Network analysis of tweet's sentiment

Salvatore Romano[†], Alberto Zancanaro[‡], Enrico Lanza[‡], Carlo Facchin[‡],

***Abstract*—In this paper, we perform a network analysis of the tweets related to different international public figures, combining quantitative analysis of the adjacency list made with Python and qualitative analysis of the network visualization made with Gephi. We focus on the corpus' sentiment, performing Robustness and Assortativity based on single words' sentiment score and Page Rank nodes removal related to a global sentiment score. Starting from that analysis, we create network visualization highlighting networks' words positivity and negativity, and clusters to better understand the discussions around specific twitter accounts qualitatively. We aimed to create a new mixed methodology, ables to consider words corpus' sentiment both during the quantitative and qualitative analysis.**

We will show interesting differences in network's negativity across different types of social actors, and exciting results on the discussion about Kobe Bryan's death (happened the day before the data collection) seen from different actors' followers.

***Index Terms*—Network Analysis, Sentiment Analysis, Semantic Network, Nodes Removal, Assortativity, Robustness, Twitter.**

I. Introduction

Big databases made available by the increased use of social networks allows researchers to analyze huge amount of data. Those database are mostly composed of images and text. The most accessible API (among commercials social network) is surely Twitter, therefore analyze large corpus of tweets is a common challenge in the computer science's literature. The corpus' sentiment is one of the most useful information that can be extracted.

There are different levels of automated procedures that can be used to perform this task: from a close reading of each tweet applying a manual categorization of the sentiment, till machine learning algorithms categorizing huge numbers of tweets in few seconds. The more qualitative analysis fail to use the full potential of the database, requiring huge amount of time and forces in the manual categorization, the more automated procedures struggles to create algorithms able to understand properly the context of the words and the sentiment related, especially analyzing sort text like tweets. We build a methodology to use quantitative analysis based on network's statistics, as a means to highlight the part of the corpus more central in the network, that we analyze later qualitatively.

II. Methodology

A. Work Description

On the 27/01/2020 we retrieved 3000 contents from various iconic figure. To avoid the problems related to the syntactical structure of the tweets, that often are too short to be analyzed, we considered all the re-tweets/posts/comments containing a public figures' tag as an unique corpus of single words. Analyzing contents referring to a specific Twitter user altogether, we want to make assumptions about the sentiment of the general discussions around a specific social actor.

Once we retrieve the content we clean the corpus of word and we perform the sentiment analysis of each word. With the clean corpus we create the network of word and subsequently perform the various network analysis.

B. Code Implementation

All the code was created with the idea that can be reused in future by any user that want perform similar research

The download of the tweets was performed through Tweepy, a python library to access the Twitter API. Once obtained the key from Twitter this library could automatically download a maximum of 3200 element[1] for request. We choose to download 3000 elements for each person.

After the request the tweets were downloaded, they will store in a string and clean. The clean procedure consist in:

- Remove of URL.
- A lower case transformation.
- Delete of stopwords and words with no sentiment score[2].
- Remove of word formed by single character.
- Remove of tweets formed by empty line or single word.
- Lemmization.

Lemmization and stopwords removal are performed through NLTK (Natural Language Toolkit, a python library to work with human language data).

Subsequently the tweets were saved in a txt file for further elaboration. If in that file there are past tweets the new ones and the old ones were merged together to obtain a bigger corpus of tweet. This was done to eventually overcome the maximum number of elements retrieved with a single request. A simple function also delete duplicate tweets.

With all tweets saved in a txt file the code perform the sentiment analysis of the corpus of all tweets downloaded. The sentiment was performed through the NLTK implementation of SentiWordNet a lexical resource explicitly devised for supporting sentiment classification and opinionmining applications. SentiWordNet provide a Positive, Negative and Neutral score for each word. Each score can be between 0 and 1 and their sum is equal to 1. Also each word can have different set of score based on the different meaning of that

[†]Author one affiliation, email: salvatore.romano.3@studenti.unipd.it
[‡]Author two affiliation, email: alberto.zancanaro.1@studenti.unipd.it
[†]Author three affiliation, email: enrico.lanza.2@studenti.unipd.it
[†]Author four affiliation, email: carlo.faccin@studenti.unipd.it

[1]Element can be post, retweet, tweet and comment

[2]Some special word like politician names or country names were maintained and we will assign theme a neutral score in future.

word[3]. So for each word the code assign a positive, negative and neutral score based on this formula:

$$S_{pos,fin} = \frac{\sum_{i=1}^{n} S_{pos,i}}{T_{score}} \quad (1)$$

$$S_{neg,fin} = \frac{\sum_{i=1}^{n} S_{neg,i}}{T_{score}} \quad (2)$$

$$S_{neu,fin} = \frac{\sum_{i=1}^{n} S_{neu,i}}{T_{score}} \quad (3)$$

$$T_{score} = \sum_{i=1}^{n} (S_{pos,i} + S_{neg,i} + S_{neu,i}) \quad (4)$$

Where *S*, *pos*, *neg* and *neu* stand for Score, positive, negative and neutral. This formula also provide that $S_{pos,fin} + S_{neg,fin} + S_{neu,fin} = 1$.

After that the code assign a mean score to each word. The code provide 2 type of Mean Score (MS_{V1} and MS_{V2}[4], each one with different management of neutral score) and a third type that can be create by the user with a lambda function. The code also normalize the 3 mean score. The first two Mean Score are evaluated through this formulas:

$$MS_{V1} = \begin{cases} (1 - S_{neu}) * \sqrt{S_{pos}^2 + S_{neg}^2} & \text{if } S_{pos} > S_{neg} \\ (S_{neu} - 1) * \sqrt{S_{pos}^2 + S_{neg}^2} & \text{if } S_{pos} < S_{neg} \\ 0 & \text{if } S_{pos} = S_{neg} \end{cases}$$

$$MS_{V2} = \begin{cases} S_{pos} & \text{if } S_{pos} > S_{neg} \text{ and } S_{neu} < S_{neu,th} \\ -S_{neg} & \text{if } S_{pos} < S_{neg} \text{ and } S_{neu} < S_{neu,th} \\ 0 & \text{if } S_{neu} \geq S_{neu,th} \end{cases}$$

where $S_{neu,th}$ is the threshold score for a word to be consider neutral (i.e. all the words with at least that neutral score will be consider neutral). Since a lot of words have a very high score we set $S_{neu,th} = 1$ so only the word that are completely neutral are consider neutral.

Once the mean scores are evaluated we created the CSV file of the node. Each node will have its ID (that correspond with the word) and a set of parameter (positive score, negative score, neutral score, mean scores, mean scores normalized, frequency of the word)

The last step is to create the adjacency list and store it in a CSV file. The code implementation provide this possibility to create a link between a word and the n word at its left and at its right. The n can span between 1 and all the word of the tweet[5]. We create 3 adjacency list respectively for $n = 1$, $n = 2$ and for all the word of the tweet.

Subsequently from the CSV file of the nodes and the CSV file of the edges we create a networkx object and perform the various analysis. Networkx is a python library for network analysis.

[3]For example *club* can be both a Noun of a Verb and for each one can has different score

[4]V1 and V2 stands for version 1 and versione 2.

[5]In this case the n is evaluated in automatic for each tweet.

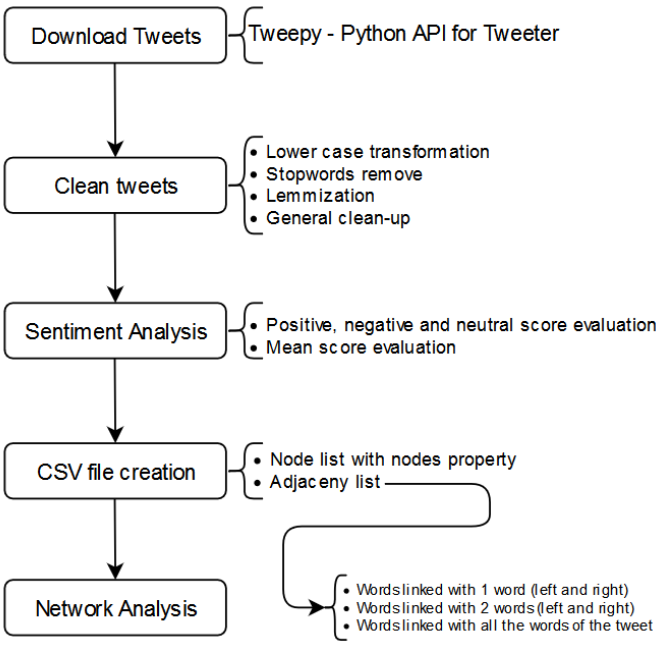


Fig. 1: Scheme of the work

All the steps in code are summarized in Figure 1.

Since a lot of word are generally neutral, during the network analysis, we implemented a script to contextualize neutral words and assign them a positive or negative score. At each word we assign the mean of the MS_{V2} of its neighbour node, based on the assumption that if a neutral word is linked to negative word is probably used in a negative context. In this way we can emphasize the results of the various analysis.

III. Network Model Properties

For analyzing the model of this type of semantic networks is possible to make a simple comparison between the parameters of the three networks of a single user, in which, as explained before, are used different types of edges connection (.1/.2/.all). Similar results about the network general properties are obtained for all the account analyzed. The table (1) summarizes the parameters of the Trump networks while the three images (2), (3), (4) represent their PDF and CCDF functions, plotted in a log-log scale.
In addiction the figure (5) shows the three different adjacency matrices of the Trump networks.

All the networks follow a scale-free behaviour ($2 < \gamma < 3$) and as expected the the power law exponent coefficient decrease as the number of edges increase.
The average degree $\langle k \rangle$ confirm the previous aspect as it increase linearly with more edges to the network. This coefficient show also two different structures that the network can assume. Infact only the first network is supercritical ($1 < \langle k \rangle < \ln N$) while the others two assume a connected structure ($\langle k \rangle > \ln N$) with all the words almost surely connected to the giant component.
From the degree distributions is also possible to detect the

	Trump.1	Trump.2	Trump.all
Number of nodes	2824	2824	2824
Number of edges	8325	14777	28015
Average degree $\langle k \rangle$	5.896	10.465	19.841
k_{max}	204	333	617
k_{min}	1	1	1
γ	2.379	2.249	2.06
Diameter	15	9	6
Average path lenght	3.96	3.28	2.81

TABLE 1: Parameters of the three different Trump networks

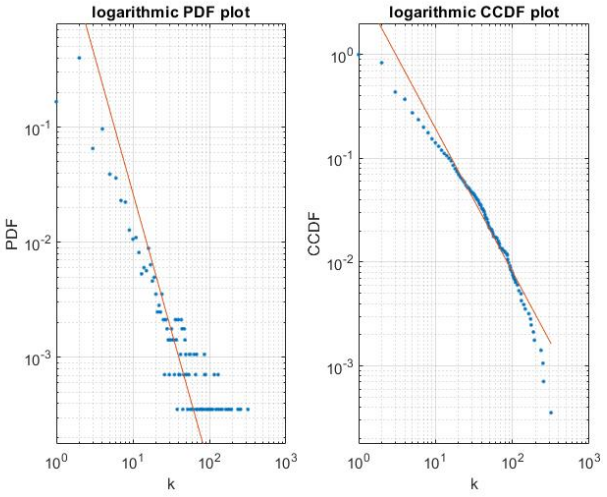


Fig. 2: Trump.1

presence of hubs in the three network, due to the fact that they all are heavy-tailed. In addiction on the third network, that has significantly more edges, is possible to see more edges to the large component of the nodes that previously were less connected.
In the analyzed network the node with the highest degree is always 'Trump', with an increasing value as the type of network change.

An important concept to understand is that the distribution and the model of the semantic networks considered does not depend by the number of nodes of the networks. In fact, as we previously said, similar results are obtain for all the users we analyze. An example are the law exponent coefficients γ of the networks generated by the comments retrieved under the posts of Alexandria Ocasio-Cortez. The three γ are respectively: 2.673 , 2.369 , 2.16 and indicate that the graph is always a scale-free network in all three cases, exact like Trump. But in this case the network has a lot more nodes (table 2. We can conclude that famous persons yield to the same type of network.

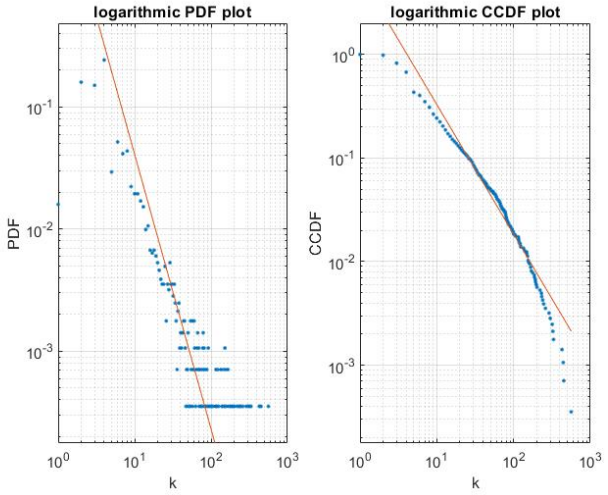


Fig. 3: Trump.2

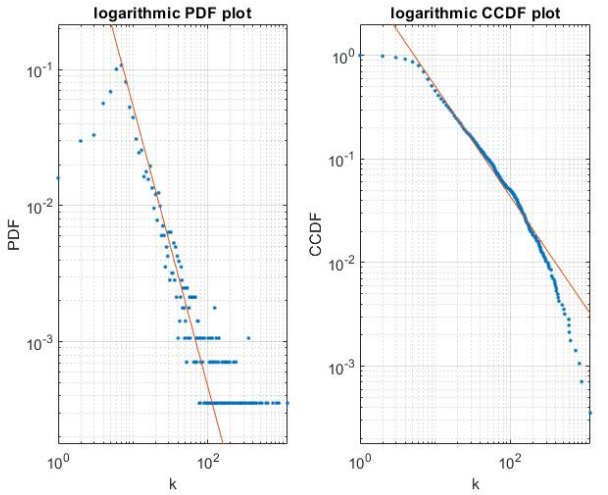


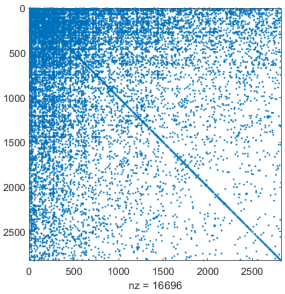
Fig. 4: Trump.all

User	N.nodes	User	N.nodes
@justinbieber	1451	@kobebryant	1606
@AOC (Cortez)	3484	@elonmusk	2925
@Cristiano (cr7)	1540	@NASA	3064
@KimKardashian	1890	@BarackObama	2186
@GretaThumberg	3090	@Mike Pence	3075
@guyverhofstadt	3033	@Pontifex	2965
@BorisJohnson	3124	@realDonaldTrump	2824

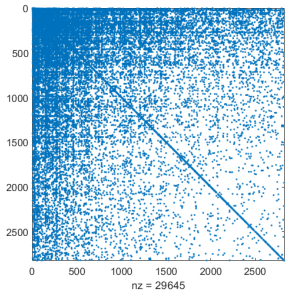
TABLE 2: Number of nodes in networks of different users

IV. Robustness

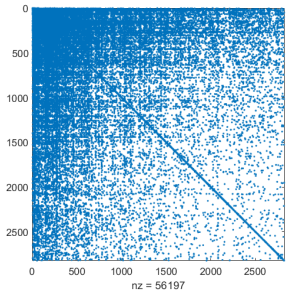
The robustness analysis is focused on the difference that emerge in the network if we removed only the negative node. Our thesis is that the importance of the negative word in the structure of the various networks changed based on person. This analysis was performed on the network created by linking every word with only the word at its left and its right.



(a) Trump.1

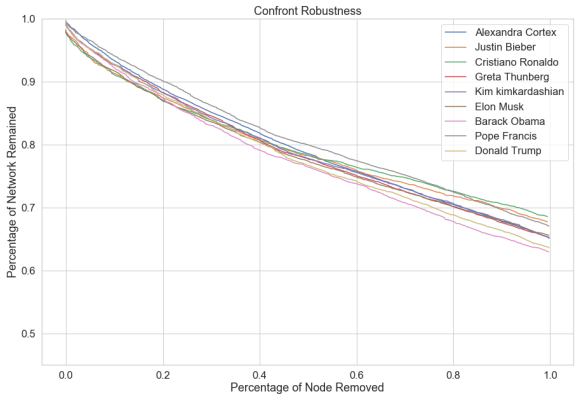


(b) Trump.2

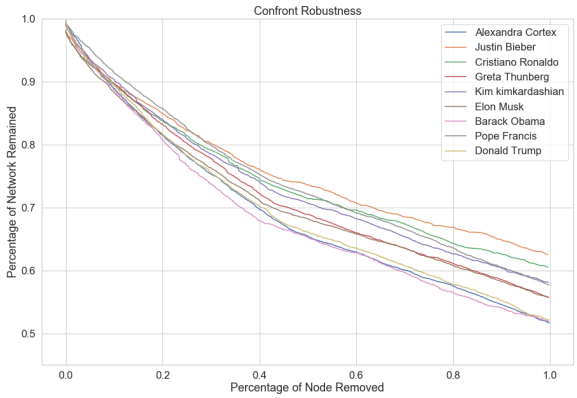


(c) Trump.all

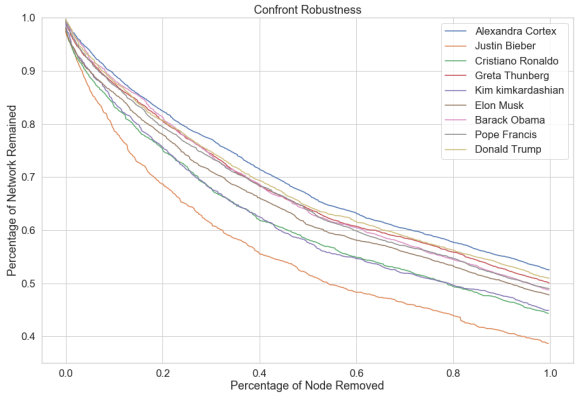
Fig. 5: Projection Matrices



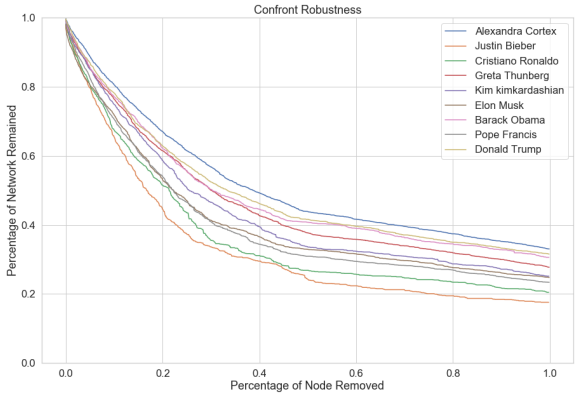
(a) Robustness of original networks to negative node removal



(b) Robustness of Polarized networks to negative node removal



(c) Robustness of original networks to positive node removal



(d) Robustness of Polarized networks to positive node removal

Fig. 6: Robustness of the networks

Person	N_1	N_2	N_3
Alexandra Cortex	1800	4	3
Justin Bieber	907	3	2
Cristiano Ronaldo	932	3	2
Greta Thunberg	1721	4	3
Kim Kimkardashian	1097	4	3
Elon Musk	1629	3	3
Barack Obama	1134	5	4
Pope Francis	1710	4	3
Donald Trump	1471	3	3

TABLE 3: Dimension of the 3 biggest component after the node removal

The analysis consist in remove the all the negative node and see at each step how big is the largest component respect the original network. So at each step we have the percentage of the original network remain.

The order or removal is based on the frequency of the word in the tweet corpus i.e. we first remove the nodes that appear more frequently in all the tweets.

From the plot in figure 6a and figure 6b we see that removing all the negative words we can classify the people in various category (this can be easily seen in the polarized network):

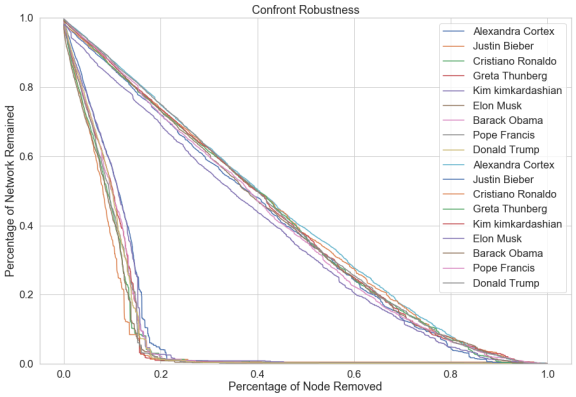
- Politicians (Trump, Obama, Cortez) tend to have a bigger portion of the network destroyed; this means that the followers of this category of people are more inclined to use negative words.
- Public but someway particular personage (Elon Musk, Greta Thunberg) have networks were negative words are less important but still present.
- Very famous people (Kim Kimkardashian, Cristiano Ronaldo, Justin Bieber) that have networks were negative word are less important. This means that in their network the negative word are generraly less important and less linked to other word. Pope Francis also falls into this category.

We also perform the inverse operation, i.e. remove all the positive node and observe the behavior of the networks. The results are shown in figure 6c and figure 6d and confirm what we previously discussed.

For example the networks of the politicians are less affected by the removing of the positive words (around the 30% of the original network remain for both Obama, Cortez and Trump). Instead the networks of Justin Bieber and Cristiano Ronaldo are severely damaged by this operation (respectively only the 17% and the 20% of the original networks remain).

In table 3 we also shown the number of node in the 3 biggest component at the end of the negative node removal. We can immediately notice that the removal don't have created various sub-networks that can be easily remerged but instead has left all the node scattered.

We also perform a random removal of all node and a target removal of all node. The results are shown in figure 7a. The



(a) Robustness removing all the node

upper lines correspond to random removal and the lower lines to precise removal. We can see that the behavior is the same as that of a scale free network. In fact the random removal cause a linear destruction of the networks instead with precise removal the integrity drops drastically in a short time. This confirm the assumption of III that the networks have a scale-free behavior.

The results shown are based on the network created by linking the world with only its neighbor word but similar figures are obtained with the other two type of networks.

V. Assortativity

An important property of a network is its assortativity, that is the preference for a network's nodes to attach to others that are similar in some way. For instance, in the real life, celebrity couples represent a highly visible proof of these phenomenon. In fact social networks hubs (famous people) tend to know, date and marry each other due to their similarity.
This propriety is not present in all the networks. For example if we consider the protein interaction network of yeast, the hubs tend to avoid linking to each other and they link instead to many small-degree nodes.
The different behavior of the nodes allows to divide the networks in three categories:

- **Neutral Networks:** networks with random wiring
- **Assortative Networks:** networks with hubs that tend to link to each other and to avoid linking to small-degree nodes. At the same time the small-degree nodes tend to connect to other small-degree nodes
- **Diassortative Networks:** networks with hubs that avoid each other and link to small-degree nodes

Assortativity is a propriety that describes how nodes are correlated. A way to quantify the degree correlation between nodes is by examining for each node i, with degree k_i, the average degree of its neighbors:

$$k_{nn}(k_i) = \sum_{k'} k' P(k'|k_i)$$

where $P(k'|k_i)$ is the conditional probability that following a link of a k_i-degree node we reach a degree-k' node. For weighted networks, like in our case, an analogous measure can be defined:

$$k^w_{nn}(k_i) = \frac{1}{s_i} \sum_{j \in N(i)} w_{ij} k_j$$

where s_i is the weighted degree of node i, w_{ij} is the weight of the edge that links i and j and $N(i)$ are the neighbors of node i.
From the trend of the average neighbor degree of each node, it is possible to analyze the degree correlation between nodes. In fact if $k_{nn}(k_i)$ is constant the network is neutral, if it increases the network is assortative, instead if it decreases the network is diassortative.

In our analysis we performe two kind of assortativity: degree assortativity and sentiment assortativity. The first one describes the degree correlation between nodes. It indicates if words that have a large number of links tend to be connected with other frequent words. Instead the second one quantifies the correlation regarding the sentiment score of each nodes. In the latter case we want to verify if positive words are more likely to link with positive words and vice versa. The scale of the sentiment values, that we used, goes from 0 to 1, where 1 is the most positive score and 0 is the most negative one. If a word is near to 0.5 is considered as neutral.
In both the assortativity analyses we quantify the correlation for three different cases. In the first case we consider the whole network and all the the nodes are analyzed. Instead in the second and third cases just the positive and negative nodes are analyzed respectively.
Since most of the words has a neutral sentiment score, by taking into account just the positive or the negative nodes, it is possible to study the local behavior of the nodes. With negative and positive score we mean a score between 0 and 0.3 and between 0.7 and 1 respectively.

In our work we perform the degree and the sentiment assortativity analyses on data set of tweets related to accounts of famous people. For each different person we visualize the average degree of the neighbors of each node and, separately, of positive and negative nodes. Then we display the same information regarding, instead, the sentiment score. The assortativity propriety in each different case is evaluated by the so called "assortativity factor", that can have values from -1 to 1. If the "assortativity factor" is close to 0 the network is neutral. Instead if it is very positive ([0.1,1]) or very negative ([-1,-0.1]) the network is assortative or diassortative respectively. In our analysis we use six kind of "assortativity factor", according to the fact that we are considering the degree or the sentiment or we are analyzing all the nodes or just the positive or negative ones.
The following outcomes are given by the analysis of Donald Trump, Barak Obama, Cristiano Ronaldo, Justin Bieber's networks: Where:

	Donald Trump	Barak Obama	Cristiano Ronaldo	Justin Bieber
DAF	0.04	0.08	0.05	0.04
PDAF	0.03	0.1	0.05	0.06
NDAF	0.04	0.1	0.08	0.04
SAF	0.11	0.156	0.22	0.23
PSAF	-0.19	-0.27	0.18	-0.09
NSAF	-0.04	-0.13	0.05	0.09

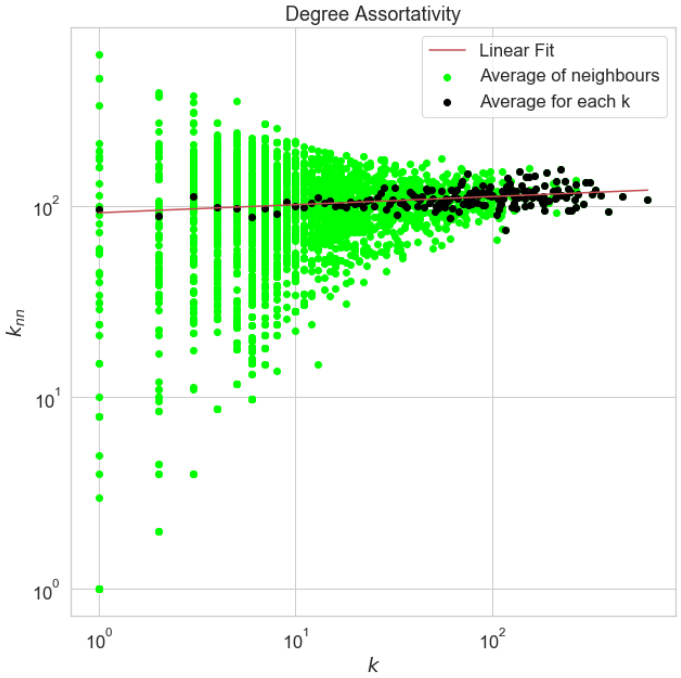


Fig. 8: Donal Trump's degree analysis

- **DAF** (Degree Assortativiy Factor): degree assortativity factor for all the nodes
- **PDAF** (Positive Degree Assortativiy Factor: degree assortativity factor for positive words
- **NDAF** (Negative Degree Assortativiy Factor): degree assortativity factor for negative words
- **SAF** (Sentiment Assortativiy Factor): sentiment assortativity factor for all the nodes
- **PSAF** (Positive Sentiment Assortativiy Factor): sentiment assortativity factor for positive words
- **NSAF** (Negative Sentiment Assortativiy Factor): sentiment assortativity factor for negative words

As we can see from the previous table, the degree assortativity factor is close to zero whether we analyze all the nodes or we select just the positive or the negative nodes. So, regarding the degree, all the networks are neutral and present random wiring. The following charts show this common behavior between Trump and Justin Bieber's networks: The results of the sentiment analysis are more meaningful. Indeed all the networks analyzed appear assortative. This means that hubs, that are the most positive words, tend to be linked with each other. In the same time the most negative words tend to be connected.
The sentiment assortativity of the networks, by considering

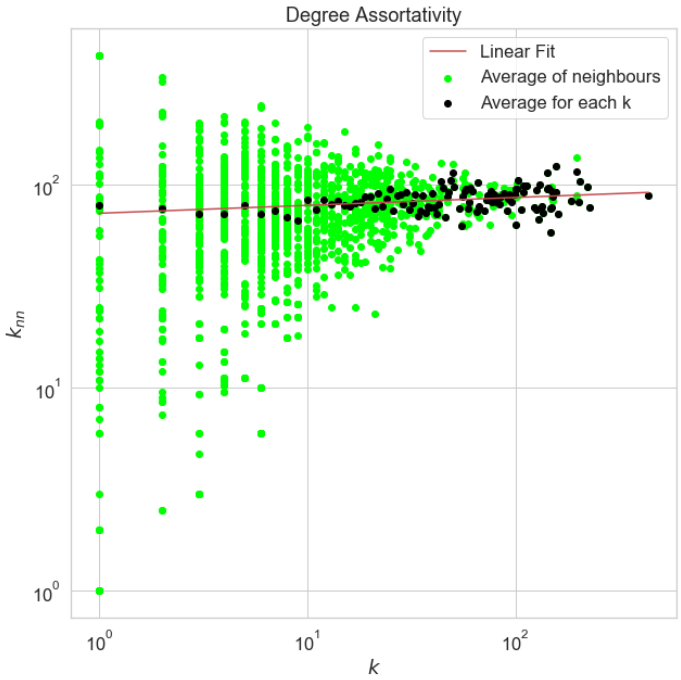


Fig. 9: Justin Bieber's degree analysis

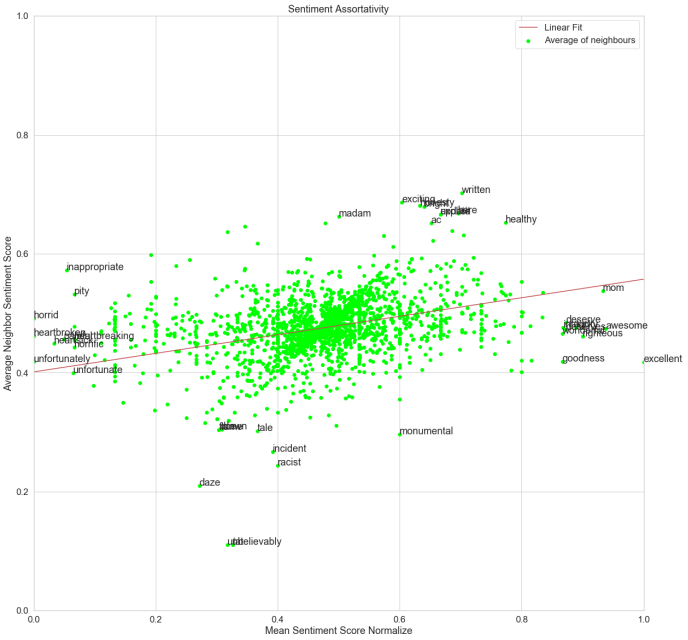


Fig. 11: Barak Obama's sentiment analysis

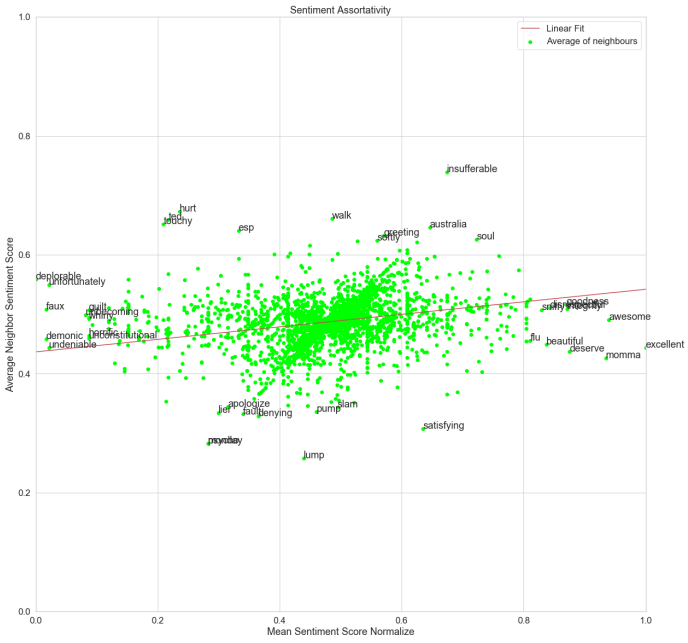


Fig. 10: Donal Trump's sentiment analysis

Fig. 12: Cristiano Ronaldo's sentiment analysis

all the nodes, can be seen by observing the trend of the function k_{nn}: If instead we consider just the positive or the negative words the network could have a different behavior locally. Donald Trump and Obama's network have meaningful outcomes in these two cases. In fact both are diassortative for negative and positive nodes. This means that the positive words tend to be linked with negative ones and vice versa.

The trends of the function k_{nn} for positive and negative nodes for Barak Obama are: In conclusion we can say that the degree correlation between the words in tweets is neutral for all the networks. This means that hub and low-degree nodes are randomly linked.

Instead if we compute the sentiment correlation between the words, we find that the network are generally assortative. This means that the positive words tend to link with other positive words and the negative ones with other negative words. Moreover the local sentiment assortativity could be different from the assortativity of the whole network. In fact even if Donal Trump and Barak Obama's network are

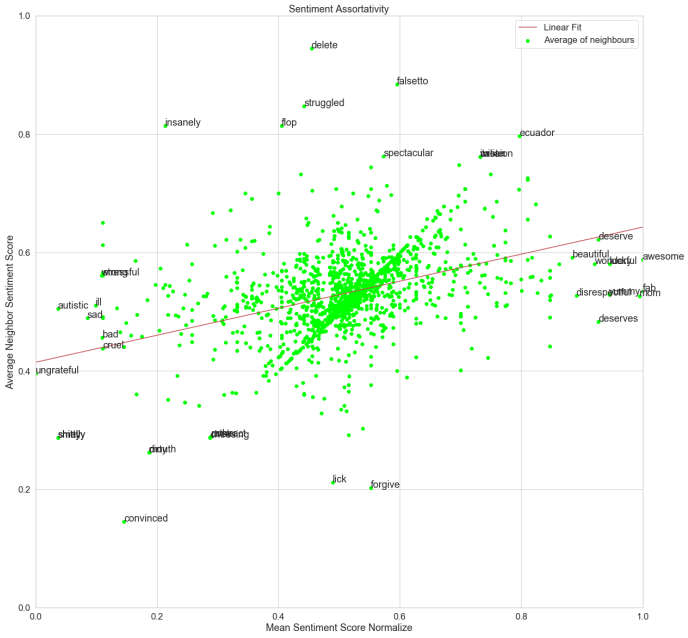


Fig. 13: Justin Bieber's sentiment analysis

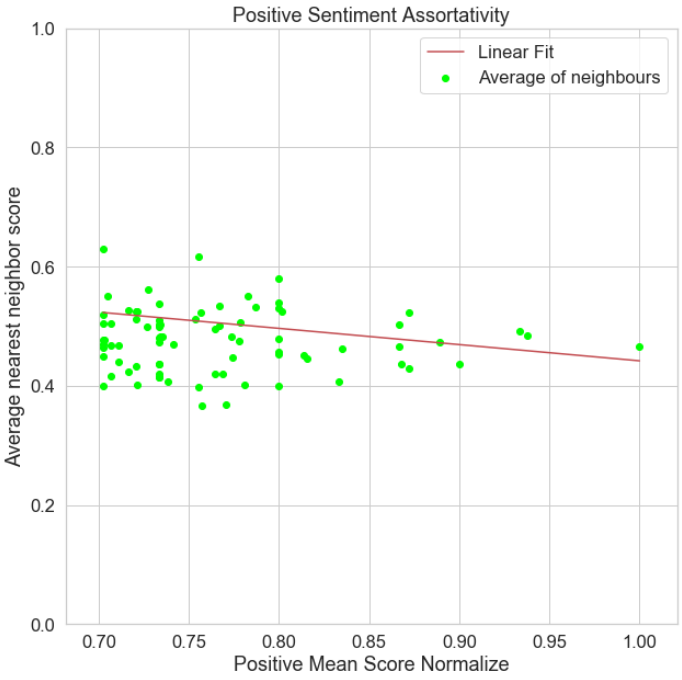


Fig. 14: Barak Obama's positive sentiment analysis

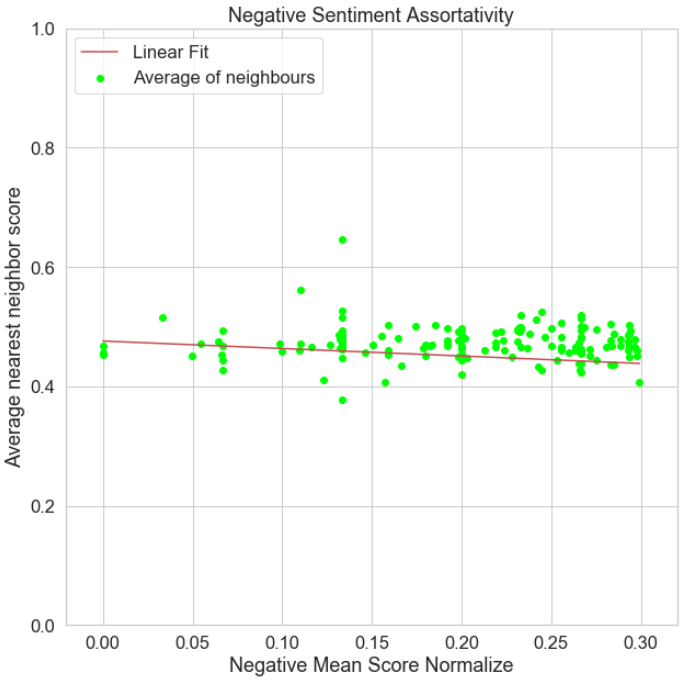


Fig. 15: Barak Obama's negative sentiment analysis

assortative, they are diassortative locally.

VI. Page rank (Carlo)

The analysis of the sentiment networks can be improved also using a pagerank evaluation of the nodes. In order to measure the relative importance of words, we used the PageRank method (16, 17).

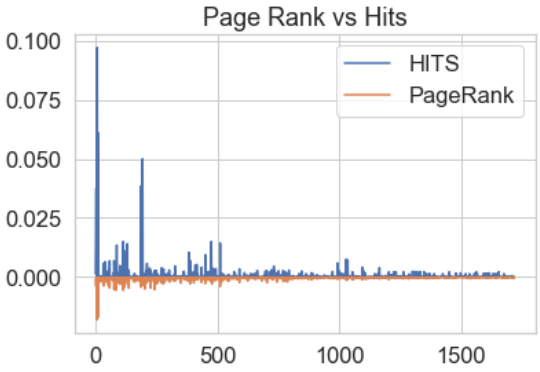


Fig. 16: Page-Rank VS HITS

This type of analysis infact can identify the most important words in the semantic dictionary created by the comments, assigning a smaller value to the irrelevant words.
To visualize clearly the top most relevant words according to the pagerank evaluation is possible to use word clouds, visual representation of text data in which the importance of each tag is shown with the font size. Some examples (60 top words) are shown in figures (18, 19, 20, 21).

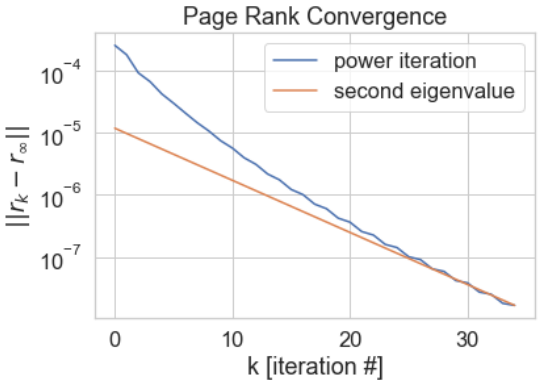


Fig. 17: Page-Rank Convergence

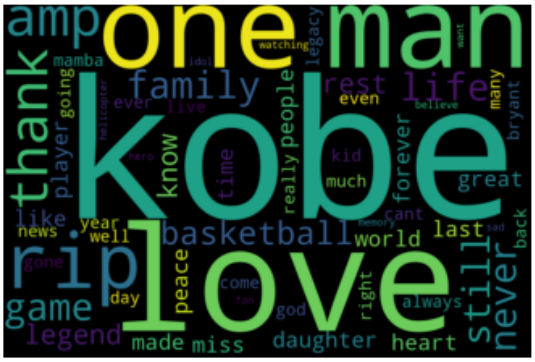


Fig. 18: Kobe Bryant top 60 words

From the comparison between Kobe Bryant and Obama top pagerank words is possible to identify a common topic. This characteristic is present also between the other networks and show that the pagerank can be a useful tool to detect common topics between the networks (hashtags).

We subsequently decided to perform the total sentiment of a network using the weight of the pagerank value for each node. This score can be computed as follow:

$$\sum_{\forall n \in N} MS_{V2n} \cdot PR_n$$

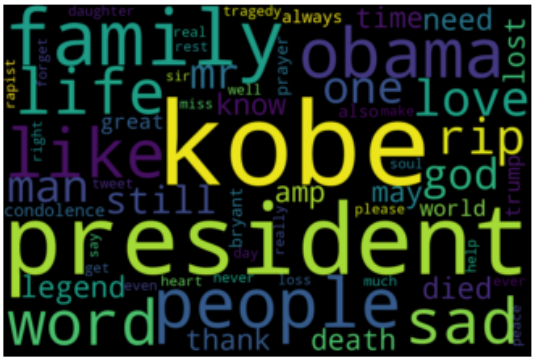


Fig. 19: Barack Obama top 60 words

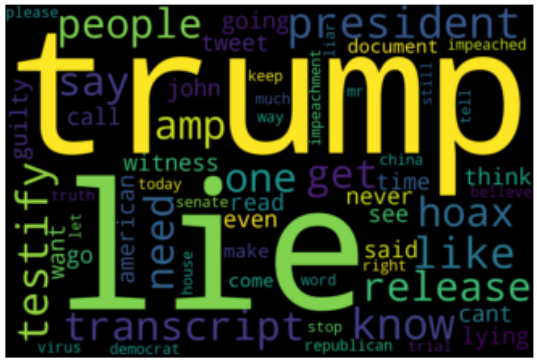


Fig. 20: Donald Trump top 60 words

Fig. 21: Guy Verhofstadt top 60 words

Using the pagerank value of each node for this type of evaluation, in fact, will result in a sentiment value that attach more weight to the important nodes.
A further analysis can be shown also in the figure (23).

This function represent the variation of total sentiment score as the least important word (the word with the smallest pagerank value) is removed from the network. The structure of the algorithm has a structure as follow:

The marks on the graph represents the words that change significantly the sentiment value of the network. As expected those words are removed in the latest iterations and have all a polarized sentiment score (MS_{V2}). The list of this words for the Kobe Bryant network and the percentage in which they were removed is in table (4)

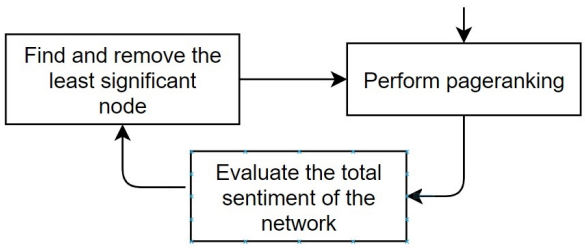


Fig. 22: Simple block structure of the algorithm

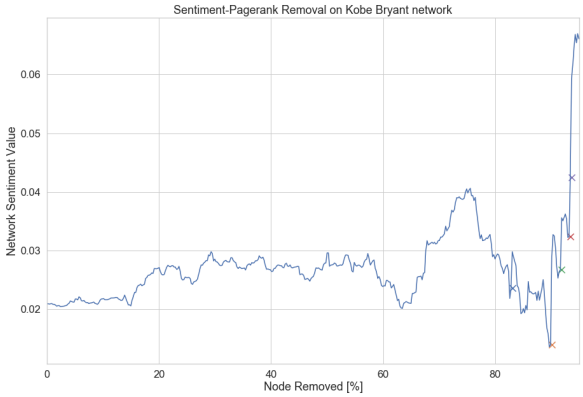


Fig. 23: Sentiment function of Kobe Bryant network

Word	% of removal
moment	83.04
hard	90.09
say	91.85
heartbroken	93.39
rip	93.61

TABLE 4: Word that cause significant variation in the total sentiment score

A relevant result that appears from this type of analysis can be seen from the comparison between Obama and the Pope sentiment functions. Fig (25), (24).

The functions have a different starting point: the sentiment value of the Pope original network in fact is slightly more positive than the Obama network (0.022 - 0.009).

As the nodes with less relevance are removed, in the Pope function is possible to see an increasing sentiment value while in the Obama network it remains almost constant in a range between 0.008 and 0.012.

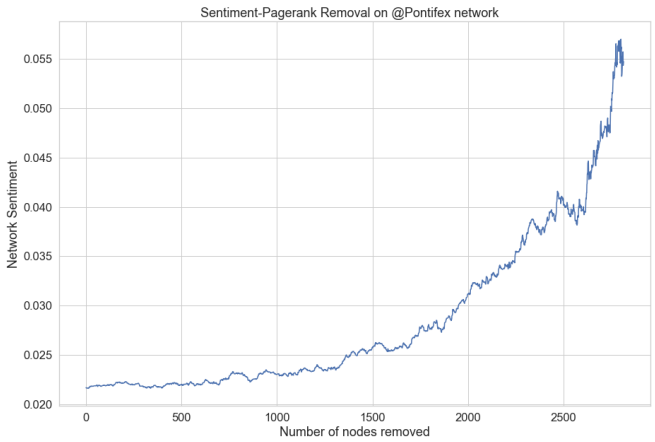


Fig. 24: Sentiment function of Pontifex network

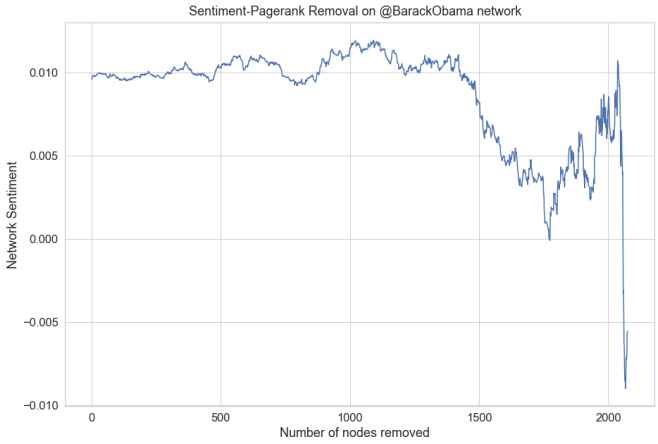


Fig. 25: Sentiment function of Barack Obama network

In the last iterations the words assume more and more relevance, having a higher pagerank value; the functions present steep changes of value depending on the sentiment of the most relevant nodes. For this reason is possible to identify a negative trend in Obama network most relevant words while the Pope network keeps getting more positive.

In conclusion this pagerank-based comparison between networks contributes to the general sentiment analysis, it prevents some random words with irrelevant link from having too much influence and helps targeting specific trends in a set of comments.

VII. Qualitative Analysis

A. *Qualitative analysis from robustness*

Having regard to these data, we apply a more qualitative analysis on two different databases: we plot Trump and Pope's network graphs.

Using the adjacency list (n=1), we run a Force atlas two layouts on Gephi (scaling 40 and preventing overlap), we size of the graph's nodes according to frequency score (0,1 to 15), we use a grey color (BDBDBD) for the edges to focus on the nodes. We have done four type of analysis:

- We highlight the most frequent words in figure 26: we size (0.1 to 8) the label of the nodes (words) on the basis of their frequency, and then we colored the words based on their MS_{V2} normalized to see the general sentiment of the most frequent words.
- We highlight just the negative words in figure 27: we size the label (0.1 to 4) of the nodes (words) on the basis of their negative score, that is 0 for words with no negative score, till 1 to the most negative terms.
- We highlight just the positive words in figure 28: we size the label (0.1 to 4) of the nodes (words) on the basis of their positive score, that is 0 for words with no positive score, till 1 to the most positive words.
- To better understand which negative words are more relevant for the network, we filter the nodes on the basis of their frequency, deleting all the words appearing less

than 20 times. Then, always watching just the negative words following their negativity score, we produce the last graph.

We can now qualitatively analyze the different centrality of the two types of words: negative and positive.

In Trump's graph shown in figure 27a the negative words appear sizing the words on the basis of their general frequency (guilty, liar). Sizing according to the negative score, we see that the more negative and central words are about the impeachment topic and with an extreme legal-discussions vocabulary (criminal, corrupt..). Filtering just the most frequent words (figure 29a), we see the centrality of those words in the graph. In Pope's graph of figure 26b, instead, we don't see any negative word between the most frequents, and the negative words are mostly connected to the satanic topic or are adjective related to the right and wrong actions (evil and fear afraid...)(figure 27b). Filtering the most frequent words (figure 29b), we see that the negative words remaining are not central as in Trump's network.

The position of the words in the graphs of figure 29 confirm the results of the robust analysis in section IV. In Trump case negative word are more central so are more linked to other words. Instead in the Pope network negative words tend to be on the edge of the graph so are less linked to other words and consequentially less important for the network structure.

B. From assortativity to cluster

Starting from assortativity analysis, we saw that Cristiano Ronaldo has negative words mostly connected to negative ones, and we know that those negative terms are not so many (as we saw in the robustness analysis). Instead, Obama's network shows an interesting outcome in local assortativity, considering separately negative and positive words. To go deeper into those results, we performed a cluster analysis on Gephi to qualitatively understand the reasons for these results.

We use the adjacency list (n = all tweet), and we run the modularity algorithm (with edge weight and resolution 1.2). We get score modularity score equal to 0.302 for Cristiano and to 0.287 for Obama, then we divide by color the most relevant clusters, to better visualize them. We performed three types of words sizing:

- We size the nodes in accordance with their frequency to understand the topic for each cluster qualitatively (figure 31).
- We size the nodes in accordance with their negative score to understand the dominant polarity of the different clusters (figure 32).
- We filtered the nodes for their degree range, deleting the nodes (degree < 25) to understand the relevance of the negativity among the most frequent words (figure 33).

Whit Cristiano Ronaldo's first network, we manually separate the four more relevant clusters, and we re-run Force Atlas 2 layout to create a better representation. We see that the two groups are related to the dead of Kobe Bryan (bottom of the image): respectively are $28,51\%$ and $12,73\%$ of the nodes of the graph. The other two cluster are generally related to the football player and his sport (upper part of the image), and they are $32,73\%$ and $11,82\%$ of the total nodes' number. In the second network, we see that the Kobe Bryan related cluster (bottom left) is the one with more negative words, mostly associated with basketball player's dead (unthinkable, heartbroken, sadness...).

The third network shows us that, among the most connected words, just in the Kobe related cluster, we have negative words (sorry, sed, heartbreaking, painful).

Whit Obama's first network, the cluster division, is evident, also without manual separation. We consider the first three relevant clusters: the first one ($31,15\%$ of the nodes) is related to the Kobe Bryan's dead, and it is in the left part of the graph; the second one ($24,89\%$ of the nodes) is about the ex american president and it is in the right part of the chart; the third one (just 6.36% of the nodes) is related to the president-in-office Trump, and it's in the left part of the graph as well.

In the second graph with just the negative words we see that the Kobe's cluster has more words that are more central and more negative (horrific, heartbreak, tragedy...); while Obama's words group has less central negative words (unfortunately, pity, painfull...); Trump's cluster it's minimal and have just a few negative words (inequality, calamity).

In the last graph we see the relevance of the negative words with a degree higher than 25, confirming our hypothesis: Kobe's cluster has much more negative and frequent words compared with Obama's cluster. Trump's words group is not big enough to appear in this visualization.

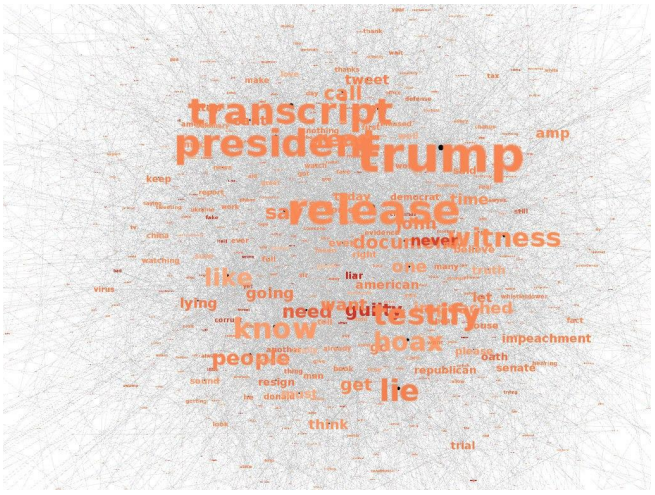
VIII. Concluding Remarks

In conclusion we managed to create a methodology to test the sentiment of a words' corpus using network analysis techniques and we achieved to use those statistical analysis to drive the qualitative part of the research.

We found a correlation between the levels of polarisation of different social actors and their networks' negativity applying a Robustness analysis. We validated this analysis with the Assortativity statistic. Using the Page Rank we tested the global sentiment of the network with the same conclusions.

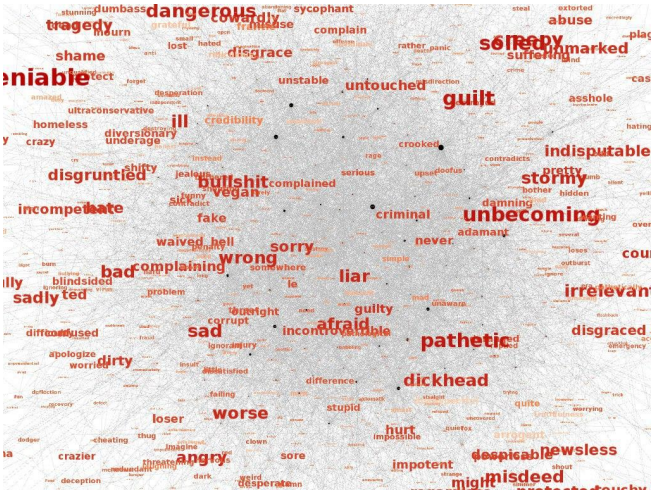
Analyzing qualitatively the words contained in the network we made assumptions about the different topics covered in the corpus and with a cluster analysis, we were able to compare a discussion about the Kobe Bryant's dead across various social actors.

We need to repeat the analysis with a bigger database to further validate our preliminary findings. Focus the research just on comments instead of analyzing all the tweets related to a specific account would probably give us more relevant results.



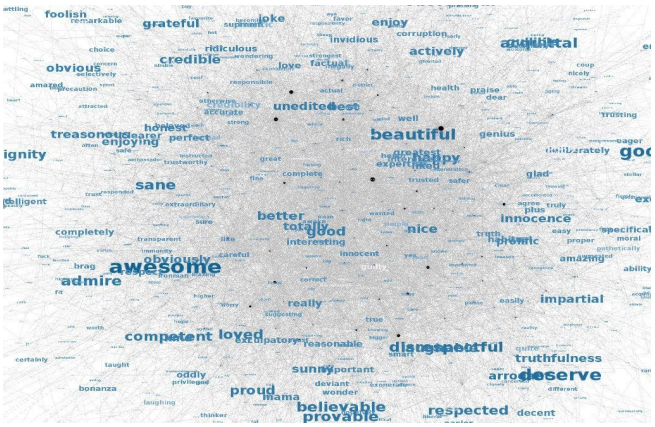
(a) Trump Network (b) Pope network

Fig. 26: Networks with most frequent words



(a) Trump Network (b) Pope network

Fig. 27: Networks with all negative words highlighted

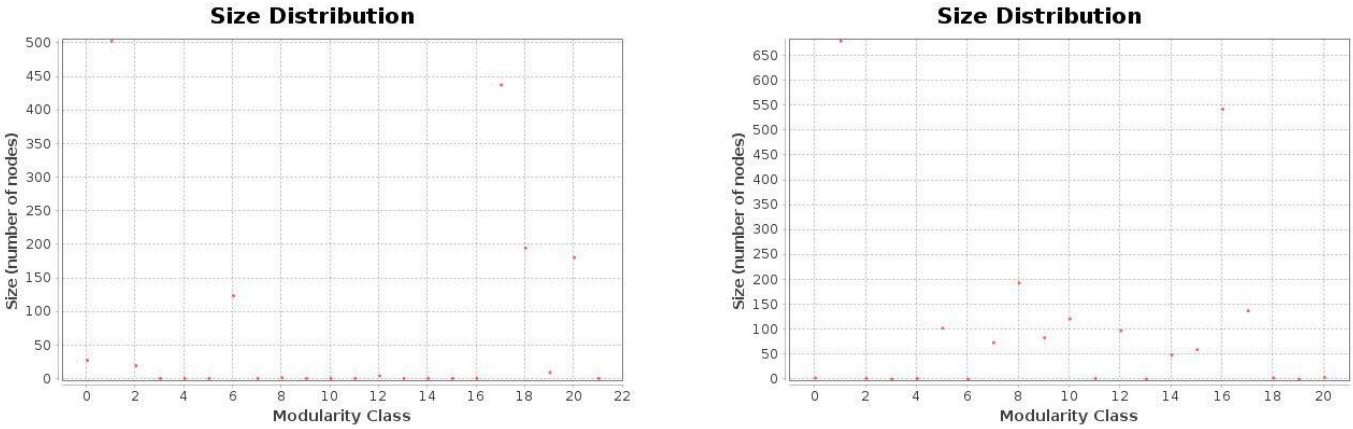


(a) Trump Network (b) Pope network

Fig. 28: Networks with all positive words highlighted

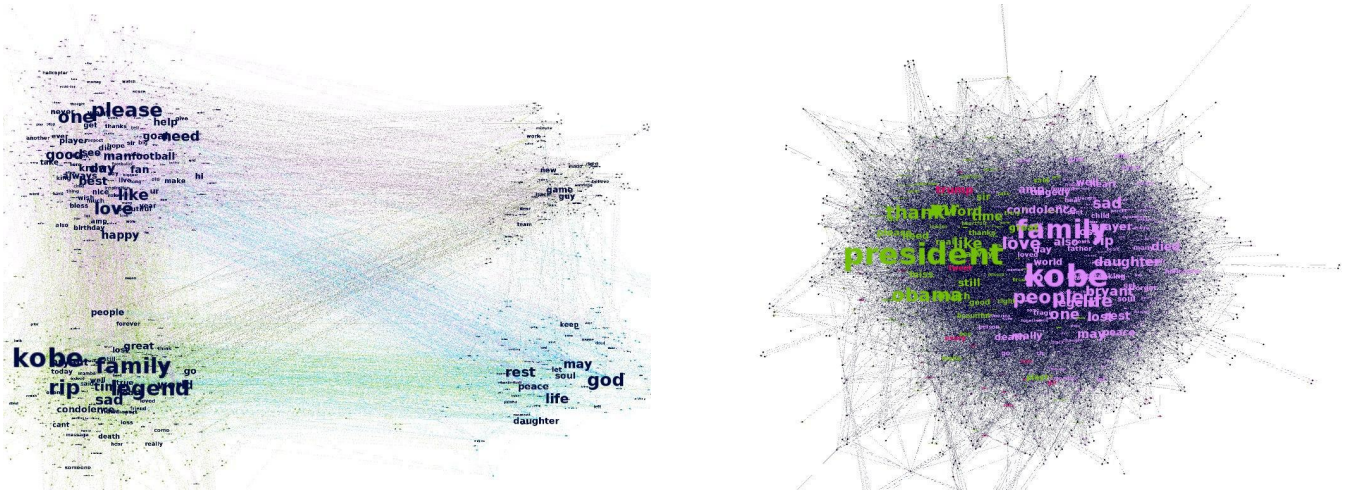
(a) Trump Network (b) Pope network

Fig. 29: Most negative words in networks



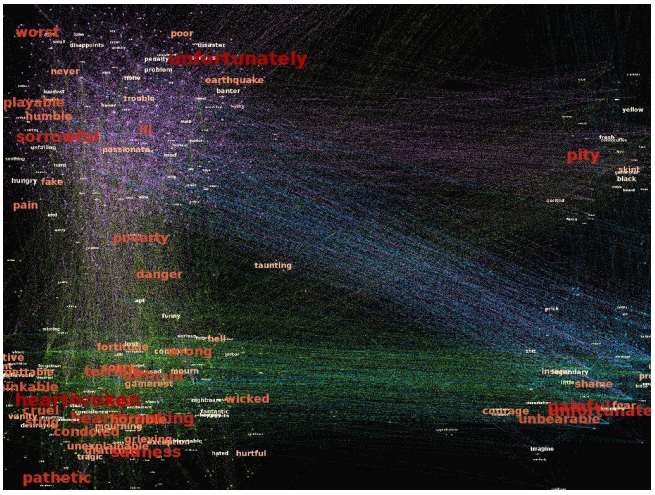
(a) Ronaldo Network (b) Pope network

Fig. 30: Modularity of the networks

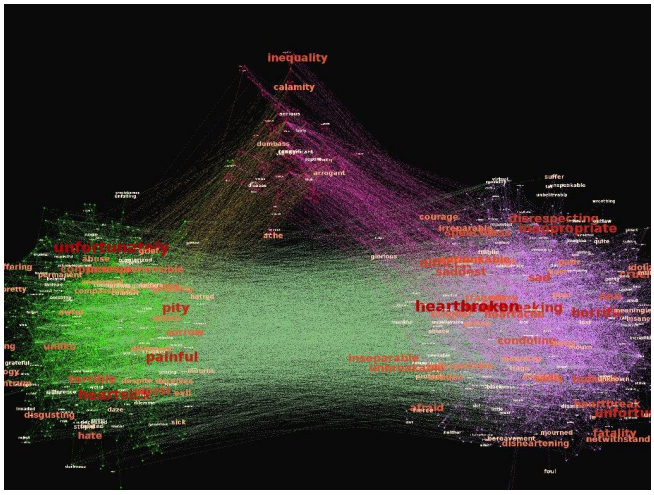


(a) Ronaldo Network (b) Obama network

Fig. 31: Cluster with node sized by word frequency

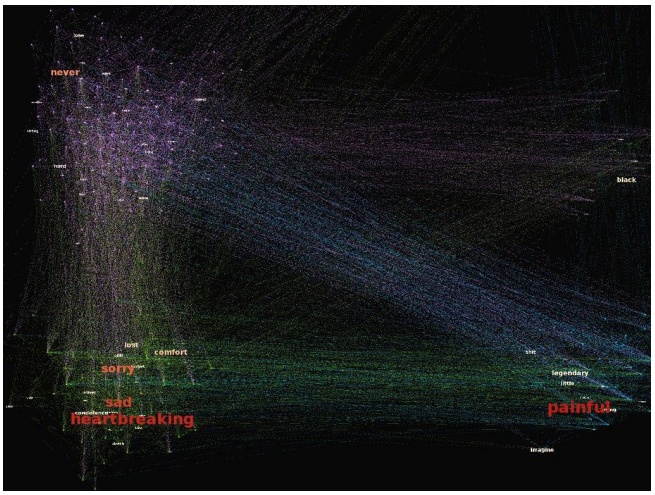


(a) Ronaldo Network

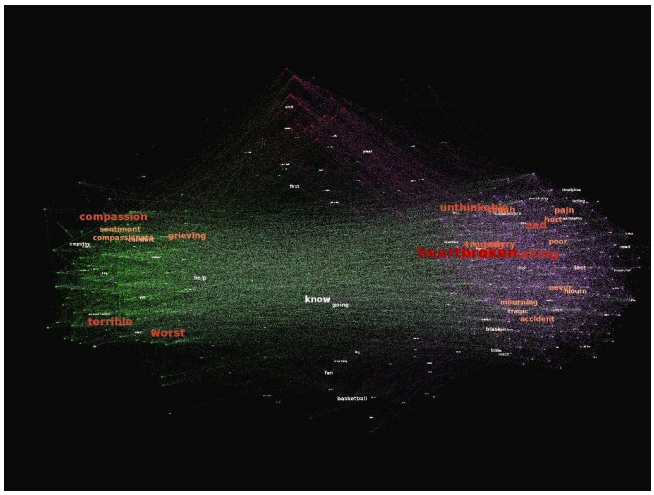


(b) Obama network

Fig. 32: Cluster with node sized by negative score



(a) Ronaldo Network



(b) Obama network

Fig. 33: Cluster with node sized by negative score and filter by degree

IX. Spectral Clustering (Abanoub)

Networks are usually composed of communities that are more linked to each other than the rest.in this section, we use spectral approach for community detection. We used spectral approach to divide our network to two communities according to the signs of fiedler's vector. For that,we need to go through this steps, to build the normalized Laplacian matrix and extract the eigenvalues. The Fiedler vector corresponds to the second smallest eigenvalue, the Fiedler vector have negative and positive values which divide the network into to communities according to the sign.

As a first step we constructed the Adjacency matrix based on the positive and negative scores of the tweets we have. Any tweets that are close to each other are linked, i.e have a similar score for positive and negative. For that we considered two tweets as linked if the difference between their scores is below 0.2. Second step was to build the Normalized Laplacian matrix L using the following formula

$$L_1 = I - D^{-\frac{1}{2}} A . D^{-\frac{1}{2}} \tag{5}$$

Where A is our adjacency matrix and the D degree matrix. Once we have the Normalized Laplacian matrix, we extract its eigenvalues. This work was done for different group of tweets,for Trump, Obama, Mike Pence and the Pope. Figures 34, 38, 36, 37 and 35 are the different plots of the eigenvalues of the normalized Laplacian matrix. The difference between the last two eigenvalues is called the eigengap and it is used to determine the optimal number of clusters, in fact, the Eigengap suggests the number of clusters k, that is usually given by the value of k that maximizes the eigengap (difference between consecutive eigenvalues). The larger this eigengap is, the closer the eigenvectors of the ideal case and hence the better spectral clustering works. The eignegaps for all the figures seems similar however the ones for trump seems slightly larger, and thus slightly better clustering. Another

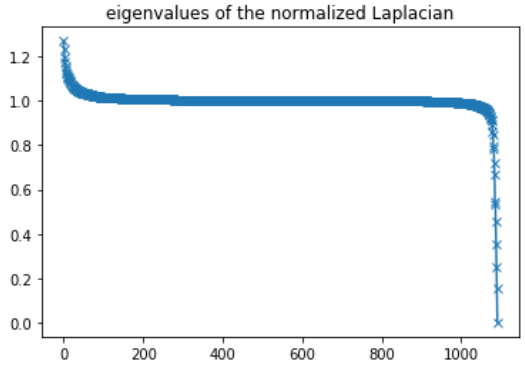


Fig. 34: pop

approach to detect the network communities is to consider the best conductance value, for that we consider ordered nodes according to their score in Fiedler's eigenvector, sweep across the nodes to check the minimum conductance value. Figures 39, 43, 41, 42 and 40 shows the plots of conductance

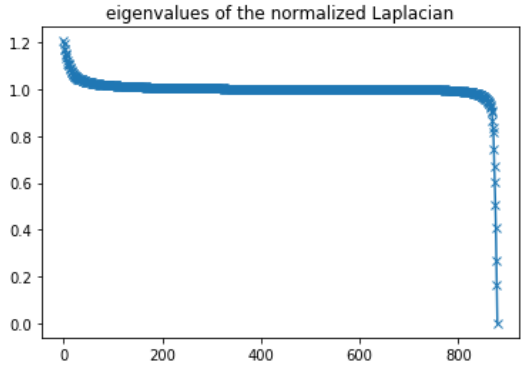


Fig. 35: musk

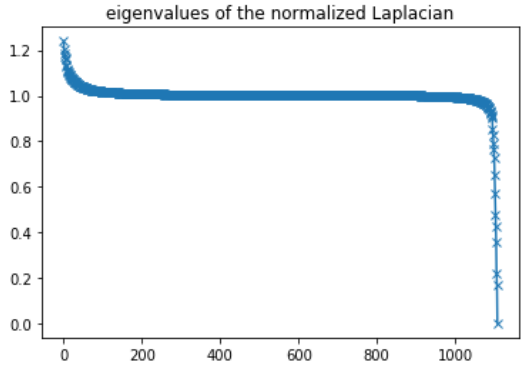


Fig. 36: pence

values of each node in the different networks, the values of the minimum conductance indicating that that's the best conductance values to divide the nodes of the network into communities.

The following table is a sum up of the size of the two communities of each network.

Communities	Obama	Trump	Pence	Il pop	Musk
Minimum Conductance	0.1	0.15	0.25	0.2	0.3
Community 1	400	550	610	450	820
Community 2	400	450	500	600	100

TABLE 5: Communities based on minimum conductance

The network of tweets about Obama and trump seem to have smaller minimum conductance which shows that these tweets are better clustered than the other networks. This could be interpreted in a way that the tweets about Obama and trump are either with a high positive score or with a high negative score, whereas the other networks have more tweets that are likely neutral which make it slightly more difficult to cluster it. On the other hand, considering that the community 1 is the community of positive tweets, the network of Musk seems to be having the highest number of positive tweets, whereas the rest seems to have more or less similar number of positive and negative tweets.

The plots in Figures 44, 48, 46, 47 and 45 present in two different colors the two communities of the networks based

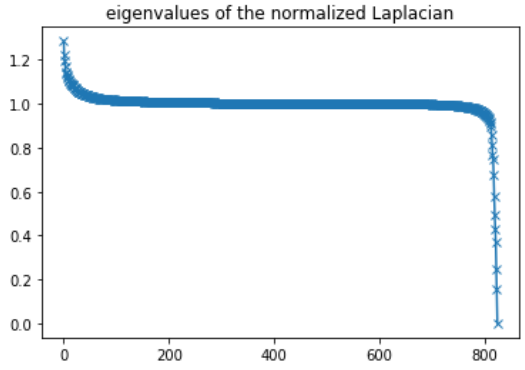


Fig. 37: obama

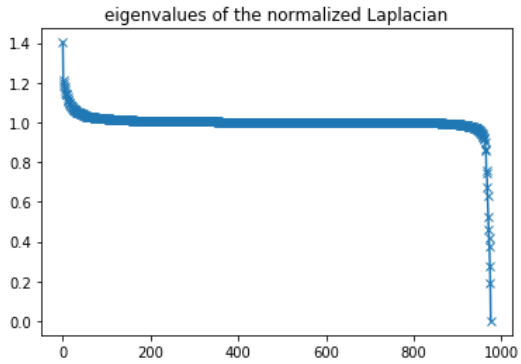


Fig. 38: trump

on the spectral approach based on fiedler's vector explained above.

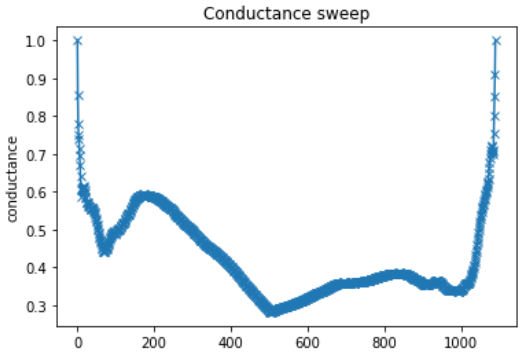


Fig. 39: pop

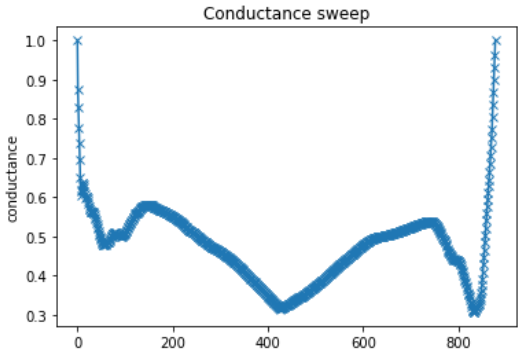


Fig. 40: musk

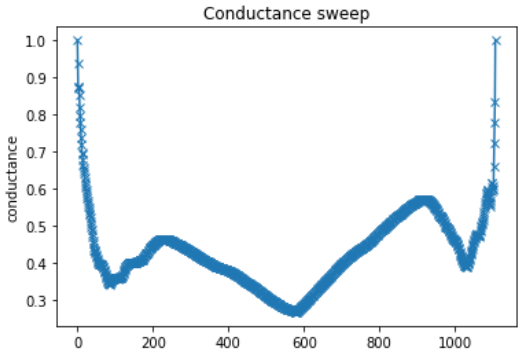


Fig. 41: pence

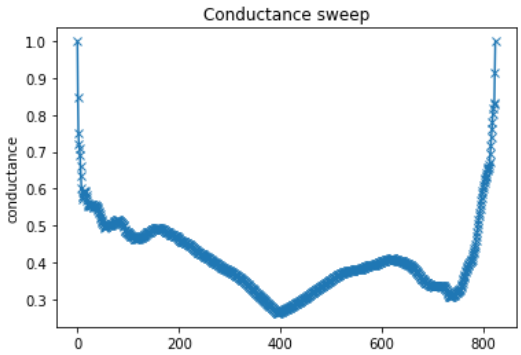


Fig. 42: obama

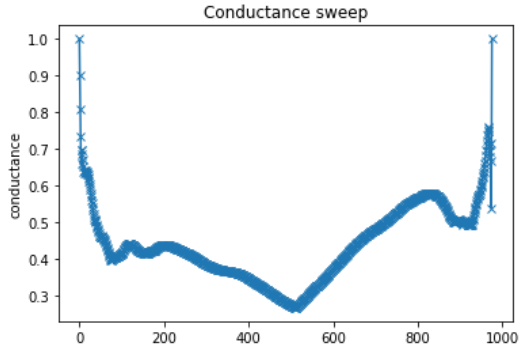


Fig. 43: trump

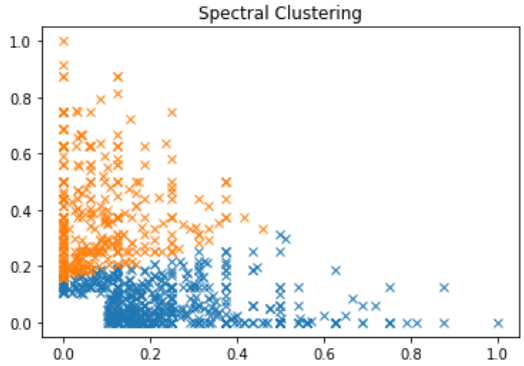


Fig. 44: pop

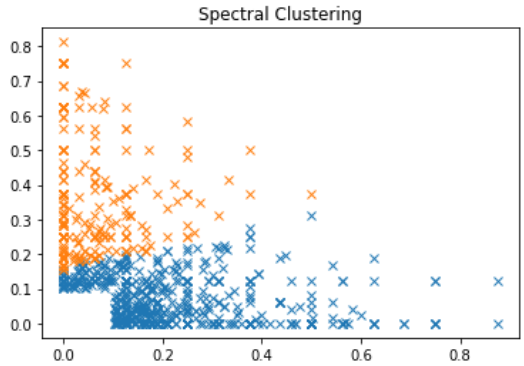


Fig. 45: musk

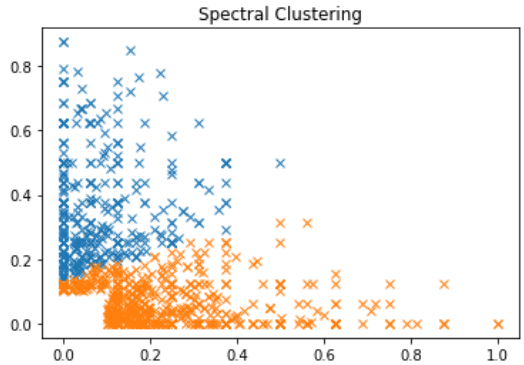


Fig. 46: pence

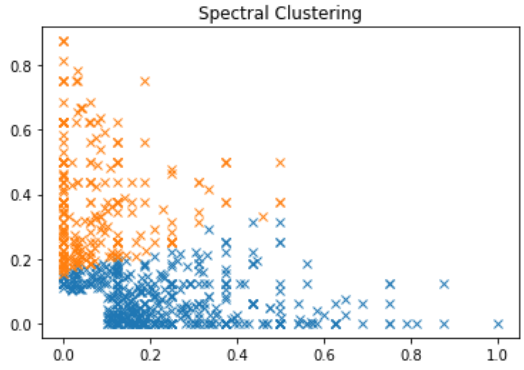


Fig. 47: obama

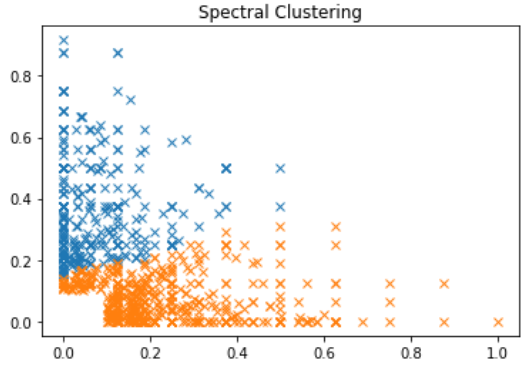


Fig. 48: trump