end up taking credit for the job, which eventually led to the said, 'harassment'. Mukhtar also mentioned he has substantial proof to back up his claims, where he has been bullied and blackmailed with threats of being 'exposed' as a male artist who made the environment for a female colleague 'uncomfortable'. The difference between Ali Zafar and this case is that in the former, there is evidence against the male as the perpetrator and the nature is defensive aggressive; whereas the latter is not as affluent, has no proof against the victim and there's actual evidence of the woman harassing him over the years into owning a music video that he creatively did not feel aligned with. The nature of this case is entirely defensive and sympathetic on part of Usman.

The third case and set of hashtags are of a journalist who was subjected to the casting couch rituals of an abuse-dominated media industry. Jamshed Mehmood was raped by the CEO of a major news corporation in the country. He is a rare and reporting male victim of the #MeToo movement.

All three hashtags were carefully selected to portray a diversity of crimes against men but it is easy to notice that all belong to the media industry. It is possible that the shame associated with being a male victim has made it impossible for men from lower classes of society to come forward and it is only men who have a strong social voice are the ones who are able to speak up. The media industry is more woke and accepting of male victimization to see such crimes as criminal, rather than shameful emasculation.

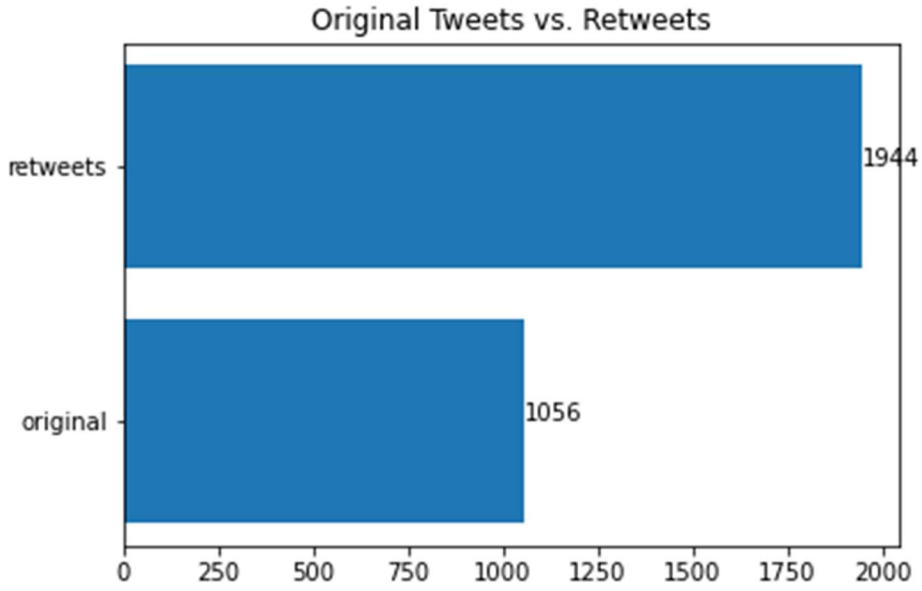


Figure 8: Original Tweets vs Retweets in Male Cases

Since we can't qualify the number of tweets per case, it is impossible to argue that more or less attention has been paid to male victims than females. But in terms of the initiative taken to start these conversations and its ratio to those who reshared or jumped on the bandwagon, the ratio is quite similar to that of the female cases.

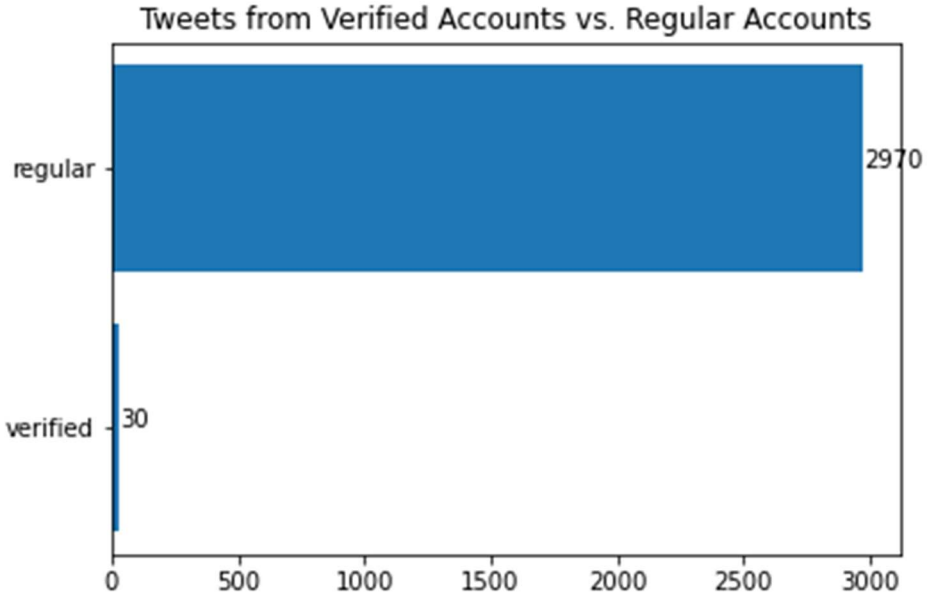


Figure 9: Tweets from Verified vs Regular Accounts in Male Cases

Figure 9 is also surprisingly similar to that of the female output, showing that the same number of strong social media voices began these conversations as they did for women. However, if we look into how all three cases of the male hashtags were from the media industry, it is fair to assume that male cases should have had a higher number of verified accounts as peers of the victims are more likely to have been more engaged in the dispersal of content. But this not turning out to be the case shows that beyond influence and industry, the male gender is treated in the same manner as are female cases.

1.1.3 Pakistan Transgender Cases

Date	Case	Hashtags
2020	Gulpanra murder	#gulpanra #JusticeforGulPanra #StopGenocideOfTransgenders #TransLivesMatter
2021	Bijlee, Sherly, Toffi murder	#JusticeForBijlee #JusticeForSherly #JusticeForToffi
2021	Muhammad Moiz mob attack	#EndTransViolence #BeelaCrisis #AuratMarch #Khwajasira

Table 3: Hashtags

Gulpanra was a trans rights activist in the city of Peshawar who was shot dead as an act of suppression by her haters. She used to be a vocal advocate of her gender minority's rights, a mission that took her life as she was murdered into silence.

Bijlee, Sherly, and Toffi are three cases that came to light all at once, demanding attention as a collective surge in trans violence. Bijlee, a transgender woman who begged in the streets to make ends meet, was strangled to death. Toffee, a 19-year-old transgender woman, was refused treatment at the Civil Hospital upon being tested HIV positive and, hence, died on the spot. Sherly, a transgender woman who was so heavily discriminated against in healthcare spaces that she

was terrified to go and seek treatment when she needed it, passed away at the hands of our healthcare system's transphobia.

This set of cases identifies an array of disparities and different forms of abuse the transgender community faces. However, they were not famous activists or well-known people, for which the news of their demise did not trigger as big of a conversation.

The third hashtag was triggered when Mohammad Moiz, another trans rights activist was attacked by a gang well known for hunting down transgenders. The gang is referred to as Beelas and they are famous for kidnapping transgenders, leading up to abuse, harassment, rape, and even death. In the case of Moiz, the difference lied in the outcome, as Moiz was able to survive and create more awareness against the gang.

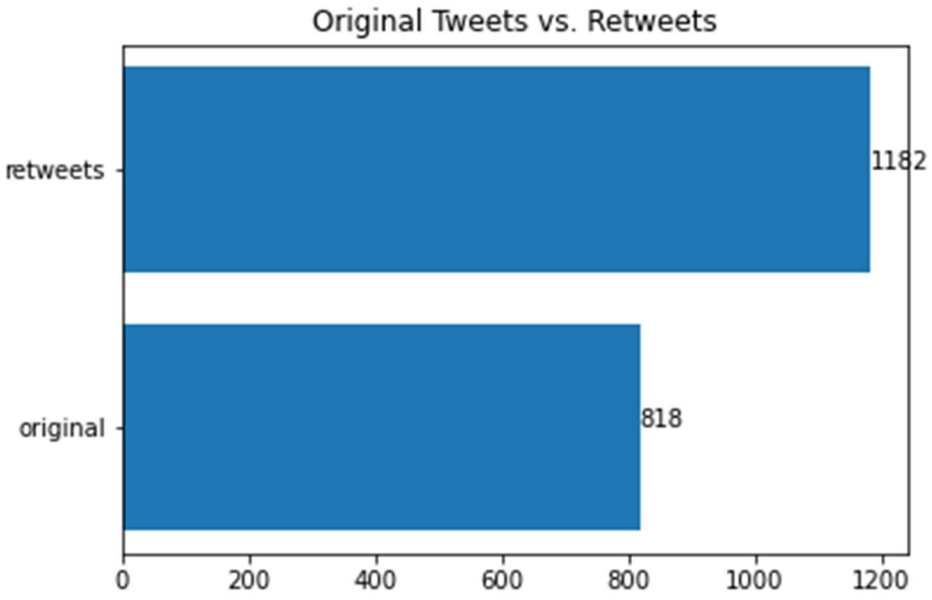


Figure 10: Original Tweets vs Retweets in Transgender Cases

The problem with the limitation in the number of tweets and the actual impact of each of these genders in society remains a critical factor in the true representation of each type of gender crime. However, in the ratios of conversation starters and those who supported them thereon, the graphs show similar results to those of men and women.

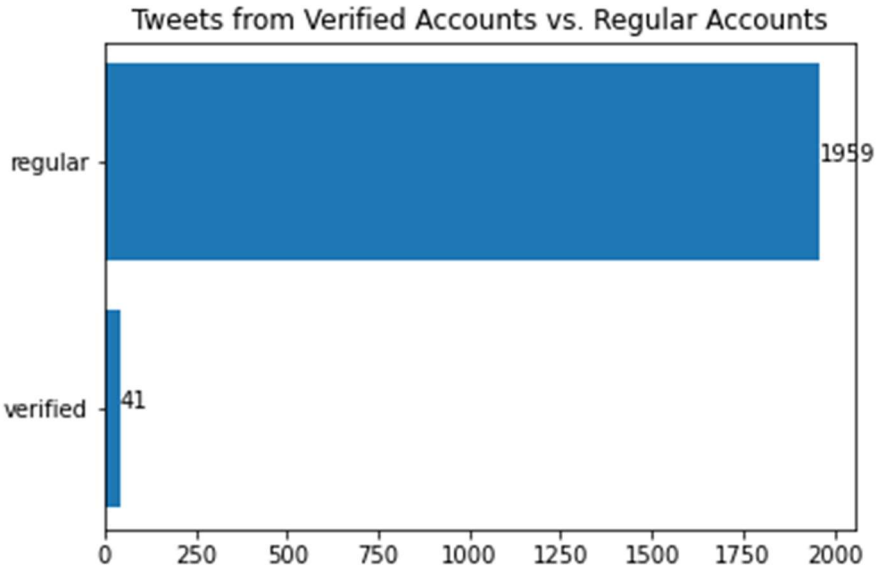


Figure 11: Tweets from Verified vs Regular Accounts in Transgender Cases

Even in the case of social media influencers (verified accounts and therefore people of marked influence), the ratio is quite similar in Figure 9 and Figure 11.

1.1.4 Pakistan Female and Male Community

As mentioned above, in Pakistan, violence, and abuse do not only affect the transgender community but other minorities like the female community, which even if not a minority is however below the patriarchal power typical of Arab countries. Therefore, it was decided to identify another network looking for information about maltreatment and feminicide in the country in the last five years. The keywords used to detect the network were: rape, victim-blaming women in Pakistan, incidents, female murder, brutally murdered women in Pakistan. The research provides us with cases of abuse, rape, and murder all over Pakistan furnishing names of victims in recent years. For this reason, the data collected by Twitter look like this: #JusticeForNoor, #StandwithNoor, #qandeelbaloch, #JusticeForQandeelBaloch, #ZahirJaffer, #noormukadam, #justiceformariam, #JusticeForHareem, #noorneedjustice, #honourkilling. In most of the cases, the hashtags are used together in order to remember all victims of abuse and harassment, and the fact of using them simultaneously in the same post allowed us to identify a network of nodes supporting women's causes against patriarchy and not just nodes defending one victim while ignoring the others. In this case, the nodes are **causally related** because of **evaluative reactions** including beliefs and feelings toward this social topic.

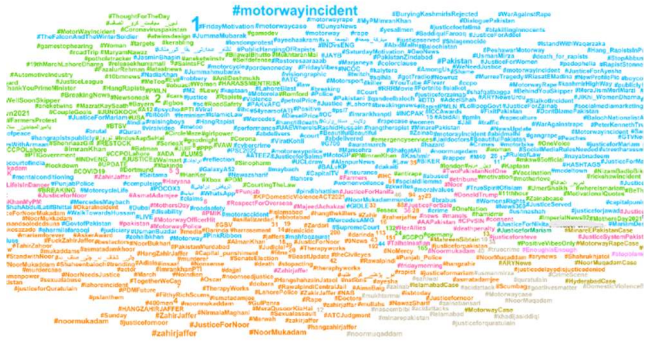


Figure 12: Main hashtags in female community

Figure 12 demonstrates that the female community is a very large network but in spite of this, violence against women is a growing reality in the country, which shows that there is no effective action by the national authorities to decrease femicide.

According to our research, we wondered if in Arab countries men suffered the same kind of violence, and if they were somehow abused as well. So, we started to look for news or any kind of report that mentions gender-based violence on men in Pakistan by using the following keywords: Acid attack, violence on men in Pakistan, abuse against male gender in Pakistan. In this particular case, the data collected are very few, the cases mainly concern women who, in order to escape the violence of their husbands, killed their executioners before they could do it to them. The most relevant hashtags for male community are: #iStandWithAliZafar, JusticeForAliZafar, #StayStrongUsmanMukhtar, #YumnaZaidi, #JusticeForNaseemBibi, #AliZafarIsinnocent, #iStandWithAliZafar.

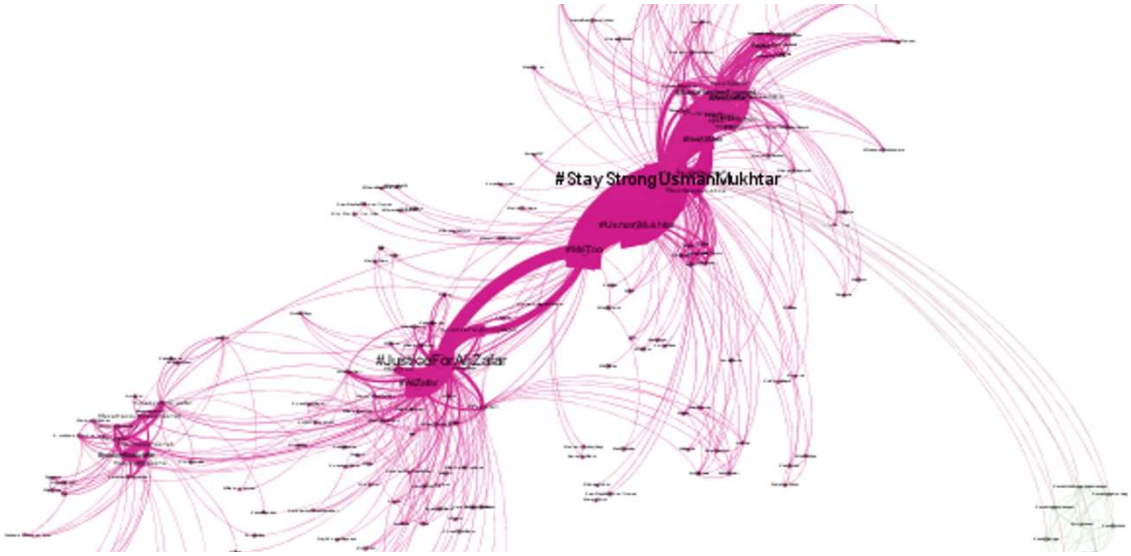


Figure 13: Main hashtags: #StayStrongUsmanMukhtar, #JusticeForAliZafar

As we can see in Figure 13 the **authorities** of the network are the following #StayStrongUsmanMukhtar, #JusticeForAliZafar, which receive links from a large number of hubs that are not strictly connected to our topic. The fact that in this community only two authorities receive many links confirms our initial ideas, i.e. men are not subject to violence based on their gender, or at least do not receive the same attention as other genders.

A frequent hashtag that occurs between the male community and female community is #noorneedjustice, as we can see in Figure 14 below #noorneedjustice is connected to the main authority in male cases: #StayStrongUsmanMukhtar. Following the path, we can see that #noorneedjustice is linked to #MeToo, a typical hashtag used in the network to identify with Women's fights for their rights. The fact that while ranking male cases for abuse and violence we found two hashtags that are typically used in the female community demonstrate our first ideas: men are not subject to violence based on their gender, or at least do not receive the same attention as other genders. According to the **Topic PageRank** theory, users searching for news about Usman Mukhtar by following the network will at some point be **teleported** to the female cases.

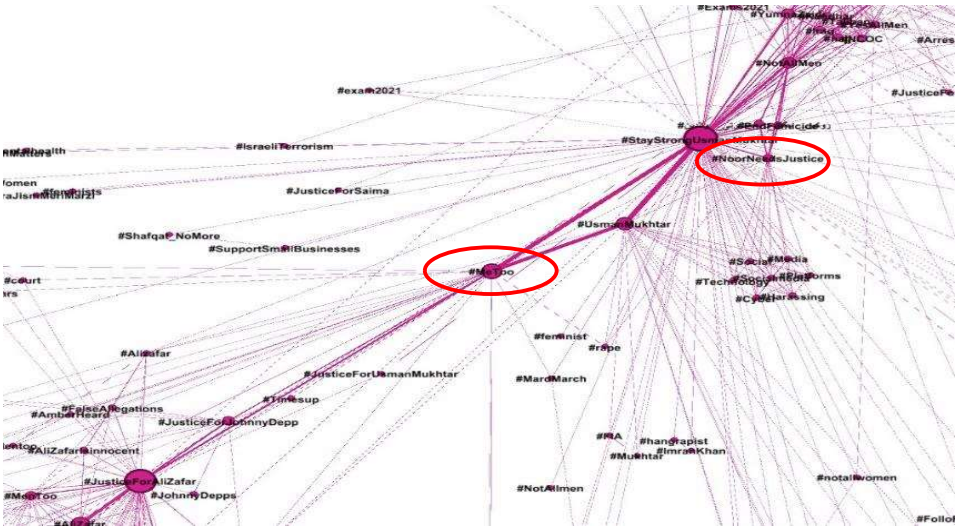


Figure 14: Hub (bridge) #noorneedjustice

1.2 Statistical Network Analysis Through Obtained Metrics

Thanks to the open-source network analysis and visualization software Gephi (Wikipedia, n.d.), we managed to measure various parameters of different networks we had created. Statistical analysis of a network allows us to summarize some aspects of a network by providing compression of information (Arratia). This section will make some general comments and interpretations about the networks based on various parameters.

	Egypt Transgender	**Egypt Female**	**Egypt Male**	**Pakistan Transgender**	**Pakistan Female**	**Pakistan Male**
Average Degree $< k >$	10.698	8.061	0.667	11.794	17.639	6.897
k_{min}	1	0	0	0	0	0
k_{max}	187	28	1	545	1596	97
Average Path Length	2.142	1.786	1	2.741	2.437	2.634
Average Clustering Coefficient	0.8256	0.922	0	0.926	0.855	0.889
Network Diameter	4	3	1	6	6	6
Graph Density	0.051	0.252	0.333	0.011	0.027	0.019
Modularity	0.487	0.403	0	0.602	0.386	0.566
Γ (gamma)	2.780	5.319	NaN	2.652	1.63	3.72

Table 4: Network parameters for hashtag networks

	Egypt Transgender	Egypt Female	Egypt Male	Pakistan Transgender	Pakistan Female	Pakistan Male
Average Degree $< k >$	10.698	30.711	21.913	63.562	17.639	82.264
k_{min}	1	0	6	1	1	1
k_{max}	1640	129	126	3610	1453	3570
Average Path Length	2.046	1.951	1.509	2.095	2.437	2.089
Average Clustering Coefficient $< C_{i\|k_i=k} >$	0.848	0.908	0.963	0.851	0.855	0.853
Network Diameter	4	4	4	4	5	4
Graph Density	0.054	0.135	0.333	0.012	0.002	0.015
Modularity	0.16	0.472	0	0.152	0.08	0.143
Γ (gamma)	2.37	5.753	2.875	2.320	1.63	2.108

Table 5: Network Parameters for Complete Networks

1.2.1 Average Degree

This measure will help any visual impressions we might already have about hubs and perhaps clustering by informing us about the typical number of neighbors per node (Cherven, 2015).
The average degree is simply the average number of edges per node in the graph. It is relatively straightforward to calculate (Lizardo, n.d.).

$$Average\ Degree = \frac{Total\ Edges}{Total\ Nodes} = \frac{m}{n}$$

Let's have a look at the Pakistan & Transgender & Hashtag network for this metric. In Figure 6, we see the average degrees of various hashtags.

The analysis of the degree distribution can give us some information on the network. Although we cannot generalize it to all networks, we can claim that having a strong tail usually is a good indicator of the existence of large degree nodes, in other words, hubs in the network. Furthermore, if the distribution follows a power law, we can go further and expect the network to be scale-free. Power laws are probability distributions with the form in Equation 1.

$$p(x) \propto x^{-\gamma}$$

Equation 1 : Power law generic form

Where γ stands for degree component and the overall equation is called power law. Networks whose degree distribution follows a power law rather than a Poisson distribution are called scale-free networks(can be generated with Barabási–Albert model).

For the gamma calculation, we utilized the power law library which uses Equation 2 below. The calculations in Python can be seen in the "Utilizing Python" section.

$$p(k) = Ce^{-\gamma k}$$

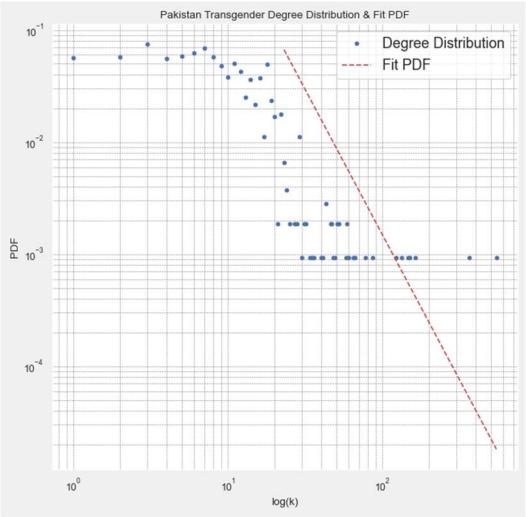
Equation 2: Power law open form

This differentiates those from random networks (can be generated with Erdős–Rényi model with small clustering coefficients which is unlikely in real-life networks). Since scale-free networks typically hold ($2 < \gamma < 3$) property, we can check gamma values and see if the distribution plot yields the same result.

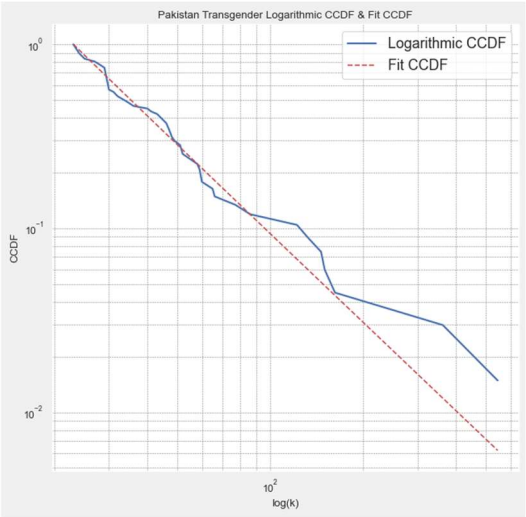
	Egypt Transgender	Egypt Female	Egypt Male	Pakistan Transgender	Pakistan Female	Pakistan Male
Γ (gamma)	2.780	5.319	NaN	2.652	2.69	3.72

Table 6: Gamma values for hashtag networks

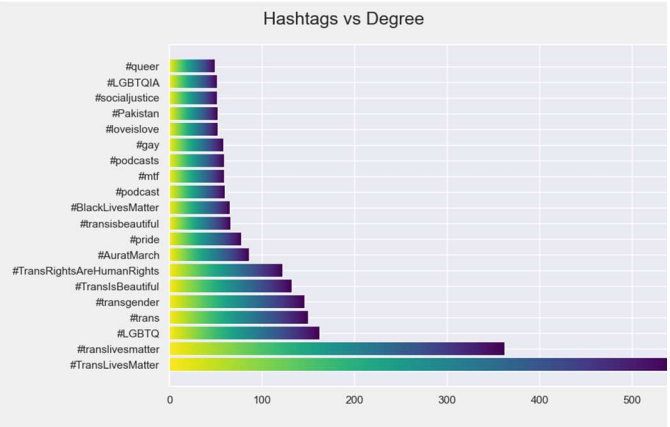
As seen in Table 6, Pakistan & Transgender & Hashtag, Pakistan & Female & Hashtag, and Egypt & Transgender & Hashtag networks seem to have a scale-free behavior. Let's inspect the first network thoroughly. In Figure 15a, the heavy-tailedness can be observed quite easily. It religiously follows the power law in the plot. This tells us there are numerous small-degree nodes that coexist with a few highly connected hubs in the network. To verify this, let's look at the network in Figure 15d. As seen in the figure, besides many small degree nodes, there are only a couple of nodes with high degree or hubs. Also, having more hubs causes the CCDF plot to decrease slowly as in Figure 15b. From Figure 15c, we can see that they are hashtags #TransLivesMatter, #translivesmatter, and #LGBTQ which makes perfect sense. These nodes have been used as bridges to carry the information by many people, which is the main definition of a hub.



(a)
Degree distribution & fit Probability density function (PDF)



(b)
Complementary cumulative density function (CCDF) & Fit CCDF

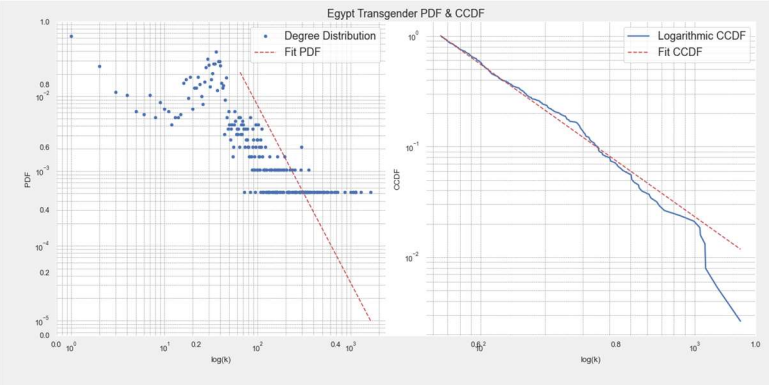


(c)
Degree bar graph

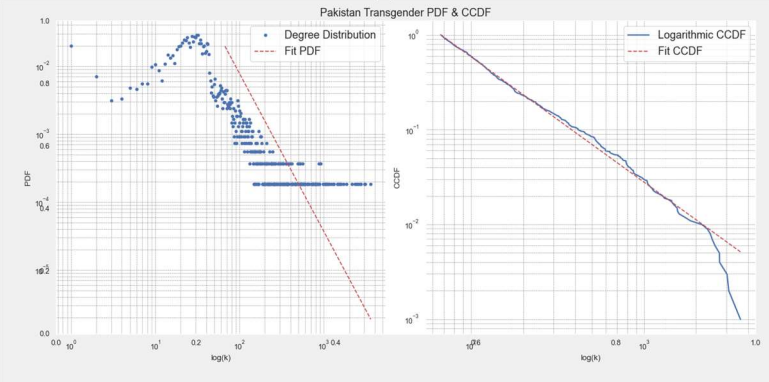
(d)
Network illustration

Figure 15: Degree distribution and fit PDF line & illustration of Pakistan & Transgender & Hashtag network. Nodes having red color and bigger size are of high degree

Even though we've decided to work on hashtag networks, complete network graphs have been inspected for further confirmation in Figure 16 below. As stated before, a clear heavy-tailed behavior is present in both networks due to large hubs.



Egypt & Transgender & Complete Network
(a)



Pakistan & Transgender & Complete Network
(b)

Figure 16: PDF and CCDF plots

If we would like to inspect the node degree from another perspective for our network, we can look at Figure 15c. Considering the bar graph, it is safe to say that the #TransLivesMatter hashtag has the highest number of edges; that is, it is the most commonly used one in the entire dataset. This makes sense for such a popular hashtag to have the highest rank in this plot because it reaches a larger audience.

If we would like to inspect the node degree from another perspective for our network, we can look at Figure 15c. Considering the bar graph, it is safe to say that the #TransLivesMatter hashtag has the highest number of edges; that is, it is the most commonly used one in the entire dataset. This makes sense for such a popular hashtag to have the highest rank in this plot because it reaches a larger audience. **However, these commonly used hashtags don't necessarily represent cases localized to our regions of study as they are part of global movements. The success of #TransLivesMatter is not a success for the attention received or impact-driven by crimes against transgenders in Pakistan or Egypt.**

1.2.2 HITS

The Hyperlink-Induced Topic Search (HITS) function is based on the work by Kleinberg, and it calculates two separate values for each node. The first, termed Authority, provides a measure for how valuable information stored by a particular node is, while the Hub number measures the quality of the links to and from that specific node. These measures can help identify or confirm the roles played by critical members within the network (Cherven, 2015). Let's start analyzing Authorities first.

1.2.2.1 Authority

This metric measures the quality node as a content provider. It is helpful to detect the nodes that contain useful information or frequent connections with other nodes (Erseghe, 2020). One comment that can be made on our case by looking at the authority numbers in Figure 17 (Pakistan Transgender Case) is that generic hashtags such as #TransLivesMatter, #trans, or # LGBTQ are immensely used in many tweets. Rather than specific hashtags for the crimes themselves, users felt the need to include the commonly used hashtags to reach a larger audience. In that, we can interpret this pattern in a way that the crime events didn't attract much attention, so using the case-related hashtags wasn't enough for users to raise their voices. **However, these commonly used hashtags don't necessarily represent cases localized to our regions of study as they are part of global movements. The success of #TransLivesMatter is not a success for the attention received or impact-driven by crimes against transgenders in Pakistan or Egypt.**

Figure 17: Authority pie charts for hashtag networks

1.2.2.2 Hub

Hubs are bridge notes between important nodes that carry information of paramount importance. It is related to the number of links that connect many authorities. One wouldn't expect the same hub and authority value for a network, but we take a look at the plots in Figure 18, we observe that they are identical. As a first reaction, we assumed to have made a mistake in one of the intermediary steps. However, after making sure that there was no error on our side, we tried to find an explanation for this occurrence. Thus, the explanation for this phenomenon is that all nodes have the same number of parent and child nodes in this undirected network. Therefore, in the link perspective, they are basically all the same. That's why they all have the same authority and hub value (Chonny, 2021).

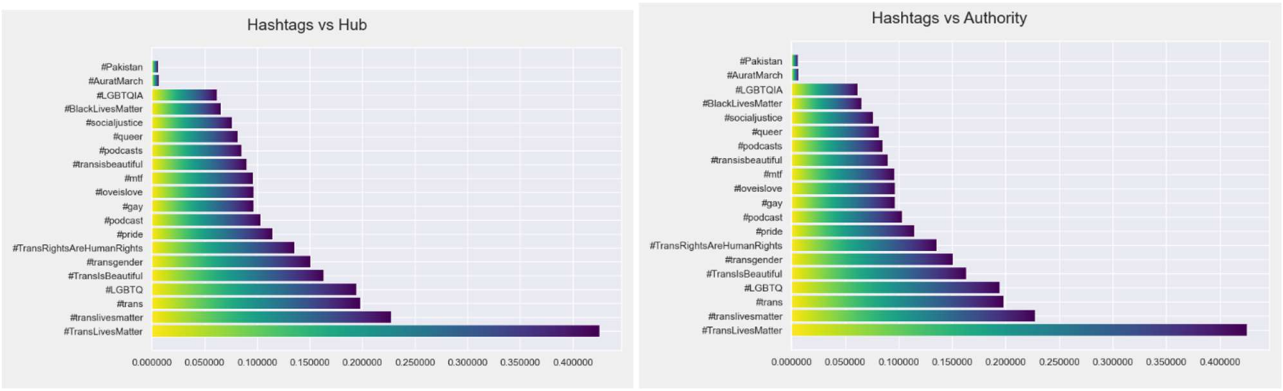


Figure 18: Hashtags vs Hub plot

1.2.3 PageRank

The degree of a node in social networks can be prescribed as the node which has the most influence in the network. In a topic-specific graph, the degree of nodes can be correlated as the issuer of a topic. The degree of each node can be computed directly with valency or through a graph-based ranking. One of the most popular graph-based rankings is PageRank which ranks nodes of a graph based on the degrees of the nodes. PageRank determines the importance of a node within a graph, by computing the information on the graph globally and recursively. The original purpose of PageRank is to rank all web pages based on the interconnection around that page, aside from each content of the pages (Sigit Priyanta, 2019). The PageRank formula is:

$$x_i = d \sum_j A_{ij} \frac{x_j}{\text{outdeg}(j)} + \beta$$

Equation 3: PageRank formula

Where d is called the damping factor, which can be set to between 0 and 1(or the largest eigenvalue of A). Moreover, the default value for β in Gephi i the probability = 0.85 (Gera). Since we're dealing with undirected networks, PageRank is calculated by treating the undirected graph as a directed graph; that is, it makes each edge bidirectional. PageRank values for hashtag networks are given in Figure 19.

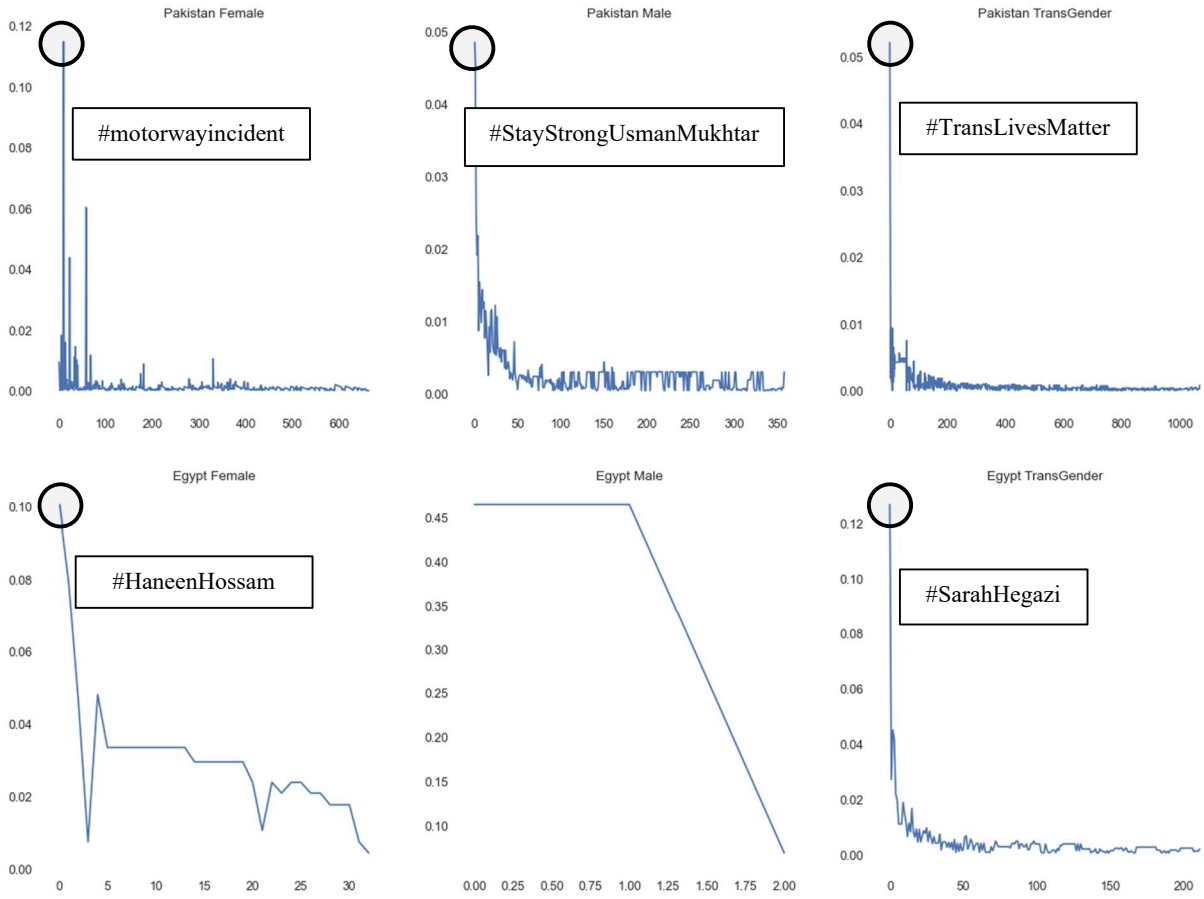
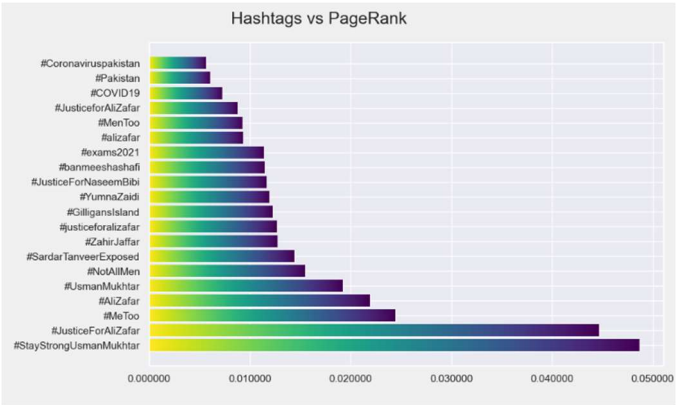


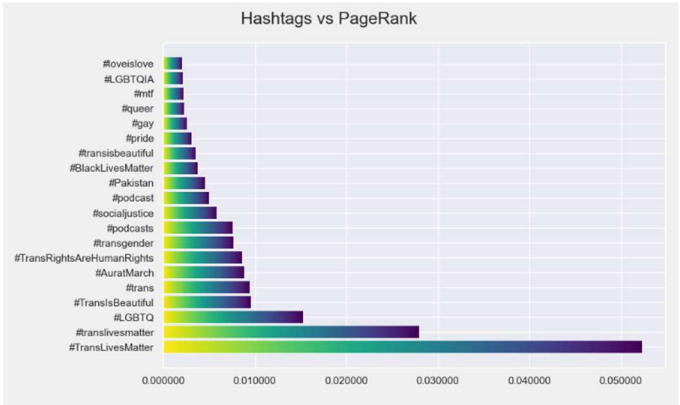
Figure 19: PageRank plots for hashtag networks

As a confirmation, when we look at the bar graphs below in Figure 20, we can see the hashtags match the previously obtained values.



Pakistan Male Hashtag

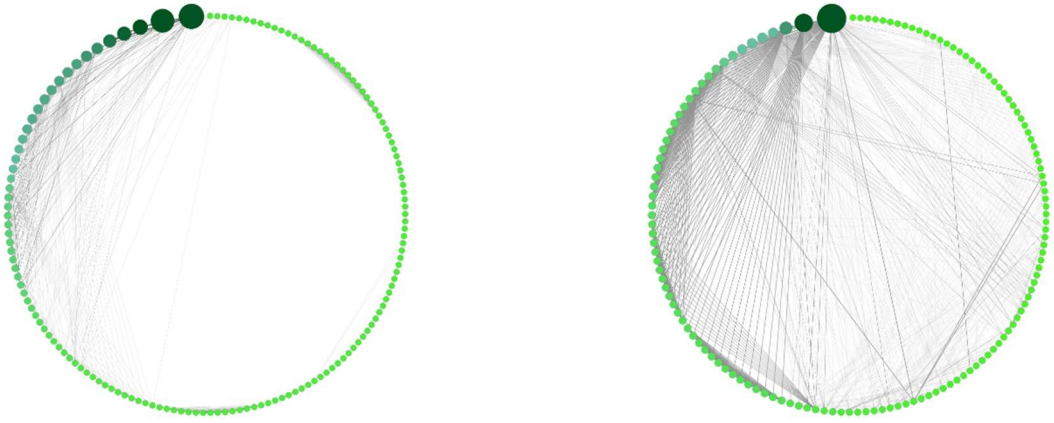
(a)



Pakistan Transgender Hashtag

(b)

Figure 20: PageRank bar graphs of two example networks



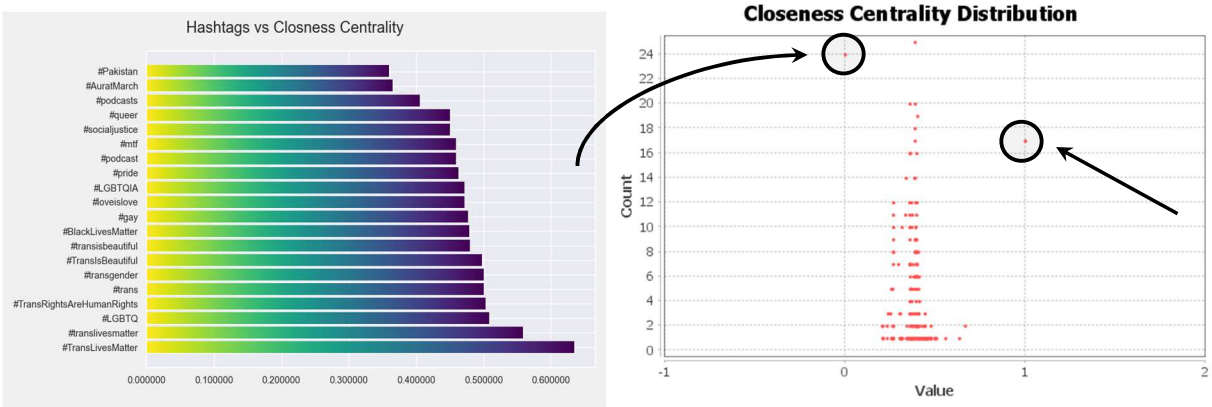
Pakistan Male Hashtag
Node reduction applied
(PageRank > 0.00267)
(a)

Pakistan Transgender Hashtag
Node reduction applied
(PageRank > 0.00123)
(b)

Figure 21: Circular Layout applied networks whose node size & colour are determined by PageRank. Bigger and darker nodes are of higher PageRank values

1.2.4 Closeness Centrality

Closeness centrality represents an interesting case wherein the selected node might actually be poorly connected in a direct sense yet is still highly influential due to the proximity of well-connected neighbors. Consider the case of someone who is seen as a gatekeeper to an important and influential person; this individual might have relatively few first-degree connections but is still well-positioned due to the presence of a high proportion of highly connected nodes as direct connections (Cherven, 2015). Simply put, it ranks the nodes based on the fact that which one is the easiest to reach and which is the best for spreading the information (Erseghe, 2020).



(a) (b)

Figure 22: Closeness Centrality bar graph and plot, Pakistan Transgender Hashtag Network

When we investigate Figure 22, we observe the expected result. Since closeness centrality lets us detect the nodes which are the best to spread information, one could easily claim that the hashtags with higher closeness centrality values are the popular ones. This way, the message the user wants to convey to the media will have a better chance of being heard of and seen.

In Figure 22b, it is safe to say that the node on the far left was unable to become successful in spreading the information, unlike the one on the far right, thus being ranked the lowest in this distribution.

1.2.5 Betweenness Centrality

The betweenness centrality presents us with a rather unique case, identifying nodes that might well be poorly connected as defined by other centrality measures. In cases where these nodes offer the most direct path between otherwise disconnected clusters, we have what is often termed as a bridge. Being a bridge is not a necessary precondition to have a high betweenness centrality score, but it is often the case that these nodes will rank as critically important using this measure (Cherven, 2015). To prove this claim, let's look at Figure 23. Since being a bridge is usually related to hubs, we can observe that for most of the cases, nodes with high betweenness centrality values also exhibit high hub values.

Additionally, this metric allows us to detect the node which takes us elsewhere, or the bridge nodes in other words (Erseghe, 2020). In Figure 24, we observe that frequently connected nodes are the generic and popular ones such as #TransLivesMatter, #MeToo, #LGBTQ, #Pakistan, and so on. Since they can be used in different contexts, they connect different tweets that are not necessarily strongly correlated. As a consequence of this attribute, these nodes are positioned at a higher level in the betweenness centrality graph.

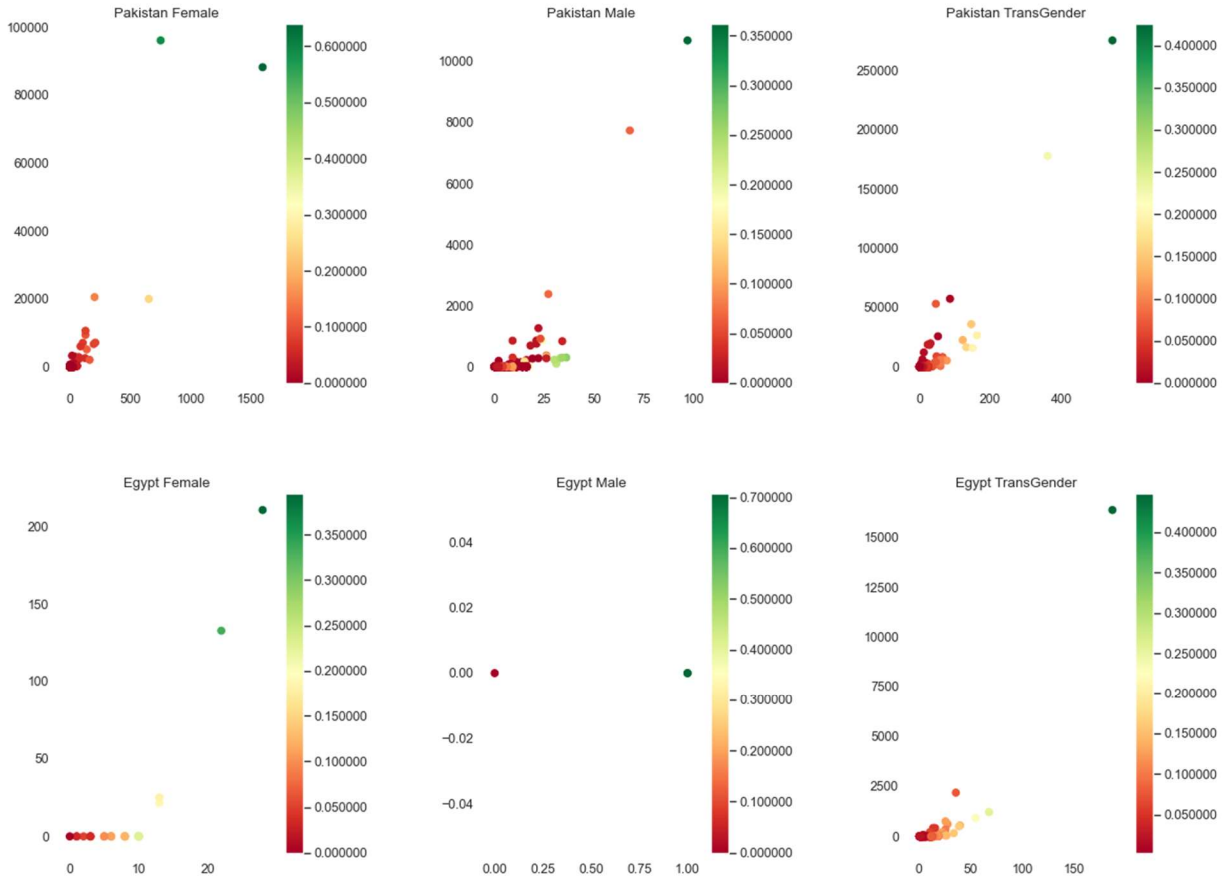
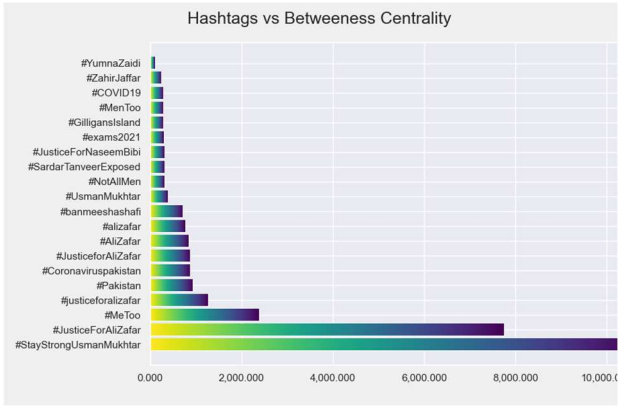
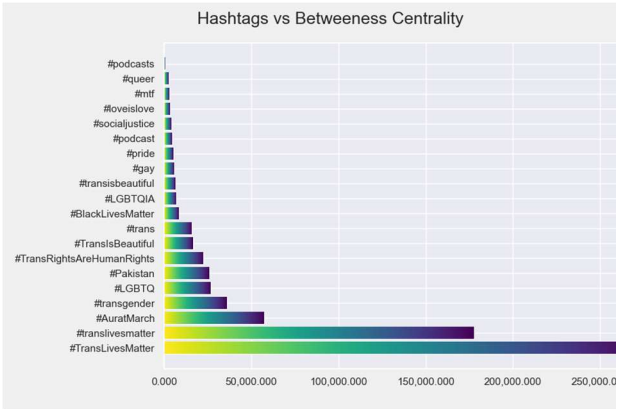


Figure 23: Color map plots of hashtag networks. Y-axis represents betweenness centrality, X-axis represents average degree, and color map represents hub values of the nodes



Pakistan Male Hashtag

(a)



Pakistan Transgender Hashtag

(b)

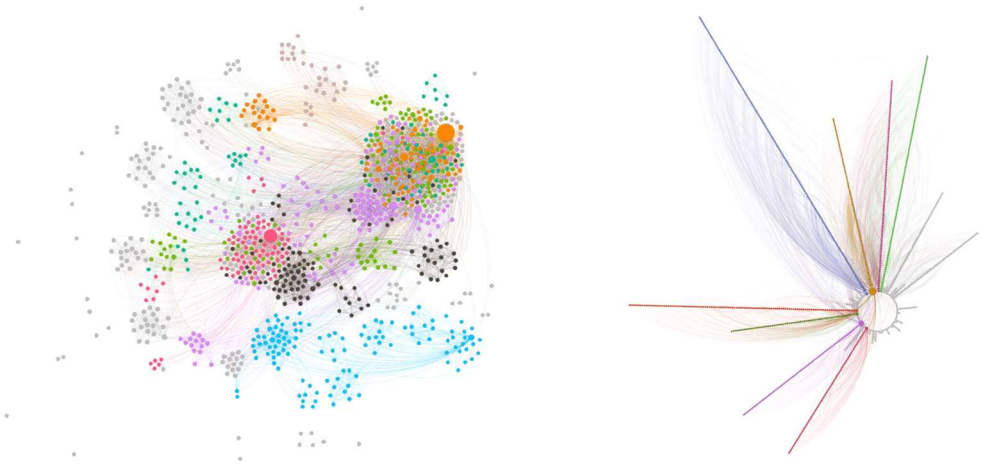
Figure 24: Betweenness centrality plots

1.2.6 Modularity

One of the approaches to measure clustering in a network is through the application of the modularity statistic, which attempts to assess the number of distinct groupings within a network. This can be done simply by using this statistic or through the use of one of the Gephi plugins geared to parse nodes into distinct groups. Many algorithms can be used for this purpose. The end goal for any of these algorithms is to group nodes based on the strength of their relationships. Nodes that are highly connected are likely to wind up in a common cluster, regardless of which algorithm is employed. Yet each approach will return slightly different results depending on the size and structure of the network as well as the statistical thresholds employed (Cherven, 2015).

In Figure 25a, Pakistan & Trans & Hashtag network is taken as an example. We can interpret the circled value in the figure as cluster 55, having almost 125 nodes in it. Also, it is shown that the network is separated into approximately 57 classes, and each class has a different number of nodes. In Figure 25b, the network is illustrated to show various clusters using the Frutcherman-Reingold algorithm. In Figure 25c, Radial Axis Layout has been used for a clearer cluster view. As seen in the network, there are 57 clusters having different colors. Due to the high number of clusters, we can say that shared tweets contained a high volume of unrelated information besides the main topic, and thus this has caused the network to be divided into many groups.

(a)

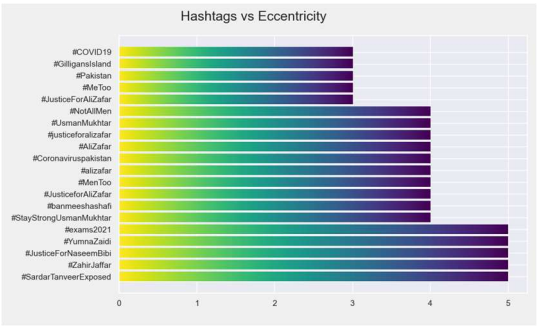


(b) (c)

Figure 25: Modularity class distribution and network designs. Different colors represent different clusters.

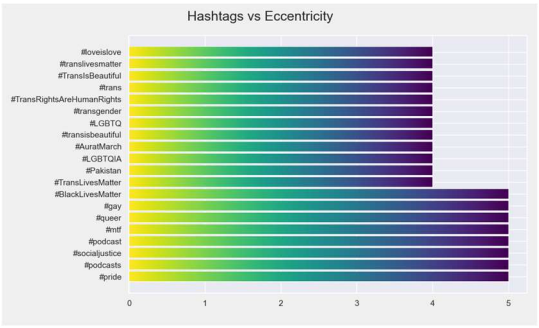
1.2.7 Eccentricity

Eccentricity refers to the number of steps required for an individual node to cross the network. When used to compare nodes, eccentricity can help provide some context to assess the relative position and influence of nodes within a network. While it is not a substitute for the various centrality measures, eccentricity can nonetheless provide some clues into the relative importance of individual nodes within the network (Cherven, 2015). We deduct from Figure 26 that even though this metric is not sufficient to decide the importance of a single node, it provides insight into the structure of the network and trend hashtags used in various cases.



Pakistan Male Hashtag

(a)



Pakistan Transgender Hashtag

(b)

Figure 26: Hashtags vs Eccentricity bar graphs

1.2.8 Clustering Coefficient

Scores on this measure have an inverse correlation with other statistics. The calculation of this metric is based on calculating the number of closed triangles (triplets) relative to the potential number of triangles (triplets) available in the network. Also, the clustering coefficient places more weight on the low degree nodes, therefore exhibits an inverse proportional relationship with the node degree as in Figure 27. The slope of the graph gives us the inverse proportion coefficient between these two metrics. Additionally, we can say that the information carried on a node with a high clustering coefficient is more likely to be true. Since this node is of a low degree, it doesn't have different types of nodes as neighbours. One can then suggest that the information transmitted through a cluster of nodes of similar types, having higher clustering coefficients, is more reliable. As an example, since the hashtag #pride is widely used, it is highly likely for it to be used in an unrelated tweet that includes wrong information.

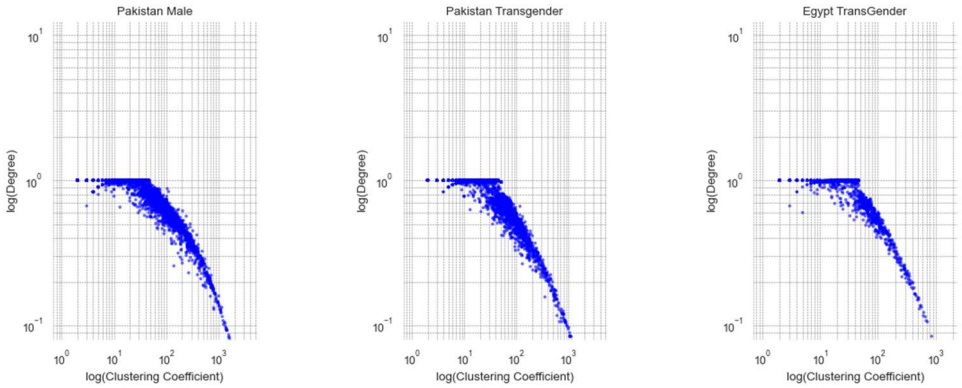


Figure 27: Blue dots are nodes of the network, represented according to their degrees (X-axis, in logarithmic scale) and their local clustering coefficient (Y-axis, in logarithmic scale)

1.3 Egypt Cases

Once we identified these three different clusters for one country, we did the same for the other one. Once again, Egypt is a strongly Islamic country where the law to be respected is the so-called Sharia, Sharia is Islam's legal system which is derived from the Quran, the holy book in Islam. Egyptian law, by contrast to Pakistan, doesn't recognize the change of identity of transgender people but rather criminalize any behaviour or the expression of an idea that is deemed to be immoral, scandalous, or offensive to the teachings of the recognized religious leader. In light of the public opinion, these public moralities and public order laws have been used against LGBT people as well as anyone who supports approaches to LGBT issues, this is why Egyptian society, do not express public support for LGBT rights.

Most of the data were in the Arabic language, this emphasised that Arab culture in Egypt is narrow and therefore not very open to social issues that are being fought against all over the world and that in a culture that condemns any kind of diversity, these issues must be fought against even more.
To gather the information about the first cluster, the transgender community in Egypt we looked for any type of crime against the community, the keywords used were: Egypt sexual and gender-based violence, human security, human rights, the crackdown on transgender, anti-LGBT crackdown. However, the outputs from the web and Twitter were very low: #ComingOutDay, #SarahHegazi, #SuicidePreventionDay, #LOVEISLOVE, #RaiseTheFlagForSarah, #LGBTQ. The fact that we were able to collect few data demonstrate that there is a sort of **ostracism** that led to negative psychological effects (depression, suicide) and that Egyptian authorities seem to be competing for the worst record on rights violations against LGBT people in the region, for that reason few nodes decide to publicly support the topic aware of the social and criminal repercussions.

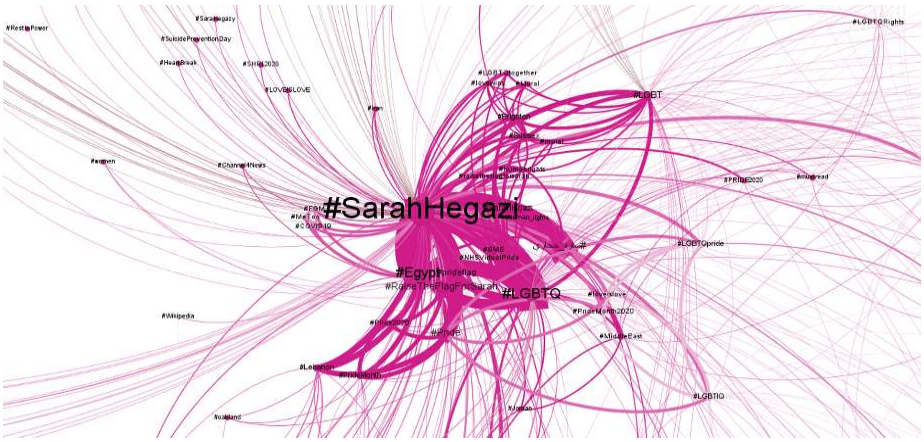


Figure 28: Main hashtag: #SarahHegazi

Having had these kinds of problems with the transgender community: users are afraid to talk and stand up with this topic, we wondered if is the same with the events concerning the rape, kidnapping and murder of women, a frequent reality in Egypt. Indeed, we looked up for information on news, articles on blogs about harassment on women, rape, violence and abuses on women and also in that case the network were built around specific names of victims like #ManarSamy, #HaneenHossam, #MennaAbdelAziz, #BassantMohamed, #HadeerHady, the nodes, in that case, represent **authorities** that receive direct links from **hubs** like #tiktokers, #Egyptianmoralitylaws, #HumanRights, #womensright, #Censorship.

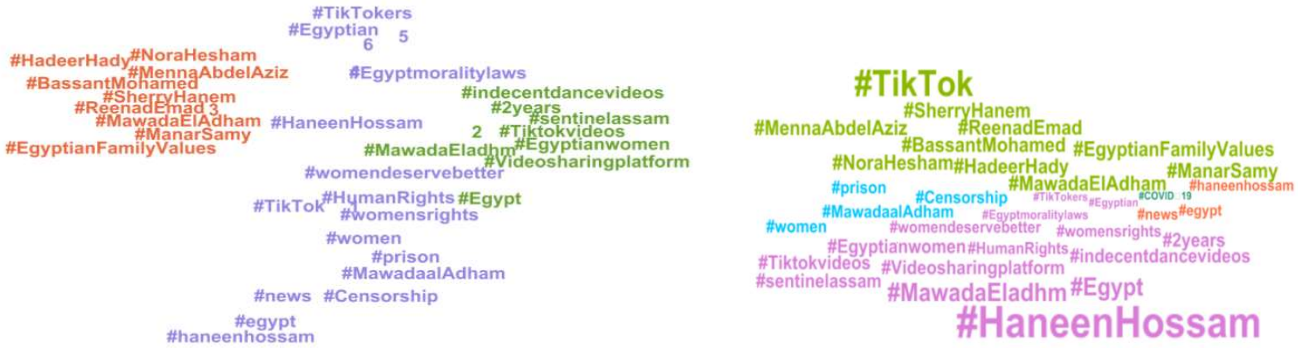


Figure 29: Cloud of words: authorities and hubs

In contrast to outputs concerning the transgender community, people are more likely to not shut up in front of harassment on women, and they are more likely to raise their voice when the event occurs remembering all victims to make national authorities to act in order to protect Egyptian women.

Although the topic of violence against women is one that touches Egyptians very much, there is still a general fear of talking about certain topics publicly. This is also demonstrated by the following Figure 30, where we tried to see if there are certified accounts (i.e. recognised by Twitter as people with a certain relevance to their followers) if they use the most common hashtags to identify themselves with a community, creating a kind of homophily. The tweets from verified accounts that we were able to collect are very few compared to those from regular accounts. PageRank was used in order to rank main hashtags for the female community like #ManarSamy, #HaneenHossam, #MennaAbdelAziz, #HadeerHady, #womenrights and #Egyptinawome, But the results underline once again that freedom of speech in Egypt is a luxury few are willing to pay for.

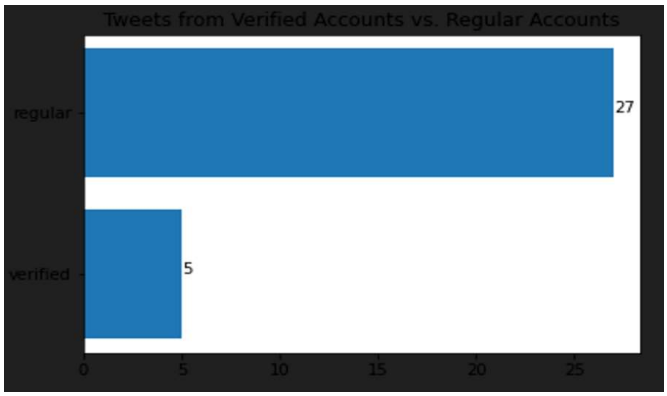


Figure 30: Tweets from Verified accounts Vs. Regular accounts

In the end, we investigate men cluster: do they undergo the same maltreatment or because their gender are more advantaged? In such a case we were not able to collect as much data as we wanted and that demonstrate that in Egypt male gender does not suffer the same injustices as the transgender community and women. The lack of hashtags and information, therefore, confirms our initial thought: in Egypt, as in most Arab countries, men are less prone to gender-based crimes.

1.3.1 Egypt Transgender Cases

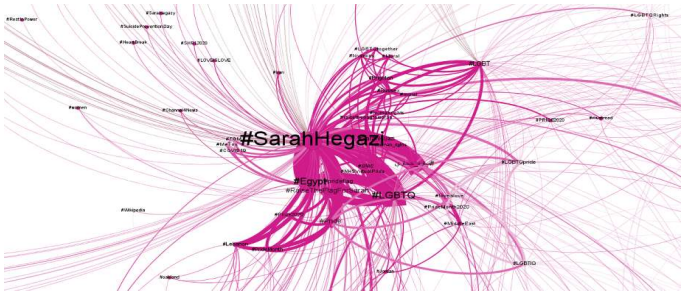


Figure 31: Main hashtag: #SarahHegazi

The main hashtags #Sarah Hegazi she was a software developer, a "feminist, interested in politics and a queer activist," fellow LGBTQ activist Sarah Hegazi the girl who was arrested after raising a rainbow flag in a Mashrou' Leila concert (Egypt) choose to end her life in Canada. Her last words to the world: "You were extremely cruel to me but I forgive you". In Egypt, transsexuals are fighting for their legal right to change their official government documents and there is a lot of connection to some hashtags like #Egypt,# free Sanaa, #Boycott, #tourism,#Patrick Zaki which present a small **cluster** in transgender community related to the main problem that is no freedom in Expression in Egypt and no one can express or speak about human rights and anyone speak or make comment will be arrested as #patrik zaki EIPR Gender and Human Rights Researcher Patrick George Zaki, was stopped at Cairo airport on his return home from abroad. Patrick, who has been on leave since August 2019 to study for a postgraduate degree in Bologna, Italy, was returning for a brief family visit when he was taken into the custody of Egypt's National Security Investigations at the airport and disappeared for the following 24 hours.

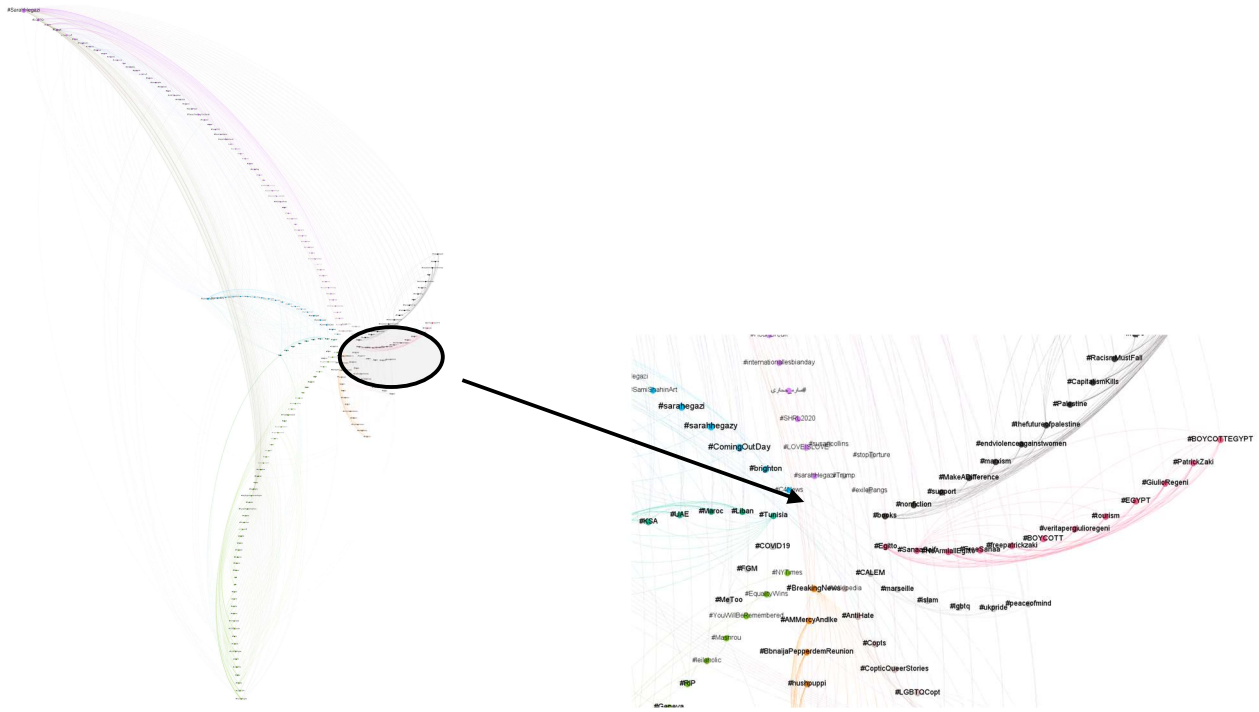


Figure 32: A clear cluster example in transgender community

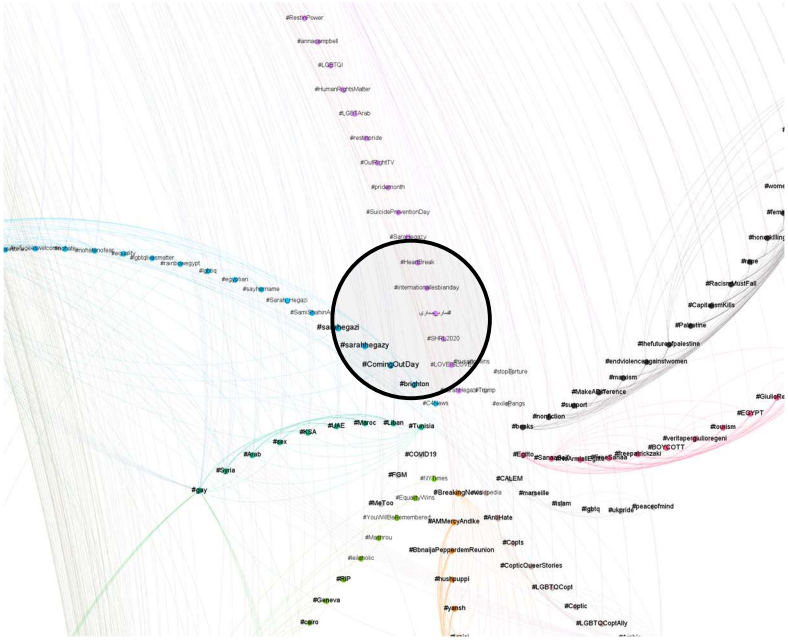


Figure 33: Bridge node considering the Betweenness Coefficient value

As we can see in Figure 33 above the node #حجازي_سارة which is very close to the main authority is a bridge that connects the transgender community to a small peripherical cluster. In the cluster, as we can see, the word women appear in several languages, an attempt by the transgender community to have their situation talked about all over the world. We would also like to underline that it was the linguistic diversity that made us focus on hashtags rather than words.

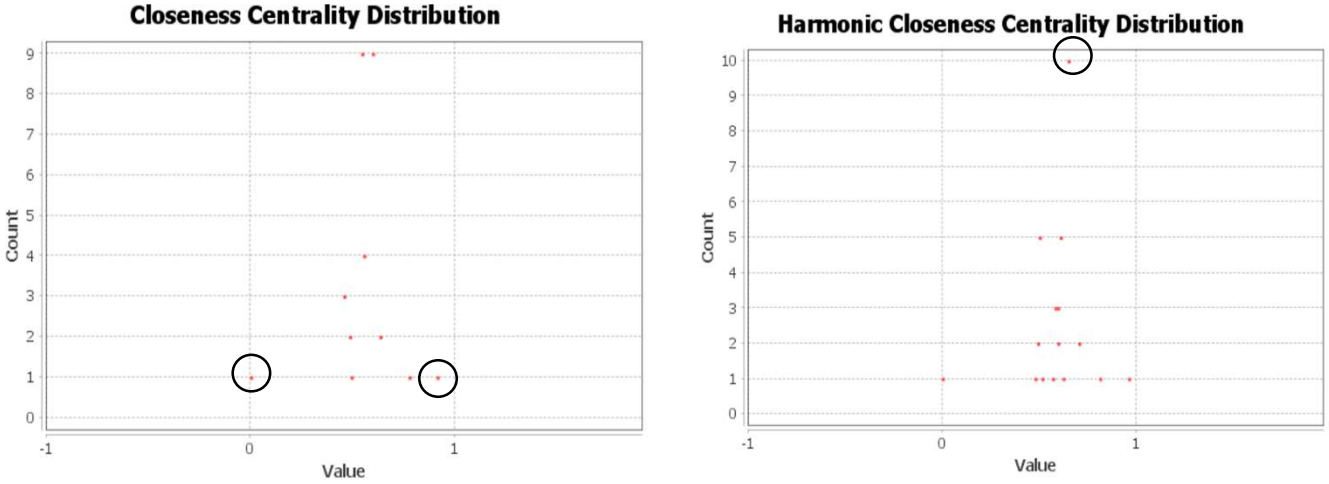
Then will find some connection hashtags like #rascism, #capitalism #kills, woman's and then hashtags #Suicide which related to Sarah Hegazi case Suicide of Egyptian activist Sarah Hegazi exposes the 'freedom and violence' In Canada, Sarah Hegazi escaped the violence of the Egyptian state but not, as she wrote in a 2018 essay, the post-traumatic stress disorder, depression and loneliness caused by her past. Like many queers and trans people living in exile, Hegazi felt rejected by her own people yet mourned the home she left behind, and that presents a typical case of **ostracism**, cause at the begging the case of Sarah Hegazy was very discussed but then the news quickly faded. Indeed #SaraHegazy became viral only after her death. Also, the #MalakAl-Kashif hashtag is very popular in newspapers more than on twitter a woman human rights defender, who defends trans people's rights and counters transphobic discrimination. She also works on social and economic rights. Malak Al-Kashif has not been able to obtain official papers reflecting her gender, despite attempting to do so for nearly three years. She is therefore treated as a male in all official papers, which means that she could soon be detained in the men's prison.

1.3.2 Egypt Female Cases

Figure 34: Hashtags vs Words Networks

As we can see in Figure 34, hashtags are well used to express membership to a community; to express proximity to a specific topic twitter users prefer to use hashtags and in most cases, hashtags co-occur together even if the cases are not strictly connected but all together belong to a social fight: violence, harassment and rape on women need to stop.

Figure 35: Closeness Centrality and Harmonic Closeness Distribution



As we said nodes are very few and many cases are repeated, however, there are some nodes that are central in our network while others are of little relevance in the dissemination of information, as can be seen in Figure 34 the nodes with the lowest value are distant from the authorities.

The main Hashtag here is #HaneenHossam because it is the most popular one on Twitter. the prosecution charged Haneen Hossam and Mawada Eladham with human trafficking for employing girls in crimes against Egyptian society's norms and values in order to obtain pecuniary gain. This humiliation is the outcome of society and the official media blaming a victim for her clothing choices and public behaviours. Haneen Hossam, a 20-year-old archaeology student at Cairo University was accused of human trafficking and arrested for "inciting debauchery, immorality and violating public morals" after she posted a clip on a short-video sharing platform Like stating that young women and girls can earn up to $3000 by "making live videos and talking to strangers."

#HaneenHossam and #Mawada Aladhm are two of nine female TikTok video makers who were detained last year on allegations ranging from "violating family values" to "inciting debauchery." Human and women's rights organisations have condemned the arrest campaign, claiming that officials used loosely worded and contentious legal regulations, such as "violating family values," to unfairly target women from low-income backgrounds. The accusations were filed under the contentious 2018 cyber-crime legislation, which criminalises activities that breach Egyptian family values without setting clear legal criteria for what constitutes a violation of such values. Legal experts and campaigners claim that the broadly written phrase leads to unjust criminalization and is disproportionately utilised to regulate women's bodies, as well as that the arrests were initially prompted by a misunderstanding of the law. Sexual harassment is a serious public hazard to women in Egypt, for which there is no long-term policy or practice. Harassment may take several forms, ranging from verbal to physical, the latter of which can result in serious injury and, in extreme cases, murder. According to a 2013 UN report, nine out of 10 Egyptian women have experienced sexual assault, which can range from simple harassment to rape.

1.3.3 Egypt Male Cases

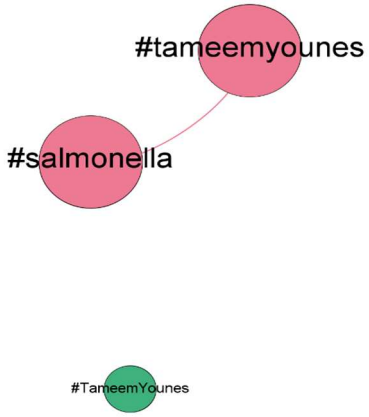


Figure 36: Egypt Male Network

#Tameem Younes This is a highly ranked case in Egypt focused around a very influential celebrity songwriter and known for his viral song "#Salmonella", who at the age of 22 was harassed by his dentist and the story of the entire event was broadcasted on his social media profile. Tameem realised right after this event the trouble and trauma that all women undergo when they are abused by men in powerful positions. The case in itself is a rarity as the power dynamics in Egypt favour men to be the privileged gender and therefore barely any gender crimes are ever committed against them. If there even are any, these crimes are not spoken about publicly because the government flags them. This is the very reason behind #TameenYounes being the only hashtag we have been able to analyse.

1.3.4 The Reason of Inadequacy in Egypt Networks

Egypt was ranked horribly by the gender inequality index because societal conventions, attitudes, economic constraints, religious beliefs, and structural forces, all contribute to the status quo. Gender-based violence is one of Egypt's most serious human rights abuses. Egyptian authorities have a horrific track record of persecuting people based on their sexual orientation and gender identity. Constitutional rights are often suspended under the state of emergency. The government can intercept and monitor all communications, impose censorship, and confiscate publications under Article 3 of Egypt's Emergency Law. The government tightly controls conventional media such as newspapers and television, and has also used court decisions to monitor internet sites outside its reach.

In the middle of all this, fair reporting on social media by the Egyptian population is impossible. People have to remain discreet about violations and are heavily penalised against any display of resistance. While living in Egypt can easily expose you to the stories of vast and deep rooted abuse, the internet or television will hardly give any information on cases. Which is why we were not able to acquire data on more hashtags here and decided that the subject of hashtag activism becomes irrelevant in the region.

1.3.5 Censorship in Egypt

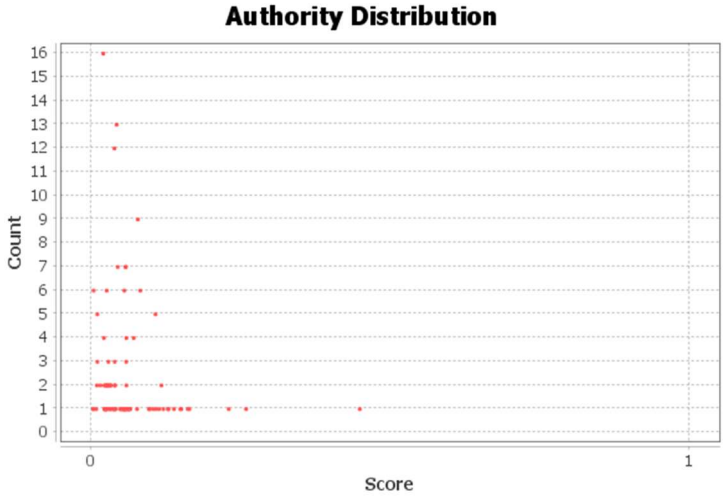


Figure 37: Authority distribution of Egypt Transgender case

The authorities are defined as nodes with smaller degrees than the average node degree. If we look at the authorities' responses in Egypt to male, female and transgender cases, we will observe that the access to the internet access in Egypt has been increasing over the last few years and the power of social media sites like Facebook and Twitter have been growing because of unprecedented political changes. For the first time, on such a large scale, technology-savvy protestors swiftly devised ways to get through the Egyptian government's roadblocks. Facebook and Twitter have made it possible for people to cooperate together on problems that they care about. The first step toward collaboration is to have a common understanding. But this use of Facebook and Twitter is being discouraged by the authoritarian government's intervention. It is unsurprising that authoritarian administrations in Egypt have attempted to limit Internet freedom. President Hosni Mubarak's government attempted to crush protesters by shutting down internet access. Egyptian authorities think that there is no difference between what a person does on social media and what he does in the real world, therefore anyone who expresses his opinion about any topic online, he will be charged with a denouncement and defamation.

2 Conclusion

We attempted to look at the differences in social response from the lens of hashtag activism in two countries, which are amongst the worst off in the world in terms of gender discrimination and minority marginalization. We labelled crimes as acts of trespassing the rights of an individual, and using this as a basis for segmenting individuals by gender, we hoped to achieve a clearer understanding of which gender/type of individual matters most to the society. Both Pakistan and Egypt are democratic countries, ideally meant to be a culmination of the voice of the people. The medium of expression in both the countries is also being shifted more and more onto social media. For which, it seemed like a plausible endeavour to gauge where society stands on the status and importance of each gender by checking their online responses to gender crimes..

Looking at the data, it is imperative to understand that our limitations as researchers and the region's limitation as political grounds plagued with censorship and social media access don't make the idealistic grounds set with expectations of a true representation of gender inequality quite possible. The threats that vocalisation faces in Egypt, due to which we had to drop the country from our research question altogether, along with hurdles in getting a full number of tweet related data for gender crimes hashtags in both countries make the conclusion a compromise.

On analysis, we see that female, male and transgenders in Pakistan face an equal ratio of social activism in two categories

1) The ratio of conversation starters to conversation supporters
2) Ratio of social media influencer support (verified accounts) to common man conversations

Beyond this, we also observed the complexities in the networks that elaborate the kind of networks these hashtags support, but our original research question lies amidst its limitations at best.

3 Drawbacks in the Research

The research was carried out, as discussed above, focusing on two countries with a predominant religion: Islam. Religious morality actually shapes the law and order system for these places. For this reason, especially in Egypt there is general censorship in order to cover up the brutalities committed for the sake of maintaining order in the country. People, for fear of being imprisoned, tend not to express their opinions on social media and when they do, they often do not indicate the position from which they are posting. They also tend to use different languages rather than Arabic so as to not identify themselves. Because of this, it was difficult to retrieve tweets based on location. We had to put faith in the fact that the hashtags themselves are such localised cases that only people belonging to the same community would be talking about them online. This is where we removed location barriers and ran a worldwide search for our communal cases.

The next hurdle lay in identifying content and relevance in tweets not made in English. We had to leave out a huge chunk of data in Arabic simply because it could not be decoded.

After conducting research on relative cases, multiple hashtags were chosen to be used. We realised though that

1. The hashtags were popular in different time periods. Therefore, searching them in a single time frame will not get a true representation of their network.
2. With the current Sandbox tier, we weren't allowed to retrieve the exact location (country name) embedded in the queried tweet.
3.

For problem 1, we rearranged our hashtags and increased our time period in the query to a logically vast period of time.

For problem 2, we created a list of all cities in Pakistan and Egypt. Then, we scanned the data manually and grouped them according to the countries.

Some confounding variables which may have played a manoeuvring role in the outcome include issues such as:

1) Not all cases get hashtags. So, there are a lot more cases out there than there are hashtags.

2) Social media algorithms can play a huge role in which sort of news gets the most amount of coverage. A parallel event that is more trending but less severe, or completely irrelevant to gendered crimes, could be isolating a hashtag

3) The hashtags studied are all in English. Local language-based hashtags could not be studied

4. Data Plotting and Visualization

To have a better view of the retrieved data, we plotted them in Python. For more attractive visuals, we utilised matplotlib and seaborn libraries. The code snippets are presented in the "Utilising Python" section. Each network metric for the relevant case is depicted in a bar and scatter plot as shown in Figure 38.

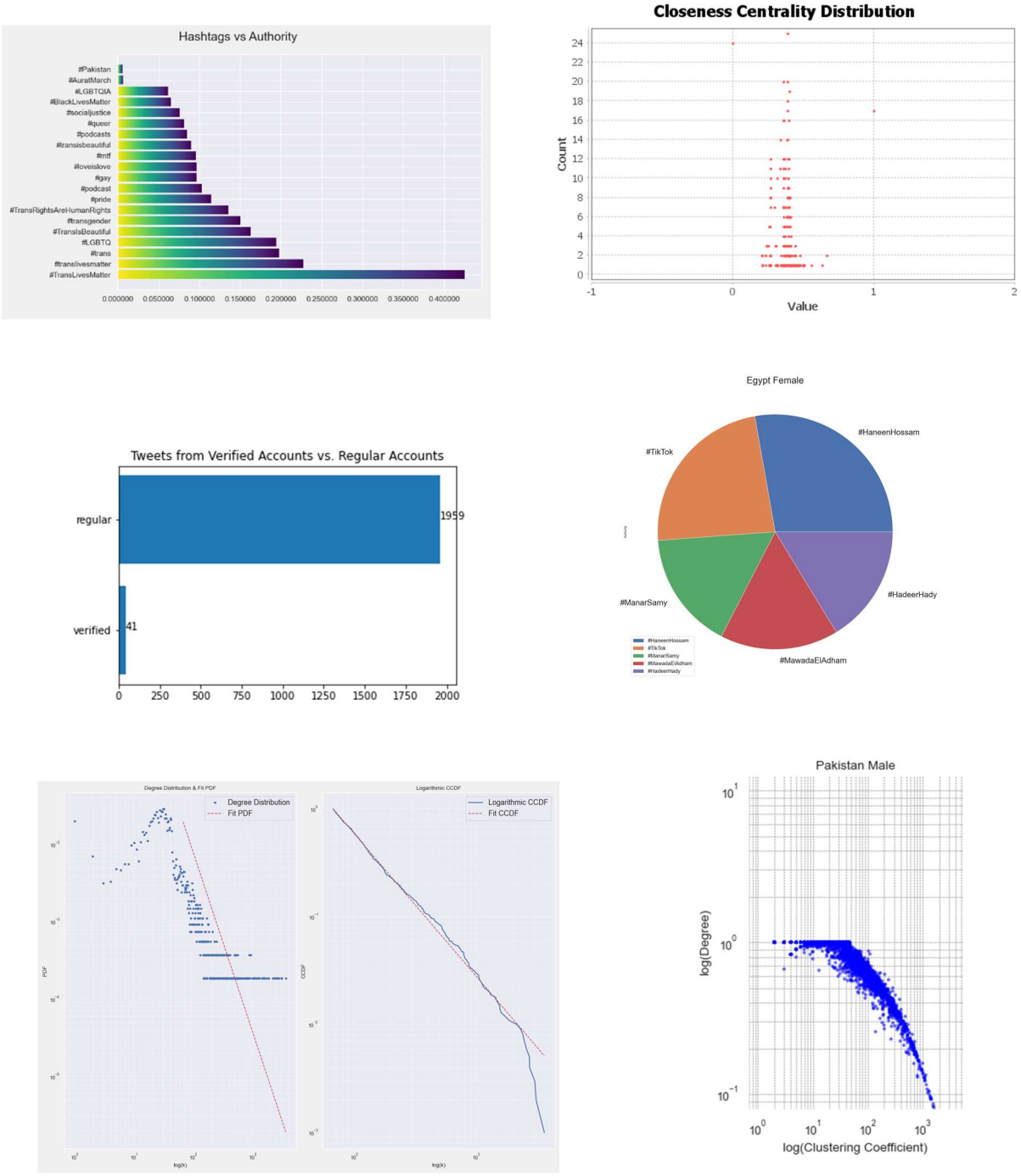


Figure 38: Various depicted metrics throughout the course of the study

5. Utilizing Python

```
import matplotlib.pyplot as plt
import matplotlib as mpl
import numpy as np
import pandas as pd
import seaborn as sns
sns.set()
df = pd.read_csv('SON_egypt_male_hashtag_trans_table.csv',sep=',',encoding="utf-8")
column_names =  list(df.columns)
column_names = column_names[3:]
for i in column_names:
    if i == 'closnesscentrality':
        title_variable = 'Closness Centrality'
    elif i == 'harmonicclosnesscentrality':
        title_variable = 'Harmonic Closness Centrality'
    elif i == 'betweenesscentrality':
        title_variable = 'Betweeness Centrality'
    elif i == 'modularity_class':
        title_variable = 'Modularity Class'
    elif i == 'pageranks':
        title_variable = 'PageRank'
    else:
        title_variable = i
    selected_variable = i
    df = df.sort_values(selected_variable,ascending = False).head(20)
    index = df['Id'] # Variables
    values = df[selected_variable]
    plot_title = 'Hashtags vs ' + title_variable
    title_size = 18
    x_label = title_variable
    filename = selected_variable
    fig, ax = plt.subplots(figsize=(10,6), facecolor=(.94, .94, .94))
    mpl.pyplot.viridis()
    bar = ax.barh(index, values)
    plt.tight_layout()
    if i == 'modularity_class':
        ax.xaxis.set_major_formatter(mpl.ticker.StrMethodFormatter('{x:,.0f}'))
    elif i == 'Degree':
        ax.xaxis.set_major_formatter(mpl.ticker.StrMethodFormatter('{x:,.0f}'))
    elif i == 'Weighted Degree':
        ax.xaxis.set_major_formatter(mpl.ticker.StrMethodFormatter('{x:,.0f}'))
    elif i == 'Eccentricity':
        ax.xaxis.set_major_formatter(mpl.ticker.StrMethodFormatter('{x:,.0f}'))
    elif i == 'betweenesscentrality':
        ax.xaxis.set_major_formatter(mpl.ticker.StrMethodFormatter('{x:,.3f}'))
    else:
        ax.xaxis.set_major_formatter(mpl.ticker.StrMethodFormatter('{x:,.6f}'))
    title = plt.title(plot_title, pad=20, fontsize=title_size)
    title.set_position([.33, 1])
    plt.subplots_adjust(top=0.9, bottom=0.1)
    ax.grid(zorder=0)
    def gradientbars(bars):
        grad = np.atleast_2d(np.linspace(0,1,256))
        ax = bars[0].axes
        lim = ax.get_xlim()+ax.get_ylim()
        for bar in bars:
            bar.set_zorder(1)
            bar.set_facecolor('none')
            x,y = bar.get_xy()
            w, h = bar.get_width(), bar.get_height()
            ax.imshow(grad, extent=[x+w, x, y, y+h], aspect='auto', zorder=1)
        ax.axis(lim)
    gradientbars(bar)
```

Figure 39: Python code for plotting horizontal bar graphs

```
import matplotlib.pyplot as plt
import matplotlib as mpl
import numpy as np
import pandas as pd
import seaborn as sns
from collections import Counter
import powerlaw
import scipy
sns.set()
df = pd.read_csv('complete_pakistan_trans_table.csv',sep=',',encoding="utf-8")
df
# Obtain PDF and CDF
data = df['Degree'].values
sorted_data = sorted(data)
freq = Counter(sorted_data)
k = list(Counter(freq).keys())
pk = list(Counter(freq).values()) #PDF
pk = pk/np.sum(pk)
Pk = 1 - np.cumsum(pk) # CDF
Pk[-1] = 1
Pk = sorted(Pk, reverse = True)

data2 = data[data != 0] # Take out all zeros if exists
fit = powerlaw.Fit(data2,discrete=True) # fit PDF line function from power law library

print('gamma:',fit.power_law.alpha)

fig = plt.figure(figsize=(16, 12), dpi=80)
ax1 = fig.add_subplot(121)
ax2 = fig.add_subplot(122)
ax1.title.set_text('Degree Distribution & Fit PDF')
ax2.title.set_text('Logarithmic CCDF')

ax1.loglog(k, pk, 'o', markersize = 4,label='Degree Distribution') # Degree dist. scatter
ax1.grid(which='both', linestyle='--', linewidth=0.5) # Fit PDF line
ax1.set_xlabel("log(k)")
ax1.set_ylabel("PDF")

fit.power_law.plot_pdf(color='r', linestyle='--', ax=ax1,label='Fit PDF')
ax1.legend(fontsize=16)

fit.plot_ccdf(color='b', linewidth=2, ax=ax2,label="Logarithmic CCDF") #CCDF plot
fit.power_law.plot_ccdf(color='r', linestyle='--', ax=ax2,label='Fit CCDF') #Fit CCDF line
ax2.grid(which='both', linestyle='--', linewidth=0.5)
ax2.set_xlabel("log(k)")
ax2.set_ylabel("CCDF")
ax2.legend(fontsize=16)
plt.tight_layout()
plt.show()
plt.savefig('pakistan_trans'+'_complete'+'.png', facecolor=(.94, .94, .94))
```

Figure 40: Python code for calculating gamma value and plotting PDF & CCDF plots

```
df = pd.read_csv('complete_pakistan_female_table.csv',sep=',',encoding="utf-8")
df2 = pd.read_csv('complete_pakistan_male_table.csv',sep=',',encoding="utf-8")
df3 = pd.read_csv('complete_pakistan_trans_table.csv',sep=',',encoding="utf-8")
df4 = pd.read_csv('complete_egypt_female_table.csv',sep=',',encoding="utf-8")
df5 = pd.read_csv('complete_egypt_male_table.csv',sep=',',encoding="utf-8")
df6 = pd.read_csv('complete_egypt_trans_table.csv',sep=',',encoding="utf-8")

fig = plt.figure(figsize=(16, 12), dpi=80)
ax1 = fig.add_subplot(2,3,1)
ax2 = fig.add_subplot(2,3,2)
ax3 = fig.add_subplot(2,3,3)
ax4 = fig.add_subplot(2,3,4)
ax5 = fig.add_subplot(2,3,5)
ax6 = fig.add_subplot(2,3,6)
ax1.title.set_text('Pakistan Female')
ax2.title.set_text('Pakistan Male')
ax3.title.set_text('Pakistan TransGender')
ax4.title.set_text('Egypt Female')
ax5.title.set_text('Egypt Male')
ax6.title.set_text('Egypt TransGender')
ax1.scatter(df['Degree'].values ,df['Clustering Coefficient'].values , c='blue', alpha=0.6,
s=10,edgecolors='none')
ax1.set_yscale('log')
ax1.set_xscale('log')
ax1.set_xlabel("log(Clustering Coefficient)")
ax1.set_ylabel("log(Degree)")
#----
ax2.scatter(df2['Degree'].values ,df2['Clustering Coefficient'].values , c='blue', alpha=0.6,
s=10,edgecolors='none')
ax2.set_yscale('log')
ax2.set_xscale('log')
ax2.set_xlabel("log(Clustering Coefficient)")
ax2.set_ylabel("log(Degree)")
#----
ax3.scatter(df3['Degree'].values ,df3['Clustering Coefficient'].values , c='blue', alpha=0.6,
s=10,edgecolors='none')
ax3.set_yscale('log')
ax3.set_xscale('log')
ax3.set_xlabel("log(Clustering Coefficient)")
ax3.set_ylabel("log(Degree)")
#----
ax4.scatter(df4['Degree'].values ,df4['Clustering Coefficient'].values , c='blue', alpha=0.6,
s=10,edgecolors='none')
ax4.set_yscale('log')
ax4.set_xscale('log')
ax4.set_xlabel("log(Clustering Coefficient)")
ax4.set_ylabel("log(Degree)")
#----
ax5.scatter(df5['Degree'].values ,df5['Clustering Coefficient'].values , c='blue', alpha=0.6,
s=10,edgecolors='none')
ax5.set_yscale('log')
ax5.set_xscale('log')
ax5.set_xlabel("log(Clustering Coefficient)")
ax5.set_ylabel("log(Degree)")
#----
ax6.scatter(df6['Degree'].values ,df6['Clustering Coefficient'].values , c='blue', alpha=0.6,
s=10,edgecolors='none')
ax6.set_yscale('log')
ax6.set_xscale('log')
ax6.set_xlabel("log(Clustering Coefficient)")
ax6.set_ylabel("log(Degree)")
plt.tight_layout()
plt.show()
```

Figure 41: Python code for plotting various metrics on log scale

In [ ]:

```
import os

os.environ['TOKEN'] = 'your token here'
```

In [ ]:

```
from google.colab import drive

drive.mount('/content/drive')
```

Import libraries

In [ ]:

```
import requests
import pandas as pd
import time
import re
import string
import ast
```

If you need to download a library, use the following code, just specify the name of the library you need (here we downloaded emoji library)

In [ ]:

```
!pip install emoji
```

Set up headers

In [ ]:

```
def create_headers(bearer_token):
    headers = {"Authorization": "Bearer {}".format(bearer_token)}
    return headers
```

In [ ]:

```
headers = create_headers(os.environ['TOKEN'])
```

Download data

Date format and other parameter explanations available here:

https://developer.twitter.com/en/docs/twitter-api/premium/search-api/api-reference/premium-search#DataParameters

In [ ]:

```
def create_url(keyword, start_date, end_date, env_label, endpoint="fullarchive"):
    search_url =
"https://api.twitter.com/1.1/tweets/search/{}.json".format(endpoint + "/" +
env_label)

    #change params based on the endpoint you are using
    query_params = {'query': keyword, 'fromDate': start_date, 'toDate': end_date}
    return (search_url, query_params)
```

In [ ]:

```
def connect_to_endpoint(url, headers, params, next_token=None):
    if next_token is not None and next_token != '':
        params['next'] = next_token
    response = requests.request("GET", url, headers=headers, params=params)
    if response.status_code != 200:
```

```
        raise Exception(response.status_code, response.text)
    return response.json()
```

In [ ]:

```
def get_data(keyword, start_time, end_time, next_token, env_label, endpoint):
    results = []
    dCounter = 0
    while next_token is not None and dCounter < 10:  #change dCounter to 10 to
retrieve 1k data
        ##this part here for one request
        url = create_url(keyword, start_time, end_time, env_label, endpoint)
        json_response = connect_to_endpoint(url[0], headers, url[1], next_token)

        if "results" in json_response:
            results.extend(json_response["results"])
        ### up until this point
        if "next" in json_response:
            next_token = json_response["next"]
            dCounter += 1
        else:
            next_token = None
        time.sleep(1)

    return results
```

In [ ]:

```
def get_single_response(keyword, start_time, end_time, env_label, endpoint):
    #endpoint can be fullarchive or 30day
    results = []
    url = create_url(keyword, start_time, end_time, env_label, endpoint)
    json_response = connect_to_endpoint(url[0], headers, url[1])

    if "results" in json_response:
        results.extend(json_response["results"])

    return results
```

In [ ]:

```
#if you are using the 30day endpoint, make sure you specify dates that are within
30day range!
#To get small data of size "100" use, get_single_response
#change the hashtags to your liking
# with "get_data" we are querying for 1k data - see the definition of get_data func
above
tweets = get_data("(gulpanra OR JusticeforGulPanra OR StopGenocideOfTransgenders OR
TransLivesMatter) lang:en",
                  "202001010000", "202012310000", "", "NSdev", "fullarchive")
#tweets = get_single_response("(EndTransViolence OR BeelaCrisis OR JusticeForBijlee
OR JusticeForToffi OR JusticeforGulPanra OR TransLivesMatter) lang:en",
"202001010000", "202112310000","NSdev", "fullarchive")
```

Inspect data

In [ ]:

```
len(tweets)
```

In [ ]:

```
tweets[0]
```

First, we want to convert the data into Pandas DataFrame. This format enables us easy manipulation of the data as well as saving/loading data.

Since we have our tweets saved as a list of dictionaries, we can easily convert it to DataFrame by executing the cell blow.

In [ ]:

```
tweets_df = pd.DataFrame(tweets)
```

In [ ]:

```
tweets_df
```

Saving the results

Once we have our Tweets as a DataFrame it is a good idea to save it on the disk.

Be mindful of the fact that the storage of a Colab notebook is deleted everytime runtime is interrupted or restarted, so you need to manually download it to your computer or mount your Google Drive and save it there (this option is unavailable if you're using university's email account for Drive).

In [ ]:

```
path = "Desktop\network proje"  #enter the path to your Drive or leave this as default
```

We can save it as a comma-separated values file, which enables opening it in a spreadsheet editor and inspecting it.

In [ ]:

```
tweets_df.to_csv(path + "trans_case1.csv", index=False)
```

In order to preserve datatypes, we should save it as a parquet or pickle file.

In [ ]:

```
tweets_df.to_pickle(path + "trans_case1.pkl")
```

Loading the data

If you want to load the results you have previously saved, simply execute the next code, specifying the path to the file.

You will need to either upload it to the Colab workspace or copy the path to the file on Drive.

In [ ]:

```
tweets_df = pd.read_pickle("trans_case1.pkl")
```

In [ ]:

```
tweets_df
```

Preprocessing the data

In our dataframe we have the entire Tweet object. Some columns that might be of particular interest to us are:

- created_at - date when Tweet was posted
- id/id_str - unique Tweet identifiers
- text - the content of the Tweet
- user - information about the user who posted the Tweet
- retweeted_status - information about the original Tweet
- quote/reply/retweet/favorite count - Tweet metrics
- entities - hashtags, urls, user_mentions present in Tweet

We can filter the dataframe and keep only columns we are interested in. You can pick which columns you'd like to keep and put them int the column_list below.

In [ ]:

```
#Everything is filtered below in Umut's code, no need to do it now

#tweets_filtered = tweets_df.copy() #it's a good idea to work on the copy of
original dataframe, so we can always go back to it if we mess something up
#column_list = ["created_at", "id_str", "text", "user", "retweeted_status",
"quote_count", "reply_count", "retweet_count", "favorite_count", "entities"]
#tweets_filtered = tweets_filtered[column_list]
# tweets_filtered
```

In [ ]:

```
import enum


class CountMode(enum.Enum):
    count_within_all_tweets = 1
    count_within_verified_tweets = 2
    count_within_regular_tweets = 3
```

In [ ]:

```
def preProcessData(df):
    for index, row in df.iterrows():
        # get full_text and entities(hashtags mentions etc.) of the tweet
        if row['truncated']:
            tweet = {
                "full_text": ast.literal_eval(row['extended_tweet'])['full_text'],
                "entities": ast.literal_eval(row['extended_tweet'])['entities']
            }
        else:
            # if the tweet is retweeted, this status will not be na
            if ('retweeted_status' in row) and (not
pd.isna(row['retweeted_status'])):
                originalTweet = ast.literal_eval(row['retweeted_status'])
                userTagName = originalTweet['user']['screen_name']

                # if the text is truncated
                if originalTweet['truncated']:
                    tweet = {
                        "full_text": "RT @" + userTagName + ": " +
originalTweet['extended_tweet']['full_text'],
                        "entities": originalTweet['extended_tweet']['entities']
                    }
                else:
                    tweet = {
                        "full_text": "RT @" + userTagName + ": " +
originalTweet['text'],
                        "entities": originalTweet['entities']
                    }

                del originalTweet
                del userTagName

            # if the tweet is original and not truncated
            else:
```

```
                tweet = {
                    "full_text": row['text'],
                    "entities": ast.literal_eval(row['entities'])
                }

        # get Username & User Tag & Location & Verification status of account
        tweet["username"] = ast.literal_eval(row['user'])['name']
        tweet["screen_name"] = ast.literal_eval(row['user'])['screen_name']
        tweet["location"] = ast.literal_eval(row['user'])['location']
        tweet["is_verified"] = ast.literal_eval(row['user'])['verified']

        # get Quote & Fav & RT & Reply counts
        tweet["quote_count"] = row['quote_count']
        tweet["favorite_count"] = row['quote_count']
        tweet["retweet_count"] = row['retweet_count']
        tweet["reply_count"] = row['reply_count']

        # get Tweet creation date&time
        tweet["created_at"] = row['created_at']

        # get TweetID
        tweet["tweet_id"] = row['id']

        tweets.append(tweet)

        del tweet
        del row
        del index


def countNumberOfRetweetedTweets(tweets, countMode):
    # Separate RTs and original tweets
    for tweet in tweets:
        if bool(re.match(r'(RT @[\w]+:)', tweet["full_text"])):
            if countMode == CountMode.count_within_all_tweets:
                rts.append(tweet)
            elif countMode == CountMode.count_within_verified_tweets:
                verified_rts.append(tweet)
            elif countMode == CountMode.count_within_regular_tweets:
                regular_rts.append(tweet)
        else:
            if countMode == CountMode.count_within_all_tweets:
                non_rt.append(tweet)
            elif countMode == CountMode.count_within_verified_tweets:
                verified_non_rt.append(tweet)
            elif countMode == CountMode.count_within_regular_tweets:
                regular_non_rt.append(tweet)
        del tweet


# Distinguish tweets according to their account verification status
def countTweetsFromVerifiedAccounts(tweets):
    for tweet in tweets:
        if tweet["is_verified"]:
            tweets_from_verified_accounts.append(tweet)
        else:
            tweets_from_regular_accounts.append(tweet)
```

```
        del tweet


# filter locations according to city names given in listOfCities
def filterLocations(tweets, listOfCities):
    filtered_tweets = []

    for tweet in tweets:
        for index, city in listOfCities.iterrows():
            if city['city'] in tweet['location']:
                filtered_tweets.append(tweet)
                break
    return filtered_tweets
```

In [ ]:

```
import ast
import re
import pandas as pd
df1 = pd.read_csv('/Users/Selen/Desktop/nsproject/network
proje/hashtags/Pakistan_male/pakistan_male_case1.csv')
df2 = pd.read_csv('/Users/Selen/Desktop/nsproject/network
proje/hashtags/Pakistan_male/pakistan_male_case2.csv')
df3 = pd.read_csv('/Users/Selen/Desktop/nsproject/network
proje/hashtags/Pakistan_male/pakistan_male_case3.csv')



#df3 = pd.read_csv('../Pakistan_male/pakistan_male_case3.csv')
tweets = []
tweets_from_verified_accounts = []
tweets_from_regular_accounts = []

rts = []
non_rt = []
verified_rts = []
regular_rts = []
verified_non_rt = []
regular_non_rt = []

#pakistanCities = pd.read_csv('pakistanCities.csv')
#egyptCities = pd.read_csv('egyptCities.csv')
#dfs = [df1, df2 ,df3]

# define and add the concatenated data as vars into below statement.
dfs = pd.concat([df1,df2,df3], ignore_index=True, sort=False)

#index column was not incrementing from 0 through 3000 for 3k data. So we resetted
it to have an index column of 0-2999
dfs.reset_index(drop=True, inplace=True)

df_final = dfs.copy()  # to work on a copy, copied the original df
preProcessData(df_final)  # full text, column filtering are applied to the df

# Iterates over the entire dataset
countTweetsFromVerifiedAccounts(tweets)  # count of Tweets From Verified Accounts
```

```
# total number of retweeted all tweets
countNumberOfRetweetedTweets(tweets, CountMode.count_within_all_tweets)

# total number of retweeted only verified tweets
countNumberOfRetweetedTweets(tweets_from_verified_accounts,
CountMode.count_within_verified_tweets)

# total number of retweeted only regular tweets
countNumberOfRetweetedTweets(tweets_from_regular_accounts,
CountMode.count_within_regular_tweets)
# df_final
```

In [ ]:

```
# DataFrame of whole network and in the last line, it saved it as a csv file
tweets_df = pd.DataFrame(tweets)
tweets_df
tweets_df.to_csv("SON_pakistan_male.csv", index=False)  # change the name according
to your case
```

In [ ]:

```
tweets_filtered=tweets_df
```

In [ ]:

```
tweets_from_verified_accounts_df = pd.DataFrame(tweets_from_verified_accounts)
tweets_from_verified_accounts_df
```

In [ ]:

```
# DataFrame of tweets from regular accounts
tweets_from_regular_accounts_df = pd.DataFrame(tweets_from_regular_accounts)
tweets_from_regular_accounts_df
```

In [ ]:

```
# retweeted tweets
len(rts)  # length of the rt tweets
rts_df = pd.DataFrame(rts)
rts_df
```

In [ ]:

```
# not retweeted tweets, basically the remaining tweets
len(non_rt)  # length of not rt tweets
original_tweets_df = pd.DataFrame(non_rt)
original_tweets_df
```

In [ ]:

```
#GRAPH
#THIS GRAPHS WILL BE USED FOR EXPLANATION OF WHY WE CHOOSE TO USE THE COMPLETE
DATASET.
#WILL CONTAIN 4 GRAPH:
#1) TWEETS FROM VERIFIED ACCOUNTS VS. REGULAR ACCOUNTS
#2) ORIGINAL TWEETS VS. RETWEETS
#3) ORIGINAL TWEETS VS. RETWEETS FROM VERIFIED ACCOUNTS
#4) ORIGINAL TWEETS VS. RETWEETS FROM REGULAR ACCOUNTS


import matplotlib.pyplot as plt

#GRAPH 1
```

```
regular=len(tweets_from_regular_accounts_df)
verified=len(tweets_from_verified_accounts_df)

x = ['verified','regular']
y= [verified,regular]

plt.barh(x, y)

for index, value in enumerate(y):
    plt.text(value, index,
             str(value))
plt.title('Tweets from Verified Accounts vs. Regular Accounts')
plt.savefig('SON_pakistan_male_verified_vs_regular_accounts.png')
plt.show()


#GRAPH 2
original=len(original_tweets_df)
retweeted=len(rts_df)
x = ['original','retweets']
y= [original,retweeted]

plt.barh(x, y)

for index, value in enumerate(y):
    plt.text(value, index,
             str(value))
plt.title('Original Tweets vs. Retweets')
plt.savefig('SON_pakistan_male_original_vs_retweet.png')
plt.show()
```

Extracting words/hashtags

There are many ways to build networks from the data we download from Twitter.

One possibility is to have a bipartite network of Tweets and words/hashtags and then observe word, hashtag or word-hashtag projections.

Extracting words

In order to extract words, we first need to clean the Tweet text. This way we will remove punctuation, hashtags/mentions/urls (they are preserved in the entity column anyway). We will also turn all letters to lowercase.

You can also consider removing stopwords, removing words that are not in the english language corpora, lematizing the words, etc. I suggest you research nltk library and its possibilities.

In [ ]:

```
import re
import string
```

In [ ]:

```
def cleaner(tweet):
    tweet = re.sub("@[A-Za-z0-9]+", "", tweet)  # remove mentions
    tweet = re.sub("#[A-Za-z0-9]+", "", tweet)  # remove hashtags
    tweet = re.sub(r"(?:\@|http?\://|https?\://|www)\S+", "", tweet)  # remove http
links
    tweet = " ".join(tweet.split())
    tweet = str.lower(tweet)  #to lowercase
```

```
    table = str.maketrans(dict.fromkeys(string.punctuation))
    tweet = tweet.translate(table)  # remove punctuation
    return tweet
```

In [ ]:

```
tweets_filtered["clean_text"] = tweets_df["full_text"].map(cleaner)

# 3 different usages
#by_changing_to_df = pd.DataFrame(verified_non_rt)
#output1 = by_changing_to_df["full_text"].map(cleaner)

#output2 = cleaner(tweets[0]["full_text"])
#print(tweets[0]["full_text"])
#print(output2)
#output1
```

We are going to loop through the dataframe and then through the words in the clean text. We are going to add the words as keys to dictionary and use their frequencies as values.

In [ ]:

```
#initialize an empty dict
tweets_filtered["clean_text"] = tweets_df["full_text"].map(cleaner)


unique_words = {}
for row in tweets_filtered.clean_text:
    for word in row.split(" "):
        #if the word is encountered for the first time add to dict as key and set
its value to 0
        unique_words.setdefault(word, 0)
        #increase the value (i.e the count) of the word by 1 every time it is
encountered
        unique_words[word] += 1
```

In [ ]:

```
#clean_text is in tweets_filtered
tweets_filtered
```

In [ ]:

```
#remove empty word
unique_words.pop("")
#remove word 'rt'
unique_words.pop("rt")
```

We can inspect the words as a dataframe.

You can always save this dataframe as .csv for future reference.

In [ ]:

```
#In the last line, it saves word count df as csv file. It's better to save it so
that maybe we'll use it later for the presentation
uw_df = pd.DataFrame.from_dict(unique_words, orient='index').reset_index()
print(uw_df)
uw_df.rename(columns={'index': 'Word', 0: 'Count'}, inplace=True)
uw_df.sort_values(by=['Count'], ascending=False, inplace=True)
uw_df
uw_df.to_csv("SON_pakistan_male_full_network_word_count.csv", index=False)  #
change the name to your liking
```

Extracting the hashtags

We are going to loop through the dataframe and then through the hashtags in the entities. We are going to add the hashtags as keys to dictionary and use their frequencies as values. At the same time, we are going to save them in a list and add them to a separate column to facilitate our future work.

In [ ]:

```
# no need to run this cell. We already have screen_name column by this point

"""
#TASK-1

sc_name = {}
tweets_filtered["screen_name"] = ""

for idx, row in tweets_filtered.iterrows():
  screen_name_list = []
  for user_mentions in row["entities"]["user_mentions"]:
    sc_name.setdefault(user_mentions['screen_name'], 0)
    sc_name[user_mentions["screen_name"]] += 1
    screen_name_list.append(user_mentions["screen_name"])
  tweets_filtered.at[idx,"screen_names"] = screen_name_list
sc_name

sc_df = pd.DataFrame.from_dict(sc_name, orient='index').reset_index()
sc_df.rename(columns = {'index':'Username', 0:'Count'}, inplace=True)
sc_df.sort_values(by=['Count'], ascending=False, inplace=True)
sc_df
#END OF TASK-1

"""
```

In [ ]:

```
unique_hashtags = {}

df_to_check = pd.DataFrame(tweets)
# exp_df = verified_non_rt
df_to_check["hashtags"] = ""

for idx, row in df_to_check.iterrows():
    hashtag_list = []
    for hashtag in row["entities"]["hashtags"]:
        unique_hashtags.setdefault("#" + hashtag["text"], 0)
        unique_hashtags['#' + hashtag["text"]] += 1
        hashtag_list.append(hashtag["text"])
    df_to_check.at[idx, "hashtags"] = hashtag_list

unique_hashtags
```

In [ ]:

```
uh_df = pd.DataFrame.from_dict(unique_hashtags, orient='index').reset_index()
uh_df.rename(columns={'index': 'Hashtag', 0: 'Count'}, inplace=True)
uh_df.sort_values(by=['Count'], ascending=False, inplace=True)
```

In [ ]:

```
#In the last line, it saves hashtag count df as csv file. It's better to save it so
that maybe we'll use it later for the presentation
uh_df
```

```
uh_df.to_csv("SON_pakistan_male_full_network_hashtag_count.csv", index=False)  # change the name to your liking
```

In [ ]:

```
uh_df
```

Building the network

We are going to use the networkx library, which is a Python library that enables network science analysis of the data.

We are going to use it to create our network and extract edgelist from it, since we can easily import it to Gephi (a software we are going to see in visualization labs).

However, it offers implemented algorithms for analysis (for example PageRank) that you can use out-of-box to analyze your network.

But first, we will loop through our dataframe and connect words and hashtags if they appear together in the same Tweet.

In [ ]:

```
import itertools
import networkx as nx
```

In [ ]:

```
uh = unique_hashtags.keys()
uw = unique_words.keys()
```

In [ ]:

```
uw
```

In [ ]:

```
# No need to run this cell

"""
#SELEN
sc=sc_name.keys()
sc

"""
```

In [ ]:

```
df_to_check
```

In [ ]:

```
#adding hashtags column from df_to_check to tweets_filtered
hashtags=df_to_check["hashtags"]
hashtags
tweets_filtered["hashtags"]=hashtags
```

In [ ]:

```
tweets_filtered
```

In [ ]:

```
#It creates pairs from all words.
# From this cell on, we will create 3 different networks:
# Network1 = words+hashtags
#Network2 = words only
#Network3 = hashtags only
```

```
#This cell is for Network1
network = {}
network_key = 0
for index, row in tweets_filtered.iterrows():
    #hashtags extracted from Tweet do not have the # sign in front of them but we
will add it to differentiate hashtags from words
    combined_list = ['#' + hashtag for hashtag in row["hashtags"] if '#' + hashtag
in unique_hashtags] + [word for word in str.split(row["clean_text"], " ") if word
in uw]
    #itertool product creates Cartesian product of each element in the combined
list
    for pair in itertools.product(combined_list, combined_list):
        #exclude self-loops and count each pair only once because our graph is
undirected and we do not take self-loops into account
        if pair[0] != pair[1] and not (pair[::-1] in network):
            network.setdefault(pair, 0)
            network[pair] += 1
network_df = pd.DataFrame.from_dict(network, orient="index")
network_df
```

In [ ]:

```
#This cell is for Network2
# NETWORK OF ONLY WORDS
networkWords = {}
networkWords_key = 0
for index, row in tweets_filtered.iterrows():
    word_list = [word for word in str.split(row["clean_text"], " ") if word in uw]
    #itertool product creates Cartesian product of each element in the word list
    for pair in itertools.product(word_list, word_list):
        #exclude self-loops and count each pair only once because our graph is
undirected and we do not take self-loops into account
        if pair[0] != pair[1] and not (pair[::-1] in networkWords):
            networkWords.setdefault(pair, 0)
            networkWords[pair] += 1
networkWords_df = pd.DataFrame.from_dict(networkWords, orient="index")
networkWords_df
```

In [ ]:

```
#This cell is for Network3
# NETWORK OF ONLY HASHTAGS
networkHashtags = {}
networkHashtags_key = 0
for index, row in tweets_filtered.iterrows():
    #hashtags extracted from Tweet do not have the # sign in front of them but we
will add it to differentiate hashtags from words
    hashtag_list = ['#' + hashtag for hashtag in row["hashtags"] if '#' + hashtag
in unique_hashtags]
    #itertool product creates Cartesian product of each element in the word list
    for pair in itertools.product(hashtag_list, hashtag_list):
        #exclude self-loops and count each pair only once because our graph is
undirected and we do not take self-loops into account
        if pair[0] != pair[1] and not (pair[::-1] in networkHashtags):
            networkHashtags.setdefault(pair, 0)
            networkHashtags[pair] += 1
networkHashtags_df = pd.DataFrame.from_dict(networkHashtags, orient="index")
networkHashtags_df
```

In [ ]:

```
#This cell is for Network1
network_df.reset_index(inplace=True)
network_df.columns = ["pair", "weight"]
network_df.sort_values(by="weight", inplace=True, ascending=False)
network_df
```

In [ ]:

```
#This cell is for Network2
# FOR WORD NETWORK
networkWords_df.reset_index(inplace=True)
networkWords_df.columns = ["pair", "weight"]
networkWords_df.sort_values(by="weight", inplace=True, ascending=False)
networkWords_df
```

In [ ]:

```
#This cell is for Network3
# FOR HASHTAG NETWORK
networkHashtags_df.reset_index(inplace=True)
networkHashtags_df.columns = ["pair", "weight"]
networkHashtags_df.sort_values(by="weight", inplace=True, ascending=False)
networkHashtags_df
```

In [ ]:

```
#This cell is for Network1
#to get weighted graph we need a list of 3-element tuplels (u,v,w) where u and v
are nodes and w is a number representing weight
up_weighted = []
for edge in network:
    #we can filter edges by weight by uncommenting the next line and setting
desired weight threshold
    #if(network[edge])>1:
    up_weighted.append((edge[0], edge[1], network[edge]))

G = nx.Graph()
G.add_weighted_edges_from(up_weighted)
```

In [ ]:

```
#This cell is for Network2 & Network3
# list of 3-element tuples for word and hastag networks
up_weighted_words = []
for edge in networkWords:
    #we can filter edges by weight by uncommenting the next line and setting
desired weight threshold
    #if(network[edge])>1:
    up_weighted_words.append((edge[0], edge[1], networkWords[edge]))

G_words = nx.Graph()
G_words.add_weighted_edges_from(up_weighted_words)  # words graph

#to get weighted graph we need a list of 3-element tuplels (u,v,w) where u and v
are nodes and w is a number representing weight
up_weighted_hashtags = []
for edge in networkHashtags:
    #we can filter edges by weight by uncommenting the next line and setting
desired weight threshold
    #if(network[edge])>1:
    up_weighted_hashtags.append((edge[0], edge[1], networkHashtags[edge]))
```

```
G_hashtags = nx.Graph()
G_hashtags.add_weighted_edges_from(up_weighted_hashtags)  # hashtags graph
```

In [ ]:

```
# No need to run this cell, we'll use Gephi for visualization anyways
"""
import matplotlib.pyplot as plt
nx.draw(G, with_labels=True, node_size=1.5, alpha=0.3, arrows=True)
plt.show()
nx.draw(G, with_labels=True)
plt.show()

### ploting words network
nx.draw(G_words, with_labels=True, node_size=1.5, alpha=0.3, arrows=True)
plt.show()
nx.draw(G_words, with_labels=True)
plt.show()

### ploting hashtags network
nx.draw(G_hashtags, with_labels=True, node_size=1.5, alpha=0.3, arrows=True)
plt.show()
nx.draw(G_hashtags, with_labels=True)
plt.show()

"""
```

In [ ]:

```
# This cell is for Network1
# WORDS&HASHTAGS NETWORK NODES&EDGES
print(len(G.nodes()))  # nodes=each word in the texts
print(len(G.edges()))  # edges=pairs for each word
#G.edges
#G.nodes
```

In [ ]:

```
# This cell is for Network2 & Network3

#WORDS NETWORK NODES&EDGES
print(len(G_words.nodes()))
print(len(G_words.edges()))
#HASHTAGS NETWORK NODES&EDGES
print(len(G_hashtags.nodes()))
print(len(G_hashtags.edges()))
```

In [ ]:

```
#In this cell, we rank the nodes using Pagerank function and print the top20 nodes
with the highest pageranks

from collections import Counter

pr = nx.pagerank(G, alpha=0.9)  # we ranked the nodes of Network1 with pagerank

# Let's print the top-20 pairs
c = Counter(pr)
top_20_network = c.most_common(20)
print(top_20_network)
```

```
print("********************")

pr_word = nx.pagerank(G_words, alpha=0.9)  # we ranked the nodes of Network2 with
pagerank

# Let's print the top-20 pairs
c = Counter(pr_word)
top_20_word = c.most_common(20)
print(top_20_word)
print("********************")

pr_hashtag = nx.pagerank(G_hashtags, alpha=0.9)  # we ranked the nodes of Network3
with pagerank

# Let's print the top-20 pairs
c = Counter(pr_hashtag)
top_20_hashtag = c.most_common(20)
print(top_20_hashtag)
```

SAVE EDGELIST

In [ ]:

```
# Now we're going to save edgelists and nodelists to use for Gephi later
# Change the names according to your case
filename1 = "./SON_pakistan_male_network_edgelist_trans.csv"
filename2 = "./SON_pakistan_male_word_edgelist_trans.csv"
filename3 = "./SON_pakistan_male_hashtag_edgelist_trans.csv"
```

In [ ]:

```
nx.write_weighted_edgelist(G, filename1, delimiter=",")  # Graph name must be
changed!!
nx.write_weighted_edgelist(G_words, filename2, delimiter=",")  # Graph name must be
changed!!
nx.write_weighted_edgelist(G_hashtags, filename3, delimiter=",")  # Graph name must
be changed!!
```

In [ ]:

```
# If this cell doesn't work in your device, no worries you can skip it
#add header with appropriate column names (works on collab and Linux/Mac(?))
sed -i.bak 1i"Source,Target,Weight"./ SON_egypt_trans_network_edgelist_trans.csv  #
Graph name must be changed!!
sed -i.bak 1i"Source,Target,Weight"./ SON_egypt_trans_word_edgelist_trans.csv  #
Graph name must be changed!!
sed -i.bak 1i"Source,Target,Weight"./ SON_egypt_trans_hashtag_edgelist_trans.csv  #
Graph name must be changed!!
```

Create Node List

In [ ]:

```
# This creates a csv file of nodes for Network2
word_nodes = pd.DataFrame.from_dict(unique_words, orient="index")
word_nodes.reset_index(inplace=True)
word_nodes["Label"] = word_nodes["index"]
word_nodes.rename(columns={"index": "Id", 0: "delete"}, inplace=True)
word_nodes = word_nodes.drop(columns=['delete'])

word_nodes
word_nodes.to_csv("SON_pakistan_male_word_nodelist.csv", index=False)
```

In [ ]:

```
# This creates a csv file of nodes for Network3
hashtag_nodes = uh_df.copy()
hashtag_nodes["Label"] = hashtag_nodes["Hashtag"]
hashtag_nodes.rename(columns={"Hashtag": "Id"}, inplace=True)
hashtag_nodes = hashtag_nodes.drop(columns=['Count'])
hashtag_nodes
hashtag_nodes.to_csv("SON_pakistan_male_hashtag_nodelist.csv", index=False)
```

SAVE NODELIST

In [ ]:

```
# This joins the two nodelists above and creates a csv file of nodes for Network1
nodelist = hashtag_nodes.append(word_nodes, ignore_index=True)
nodelist
nodelist.to_csv("SON_pakistan_male_network_nodelist.csv", index=False)
```

Figure 42: Data retrieval code

```
import pandas as pd
import re
import string
import ast
import enum


class CountMode(enum.Enum):
    count_within_all_tweets = 1
    count_within_verified_tweets = 2
    count_within_regular_tweets = 3


def preProcessData(df):
    for index, row in df.iterrows():
        # get full_text and entities(hashtags mentions etc.) of the tweet
        if row['truncated']:
            tweet = {
                "full_text": ast.literal_eval(row['extended_tweet'])['full_text'],
                "entities": ast.literal_eval(row['extended_tweet'])['entities']
            }
        else:
            # if the tweet is retweeted, this status will not be na
            if ('retweeted_status' in row) and (not
pd.isna(row['retweeted_status'])):
                originalTweet = ast.literal_eval(row['retweeted_status'])
                userTagName = originalTweet['user']['screen_name']

                # if the text is truncated
                if originalTweet['truncated']:
                    tweet = {
                        "full_text": "RT @" + userTagName + ": " +
originalTweet['extended_tweet']['full_text'],
                        "entities": originalTweet['extended_tweet']['entities']
                    }
                else:
                    tweet = {
```

```
                        "full_text": "RT @" + userTagName + ": " +
originalTweet['text'],
                        "entities": originalTweet['entities']
                    }

                del originalTweet
                del userTagName

            # if the tweet is original and not truncated
            else:
                tweet = {
                    "full_text": row['text'],
                    "entities": ast.literal_eval(row['entities'])
                }

        # get Username & User Tag & Location & Verification status of account
        tweet["username"] = ast.literal_eval(row['user'])['name']
        tweet["screen_name"] = ast.literal_eval(row['user'])['screen_name']
        tweet["location"] = ast.literal_eval(row['user'])['location']
        tweet["is_verified"] = ast.literal_eval(row['user'])['verified']

        # get Quote & Fav & RT & Reply counts
        tweet["quote_count"] = row['quote_count']
        tweet["favorite_count"] = row['quote_count']
        tweet["retweet_count"] = row['retweet_count']
        tweet["reply_count"] = row['reply_count']

        # get Tweet creation date&time
        tweet["created_at"] = row['created_at']

        # get TweetID
        tweet["tweet_id"] = row['id']
        # if not pd.isna(tweet['location']):
        tweets.append(tweet)

        del tweet
        del row
        del index


def countNumberOfRetweetedTweets(tweets, countMode):
    # Separate RTs and original tweets
    for tweet in tweets:
        if bool(re.match(r'(RT @[\w]+:)', tweet["full_text"])):
            if countMode == CountMode.count_within_all_tweets:
                rts.append(tweet)
            elif countMode == CountMode.count_within_verified_tweets:
                verified_rts.append(tweet)
            elif countMode == CountMode.count_within_regular_tweets:
                regular_rts.append(tweet)
        else:
            if countMode == CountMode.count_within_all_tweets:
                non_rt.append(tweet)
            elif countMode == CountMode.count_within_verified_tweets:
                verified_non_rt.append(tweet)
            elif countMode == CountMode.count_within_regular_tweets:
                regular_non_rt.append(tweet)
```

```
        del tweet


# Distinguish tweets according to their account verification status
def countTweetsFromVerifiedAccounts(tweets):
    for tweet in tweets:
        if tweet["is_verified"]:
            tweets_from_verified_accounts.append(tweet)
        else:
            tweets_from_regular_accounts.append(tweet)

        del tweet


# filter locations according to city names given in listOfCities
def filterLocations(tweets, listOfCities):
    filtered_tweets = []

    for tweet in tweets:
        for index, city in listOfCities.iterrows():
            if city['city'] in tweet['location']:
                filtered_tweets.append(tweet)
                break
    return filtered_tweets


# Main

df = pd.read_csv('../raw_data/justicefornaseembibi.csv')
df1 = pd.read_csv('../raw_data/justiceforgulpanra.csv')
df2 = pd.read_csv('../raw_data/justiceforsaima.csv')
df3 = pd.read_csv('../raw_data/justicefornoor.csv')
df4 = pd.read_csv('../raw_data/savewomenofpakistan.csv')

pakistanCities = pd.read_csv('../pakistanCities.csv')
egyptCities = pd.read_csv('../egyptCities.csv')

dfs = [df, df1, df2, df3, df4]

tweets = []
tweets_from_verified_accounts = []
tweets_from_regular_accounts = []

rts = []
non_rt = []
verified_rts = []
regular_rts = []
verified_non_rt = []
regular_non_rt = []

# preprocess all the given hashtags and build
for df_instance in dfs:
    preProcessData(df_instance)

    del df_instance

countTweetsFromVerifiedAccounts(tweets)
```

```
# total number of retweeted all tweets
countNumberOfRetweetedTweets(tweets, CountMode.count_within_all_tweets)

# total number of retweeted only verified tweets
countNumberOfRetweetedTweets(tweets_from_verified_accounts,
CountMode.count_within_verified_tweets)

# total number of retweeted only regular tweets
countNumberOfRetweetedTweets(tweets_from_regular_accounts,
CountMode.count_within_regular_tweets)

original_tweets_from_pakistan = filterLocations(non_rt, pakistanCities)
all_tweets_from_pakistan = filterLocations(tweets, pakistanCities)
```

Figure 43: Data manipulation