Spring semester
Report

Final Project: Corruption simulation in social network

Complex Networks

Master
in
Artificial Intelligence

Submitted by

Miguel Méndez
Juanjo Rubio

Under the guidance of
Sergio Gómez Jiménez
Alejandro Arenas Moreno

UNIVERSITAT
ROVIRA i VIRGILI

Universitat Rovira i Virgili
Master in Artificial Intelligence

Contents

1 Introduction

The use of Social Network as a political tool is a common practice and plays really influential role, specially in young and middle-age people. During last years this practice has became more and more popular and nowadays they are considered as important statistical methods that are taken into account to predict elections results or the influence of the politic parties, we can refer for example to [1] where authors try to predict the result of US elections through a sentimental analysis with data obtained from Twitter social network, and also to [2], where a study of different blog publications was made during Swedish elections,

Besides being a communication channel, social networks underlay information from the structure of the relation between individuals. In the early 2010's the presence of politicians in Social Networks was increased, specially influenced by the important role that Social Media played during the US elections of 2008 and 2012. The structure given by the profile of a given user can provide static information such as the follower-following relation as well as information provided by interactions. This projects aims to perform an analysis on Twitter data applying techniques learnt in the *Complex Networks* course. Specifically, the analysis performed is oriented to the relations between politicians with a focus on the information about corruption that can be extracted from such analysis. Considering this, the project itself involves both the data extraction as well as an static analysis given by the graph built from the dataset.

On the other hand, we perform an Epidemic Spreading over the whole network in order to simulate how corruption evolves in a politic ambient. The main idea is being able to obtain through a starting user, which is involved in some corruption case, a path of influence with other nodes. This is that we are going to predict which politic will corrupt himself due to the influence of another politic. For doing this we have design our own spreading algorithm based on what we have learnt during this course and adding real world knowledge that can be extracted for Twitter. In following sections we will explain also how this algorithm works, how influence can flow and we will simulate some cases in order to observe the performance of this method.

2 Dataset

As stated in 1 the dataset used is extracted from Twitter and contains information about politicians. The extraction of information has been performed using the *Twitter API*[3].

As we know Twitter is an online social network service that enables users to send and read short 140-character messages called "tweets". In this project we are not interested in what people is saying, what we are looking for is how people, politicians in this case, relate with ones which others. We suppose that politicians that belong to the same politic party or have common ideas will follow between them. In the same way, politicians that work together or known each other will also maintain a relation in the social network. The main idea is being able to extract all these relations, construct a dense graph where it would be possible to observe with which people is a politician sharing information and how information is spread along the network.

Many different factors have to be taken into account in order to construct this network properly. First of all, we need to set up a starting point, which should be an user followed for many people and most of this people should be politicians. Think that if we start at a random user, it is highly probably that after a few iteration our algorithm would get stuck. In a social network like Twitter, famous or popular people have a large number of followers, which embrace many different types of user. For this reason we decide that instead of building the graph from the relations given by the followers we chose to build the connections from the friends list (i.e. the list of people that the given user is following) since it usually gives more information that defines the type of user and it usually involves a smaller amount of users than the followers list. Heuristically, we determined that considering the type of users we aim to analyse (politicians), a wide variety of users will follow them but being a heterogeneous group while the following list yield more connections to other politicians and personalities. With this consideration in mind, our root node for building the graph is *@ppmadrid*[4] since its main purpose is to track different politicians, specially politicians from the PP political party.

The main reason why this dataset is built mainly from politicians of a specific political party is due to many reasons. First of all, we needed to limit the workload of the mining process due to the limits given by the Twitter API. On the other hand, one of the goals of this project is to try to evaluate complex network techniques as tools for analyzing corruption cases in politicians and in the last years PP has been involved in many court cases involving this issue.

2.1 Preprocessing

As we explained in previous section, it is very important to filter the obtained users in order to avoid people and companies that are not involved in political world. This is not an easy task, and we spend some time thinking about how we could achieve this target.

In Twitter, every user can write a little description about himself which is usually called biography and it is extension is limited to 160 characters. This fact forces user to be very specific in a way that most of them include their actual job, the politic party they are enrolled or even their political ideas. After some deep search along different users in Twitter we extract some of this keywords and we decide to take into account only those users who include one or more of them in their biography. Some of this words can be found in table below:

Keywords
PP
Alcalde
NNGG
Partido
Popular
Derecha
España
Concejal

Table 2.1: Some of the keywords used to filter user's biography

This particular filtering based on a bag of words delivered very good results when using the biography as the source of information. It is also worth it to note that this simple approach is also filtering some non-desired users due to some of the keywords being a subset of some words (e.g. pp — happiness). At first sight we thought this effect was inconvenient but we decided to keep this outliers in order to build a slightly more heterogeneous graph even though it is mainly focused on politicians. Another effect of this filtering via bag of words is including politicians of other political parties (e.g. through keyword *concejal*) which is also a positive effect in order to compare different parties in the analysis.

Another restriction introduced in our system is to limit the search to a particular region in Spain in order to make the mining process and dataset manageable. In this case, the procedure followed was making use of the location of each user queried. The average user has a considerable probability of not giving the location but comparing manually politician twitter accounts, the vast majority of them specify their location in which they develop their political career. This preprocessing has made use of the list of a Wikipedia list [5] which contains 179 towns and cities of the Madrid autonomous community. Using the same method as with the previous case, the location is matched against this list and only those that contain information and is present in the new bag of words is considered and kept as a node. All this processing has been performed by removing case sensitivity as well as removing accents due to inconsistencies in some of the locations provided. This method also delivered good results and it proved to be more reliable than using last-tweet location due to the variability of current location in politicians.

In table 2.2 we collect some of the retrieved users and their biographies in order to facilitate a better understanding of the performance of this method to the reader.

Username	Biography
@beatrizpabraham	Licenciada en Derecho por la UCM. Teniente Alcalde y Concejal de Familia, Asuntos Sociales y Mujer del Ayuntamiento de Pozuelo de Alarcon
@laravente	Concejala del Partido Popular en el Ayuntamiento de Fuenlabrada, Funcionaria de Hacienda de profesion
@Sorayapp	Vicepresidenta, Ministra de la Presidencia y Portavoz del Gobierno en funciones. Diputada del @PPopular por Madrid
@beatrizelorriag	Twitter oficial de Beatriz Elorriaga y su equipo. Concejala del Partido Popular en el Ayuntamiento de Madrid
@ignaciobelaunde	Concejal del Ayuntamiento de El Escorial. Presidente de la empresa de regalo promocional Belaunde Global Business Holding.
@MsolHernandez	Periodista de EL MUNDO apostada en el Congreso de los Diputados. Cubriendo la informacion del PP.
@frantomoe	Miembro Comite Ejecutivo del Partido Popular de Fuenlabrada, responsable RRSS.

Table 2.2: Dataset example

2.2 Graph generation

Once we have clear what kind of preprocessing we need to apply we need to decide how to iterate over all the users that we consider valid for our purposes. We need to remind that the key of the process is find a valid user, extract its friends, filter them and repeat again the same steps iteratively.

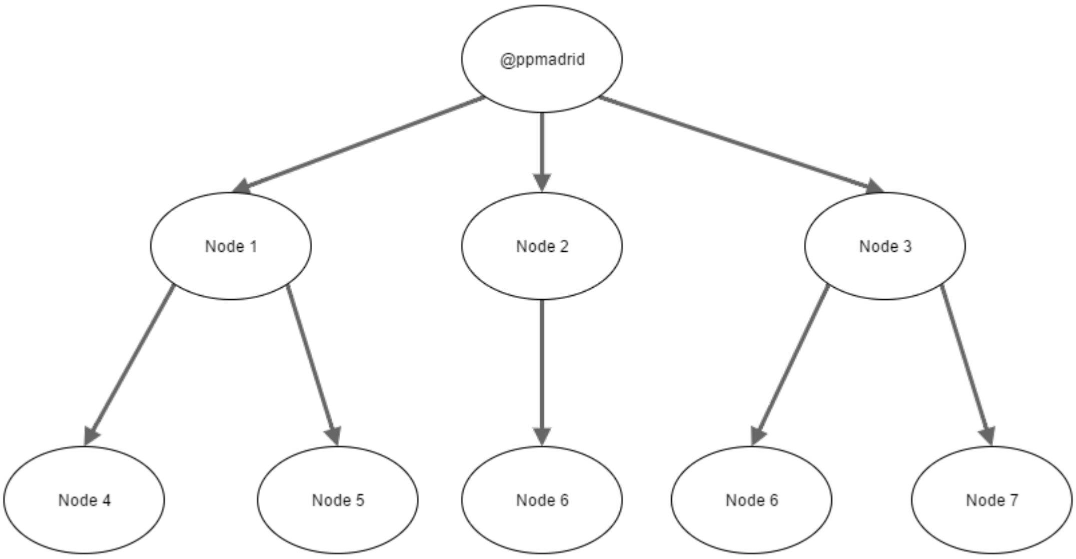

Figure 2.1: Graph generation using BFS algorithm

There are many ways of doing this and it is possible to apply a classic search algorithm. We decide to apply Breadth-first search, because we consider that having into account Twitter restrictions would be more useful to expand the graph horizontally instead of pruning it in depth. The main difference is that in this way we are given priority to relations between users, this is more edges to our graph. If we had decided to use a depth method, we would obtain more nodes or users but a more sparse graph. It is worth it to note that in a large interval of time both methods would reflect very similar results, but Twitter API is very restrictive with the number of queries, which have made us to choose the breadth method.

In Figure 2.1 we show how this method works. We start from @ppmadrid user, we extract all his friends and we filter them. This will correspond with the first second level of the graph showed in the figure. After this repeat the same process with node one, two an three achieving a new level. During this process, when we find a user that has been already visited and expanded, we add a new edge to him and we jump to the next node.

3 | Static Analysis

First of all, our first approach to analyse the data mined via the Twitter API we performed some analysis from the built dataset. This information has been extracted from the static structure of the corresponding built graph.

3.1 Degree Distribution

The first analysis performed was obtaining the degree distribution in order to make sure that the graph built is correctly generated. We should expect a power-law network since social networks tend to follow this distribution. This fact would confirm that the the graph is built correctly from the Twitter data as well as confirming the existence of *hubs* which its main identification would give very valuable information.

In the figure 3.1 we can see the computed histogram of the number of connections of the nodes which gives us an idea of the topology of the network.

As we can see, the histogram resembles significantly the expected structure present in social networks, in which the probability the fraction $P(k)$ of nodes in the network having k connections to other nodes goes for large values of k as $P(k) \propto k^{-\gamma}$. As stated before, this fact denotes the presence of a subset of nodes that concentrate a large amount of connections and are defined as "hubs" of the network. This particular subset of nodes are really important in terms of the conclusions that can be extracted in the simulations performed later on.

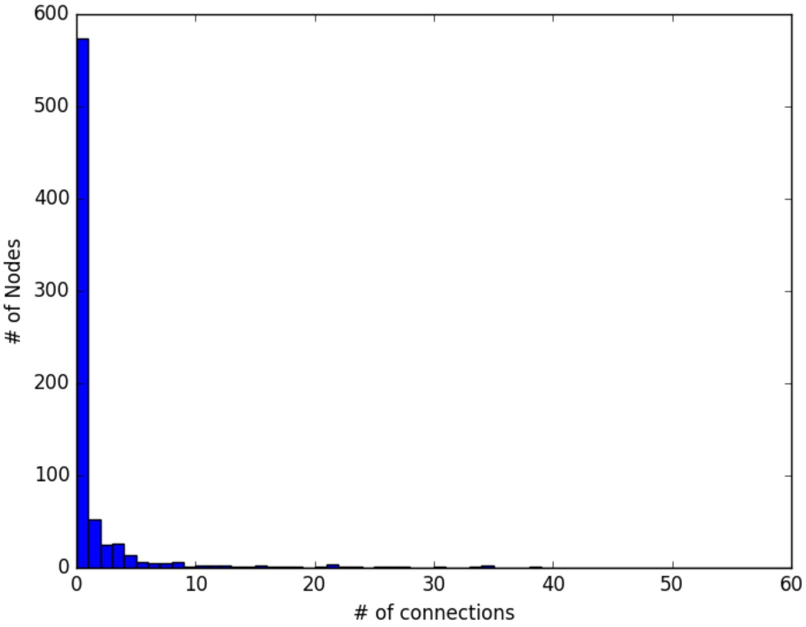

Figure 3.1: histogram of connections of nodes in the network

3.2 Hubs and Relevant Users

In this section we will study the graph that it has been created through the Twitter obtained data. The idea is try to find those users that are most connected, this is that have the higher number of neighbours. They will act as hubs in this network and they influence will be high in posterior steps when we apply a disease spreading algorithm. We have to think that if one of this nodes is infected, as it is in contact with a high number of node, the spreading will be larger because its influence is high.

If we think in real world, we can also observe this pattern. Think about a little town where the town hall has a small number of politicians. These are maintain relationship ones with each others, but they are not very known outside the town and their influence is low. For this, is one of them is corrupted, it is probable that some other politicians of the town are also corrupted but as their influence is low, it will be hard that they can corrupt people of another towns. Nevertheless, if we are speaking of a high level politician, think about a big city major, a senator or even the president of an autonomic community. In this case, it is clear that they maintain a bigger number of connections and if they get corrupted their influence will be very high.

Another important point in this analysis is that it can serve us also as a test to check the performance of our data. We have filtered a lot of users of Twitter and we have tried to maintain only politicians or accounts related with politic world, so a popular user in this network should be someone popular in such ambient. It is worth it to remind that we have restricted our search to Madrid community, so we should only obtain people related with Madrid's politics.

In table 3.1 we collect the most important or relevant user of this network.

Username	Biography
@PPopular	Twitter oficial del Partido Popular.
@EsperanzAguirre	Madrileña. Liberal. Patriota. Portavoz de @grupoppmadrid.
@nnggmadrid	Nuevas Generaciones del @ppmadrid. Presidenta Ana Isabel Perez (@Anai_pb).
@carlosizqtorres	Consejero Politicas Sociales y Familia @ComunidadMadrid / Vicesecretario Sectorial @ppmadrid / Presidente @ppcarabanchel / Economista e Interventor
@cifupresidenta	Equipo de @ccifuentes. Un proyecto nuevo e ilusionante para la @ComunidadMadrid y para el @PPMadrid
@PPAsamblea	Twitter del Grupo Popular de la Asamblea de Madrid. Presidido por @ccifuentes
@ihenriquezluna	Concejal y Portavoz Adjunto de @GrupoPPMadrid y Presidente de @dtosalamancapp

Table 3.1: Most relevant users in the network

As we can see we can say that we have obtained satisfactory results because the user accounts found match with our expectations. Let's study this table more in depth.

Two of the accounts that have more users are @PPopular and @nnggmadrid. Both of them are represent a political party, not an specific user. As it is difficult to filter only those accounts that represent persons without going into Natural Language Processing techniques which were considered out of scope of this project, this results were expected.

Nevertheless there are also very important accounts as @EsperanzAguirre which is one of the most influence persons in Madrid, so obtaining her here can be considered as a positive fact. We have also found curious users as @carlosizqtorres and @ihenriquezluna, that also match with our expectations and will be very useful for the posterior spreading phase.

On the other hand, another information we can obtain in this step for analyzing the spreading simulation is those nodes that are indicted or involved in a corruption court case. For this matter, a list[6] of politicians involved in court cases has been extracted from Wikipedia and used to find nodes in the graph that match the names of this list. Once this "busted" list is obtained, we can use them as root nodes in our simulation described in section 4.1.

Real Name	Biography
Agustin Juarez	Presidente y Portavoz del Partido Popular de Collado Villalba.
Almacenes Granados	Especialistas en estampacion digital!!! WHATSAPP 633 04
NNGG Cubas	Nuevas Generaciones del Partido Popular de Cubas de la Sagra
Jose Carlos Boza	Alcalde de Valdemoro

Table 3.2: Matched users from politicians involved in court cases in Madrid

As we can see in table 3.2 it is easy to spot some outliers. This is due to the fact that we didn't perform a very fine-grained filtering in order to have a more heterogeneous network. In this case we see how *Almacenes Granados* does not correspond to a politician but it shares the surname with an indicted one and passed the bag of words filter due to the word "whatsapp" and the keyword *pp*. On the other hand *NNGG Cubas* shares also part of the name of an indicted politician in Madrid. On the other hand, if we take a look at the two remaining users, both *Agustin Juarez* and *Jose Carlos Boza* are politicians and involved in court cases and will be used later on to perform the simulation. Things that were considered for filtering the real politician users have been taken into account such as checking the verification of the account but the tendency of local politicians is not to have their accounts verified so the only possible way is performing some natural language processing in this lister in order to filter the expected user accounts.

3.3 Community Detection

Community detection is a very important technique when we are in front of a complex network, because it allows us to extract valuable information as we have seen in previous assignments.

In social networks, communities are formed by people with similar interests. For example if we think in Facebook, people who likes a specific subset of pages or accounts, as famous actor, will form a community. Also, those that live in the same area will form another community. In this project, we have created a graph thanks to the information retrieved in Twitter social network. As we try to give priority to politician accounts, then this graph will be mostly form by politicians and its friendship relations.

We apply an community detection algorithm which is present in iGraph library and it is called Edge betweenness algorithm. The idea behind is that the betweenness of the edges that connect two different communities is usually high, which depends on the number of shortest paths between nodes that are in different communities. This algorithm remove the edge with highest betweenness from the network and after this betweenness is computed again. This is repeated in a iterative fashion until the networks breaks in two component, and then the whole process is repeated until no more edges can be removed. Is a classical divisive hierarchical approach.

One of the problems of this technique is that you do not know the optimal number of communities so different trials are necessary along with a study of the obtained results. In this work we have applied the algorithm with different parameters, we export the obtained communities to text files and the we analyse the users in them and the results were pretty surprising.

After different trials we decide to obtain three different communities. In the first one of them mostly of the obtained nodes are politicians, names and biographies can be found in file *partition0.txt*. In the second community we found users which have an important role in a certain company and the third community seems to have a more fuzzy set of users. Some of the users are related with technology world, this is due to that one of the keywords we have used to filter users is “PP” and some user that contain the word app or application in their biography are being included in the dataset. But it is very curious how politicians are only in the first of the communities, it seems that the number of edges between them is very high so this results point that our dataset is well collected and formed.

4 | Disease Spreading

The last analysis performed in the graph built from the mined data is based on disease spreading techniques. Compared to the techniques studied and applied in *Complex Networks*, in our case we tried to built our own algorithm to obtain local information of a particular node instead of obtaining general data of the full graph such as ratio of infected people.

4.1 Corruption Path Finder

One of the main conclusions we intended to obtain in this particular project was finding relations between a set of given politicians and their superiors or supervisors. In many cases, when a corruption case starts to be investigated, many politicians try to avoid at all cost relations and disassociate the relation to the indicted politician even if there is a link between them.

In our case, once built the graph and once we obtained insights from the data such as the information extracted in section 3.2 we are in a position to perform a simulation that could provide information about the flow of influence and the relation from a given indicted individual to higher instances of its political party.

The techniques studied in *Complex Networks* were focused on obtaining global information such as the ratio of infected people and the effect of different values of infection and healing rates in different simulation algorithms such as as in the *SIS* (Susceptible-Infected-Susceptible) model. In order to analyze the path, and with the knowledge obtained in class, we built an algorithm that aims to obtain a path from a corrupted node. This particular model has 3 different states: *Susceptible*, *Infected* and *Corrupted*. Instead of running a simulation over all the time steps and repeated a set of times in order to average the ratio of infected people for different values of infection like in the Monte-Carlo simulation we have performed, in our case each step is simulated a certain number of repetitions but each step at a time. During a time step, the susceptible neighbours of the infected node are infected based on a β infection rate as well as some heuristics that will be explained later. After repeating the infection process a number of repetitions, the top 3 most infected neighbours are kept as *Infected* for the next step, while the current infected is set to *Corrupted* state and for the next steps the process is repeated with the new susceptible neighbours. With this procedure, we can build a path of influence of length t, which depend on the number of steps we have selected, and we can extract information from it, such as which hubs the paths traverse the most. In figures 4.1 and 4.2 we can see an example of the behaviour of the algorithm:

This paths have been obtained using our algorithm and results are quite surprising. We set Carlos Boza, Valdemoro's Major, as starting node because it is actually involved in corruption trials. The red colour of the node means that its state is infected, as we see our algorithm return as 3 susceptible nodes (green colour). Curiously, between this people are @antoniomartin6 who is Ajalvir's Major and affiliated to Popular Party, @David-Conde-R a Valdemoro's councillor and @MariaMarinPP which biography does not specify any charge but says that she has been fighting for improve Popular Party for 15 years. Well, it is clear than results are good and this spreading is being doing with sense, even us have been surprised for having this amazing results.

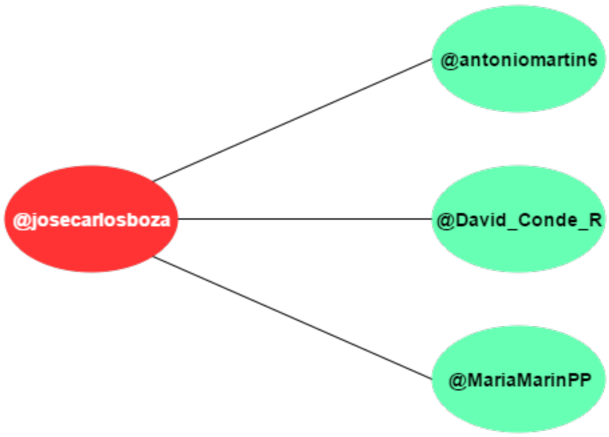

Figure 4.1: First step of the spreading process

Now in figure below, we show a second step of the algorithm, this is how is the spreading process carried out since @antoniomartin6. Now @josecarlosboza is drawn in purple colour because it is considered as a corrupted node and the state of the previous susceptible nodes has change to infected. One more time, we obtain very significant nodes as Carlos Iquierdo and Emilio de Frutos, who is Major of El Molar since 2000. Both users seems good candidates but we also have found @cifupresidenta which is not a valid user, is an account used to promote Cristina Cifuentes candidature for Madrid's community.

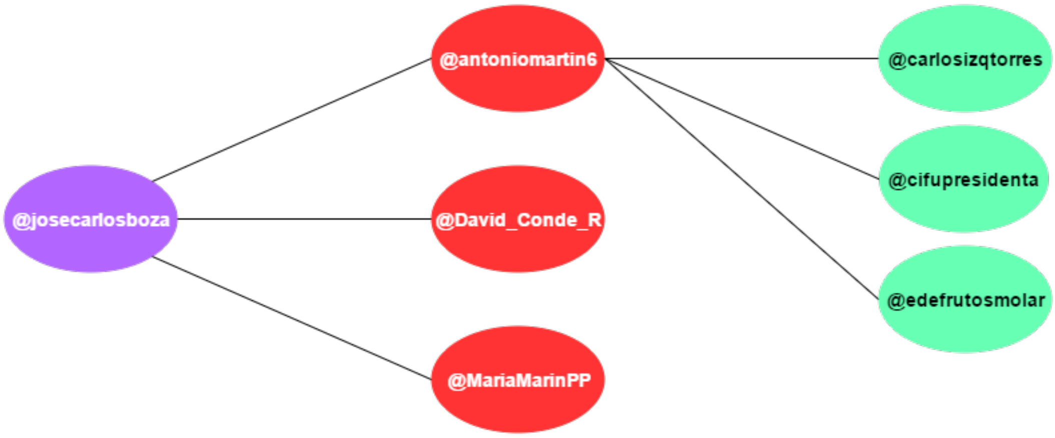

Figure 4.2: Second step of the spreading process

As stated before, the algorithm is not purely random, otherwise the paths at each different run would be completely different. The first heuristics applied are related to the location of the current infected node. As in real life, we determined that in terms of corrupted politicians, the first step is to try to obtain support from equals (e.g. a city councillor tries to convince other city councillors from the same town), and most of the times they are within the same town or city. Considering this, we make more probable at each time step to infect a susceptible neighbour if it shares the same location than the infected node. On the other hand, at the next step, we tried to apply more heuristics based on the importance of higher instances in the political party. In the past example, given a local city councillor, once he or she has some allies, the next step usually involves getting support from someone bigger in terms of position, which in our approach has been translated to giving more probability of infection if the neighbour has more followers. Generally, the higher in the chain of command a politician is in a political party, more followers it tends to have.

5 Conclusion

During the process of building this project, our aim was to study the application of some analytic techniques from *Complex Networks* for the particular case of politician Twitter account graphs. At first glance we were not expecting to deliver good results due to the humble processing and the size of the graph but in the end we were able to deliver results.

On the other hand, another personal goal was to be innovative in terms of not just testing studied techniques but trying to build a tool based on the knowledge adquired, which in the end made us try to simulate paths of disease spreading. This particular algorithm performed better than expected even though it still needs tuning and improvement as explained in 6 since at this point the heuristics are rather basic. As we saw in the figures of section 4.1 the results of some simulations were very promising and meaningful since it was able to build connections from nodes that are highly related and political hubs, which was in the end, the main goal of this project.

In conclusion, this project involved many valuable steps in terms of tools used and analysis performed. We got our hands dirty at analyzing Social Networks and using the ecosystem of tools around it, from processing data to building and manipulating graphs. Finally, the custom simulation performed was a very interesting and challenging algorithm to build.

6 | Future Work

In this section we will talk about future improvements that can be applied in this work, in order to improve results and being able to perform a more realistic simulation.

First of all we need to introduce a very important problem that have made this work tremendously hard in a computational way. When we start to use Twitter API, we realise about one important thing. When you want to extract information from the social network you have to send queries to Twitter that contain the information you want to retrieve from it. For doing this, it is mandatory to have a Twitter registered account with a valid phone number associate to it. The problem is that Twitter has a limit of queries per interval of time, in order to avoid different types of attacks as for example a DoS. In this project we need to query information from a specific user in order to obtain its attributes as name, location, biography ... After this we need to query the social network again to obtain the people who this user is following. The maximum number of queries is different for each solicitude but it much more lower than we initially expect. This together with the filtering process we apply over the retrieved user have made very difficult to construct a large network. We spent many days trying to avoid this restriction searching different methods through the web but it was finally impossible. So although we leave our retrieving information algorithm run for many days the size of the dataset is not as big as we expected. So this can be considered clearly as a possible implementation, as we know, in data mining techniques the size of the dataset is highly important and in this work the influence in the results is clear. With more data we could have created a bigger graph and much more dense, this could have been highly interesting.

This fact have influence the development of the whole project so some of the ideas we had could not been carried out. One possible improvement could be to take into account more information obtained from Twitter as the interactions between users. It is possible to obtain the number of retweets that one user does from another one, the number of messages they exchange and more important data, that could be very useful. We could incorporate it to the graph, in a way that users which interact more will have an edge with a high weight, while as those users that are not usually in contact will have lower weighted edges.

Another future improvement could be to apply some Natural Language Processing techniques in the filtering process. We think that instead of filtering them using a classic bag of words approach, we could train a system to differentiate those users who are individual persons and are very related with the political world, also another kind of users that are very common in corruption problem as for example companies. In practically all the last cases of corruption that have been discovered in Spain, many construction company has been involved on it. So add them to our graph would be specially interesting and this could be achieved with a good preprocessing algorithm.

Bibliography

[1] Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. *ICWSM*, 10:178–185, 2010.

[2] Anders Olof Larsson and Hallvard Moe. Studying political microblogging: Twitter users in the 2010 swedish election campaign. *New Media & Society*, 14(5): 729–747, 2012.

[3] Twitter. Deep learning in a nutshell: History and training, 2016. URL `https://dev.twitter.com/rest/public`.

[4] ppmadrid. @ppmadrid twitter account, 2016. URL `http://twitter.com/ppmadrid`.

[5] Wikipedia. Anexo:municipios de la comunidad de madrid, 2016. URL `https://es.wikipedia.org/wiki/Anexo:Municipios_de_la_Comunidad_de_Madrid`.

[6] Wikipedia. PolÃŋticos de espaÃśa implicados en casos judiciales, 2016. URL `https://es.wikipedia.org/wiki/Anexo:Pol%C3%ADticos_de_Espa%C3%B1a_implicados_en_casos_judiciales`.