News Network Analysis

Gabriela HERNÁNDEZ, Maíra LADEIRA

INTRODUCTION

The main goal of this work is to build a network from a collection of texts, news & articles, and implement different network analysis methods. With this purpose, we constructed three networks by examining news and articles from mainstream relevant online newspapers. We aimed at gathering news from different sources to create two different kind of communities: one regarding the current and most important news and other one of categories such as politics, economics, sports, international articles and opinion articles. We analyzed each network by measuring their average shortest path, their average degree and their clustering coefficient. We also attempted to verify the effectiveness and correctness of the "Louvain" algorithm to find communities in our a news-network, in this case we attempted to find the communities in accordance to both the news and the categories.

DEVELOPMENT

Building the Networks

We have built three different networks by gathering different news from different newspapers. The nodes in all networks always represent the articles and we linked them by using the cosine similarity measure. In particular, we plan to compare topics across the most influential news websites.

In particular, we planned to compare topics across the most influential news websites, for this purpose we constructed three different networks by using articles from different news media sources, below we explain the content of each network created.

- Network 1: For this network, we gather 2026 articles from the CNN, USA Today, The Hoffman Post and NBC News websites
- Network 2: For constructing this network, we gathered 1179 articles from CNN and Hoffman Post websites
- Network 3: This is a sports network containing only 627 articles from ESPN, Fox Sports and Yahoo Sports websites.

Data Collection

For collecting the data and create our corpus we did not use a pre-built dataset, instead we downloaded the news and articles from different newspapers using the newspaper package[1], version 0.0.9.8, in Python 3, this library allowed us to download the articles and parse them in order to obtain their titles, texts and urls. Hence, for constructing each network, we created a "corpus" dictionary, containing all the articles from the specified newspapers, whose keys are the articles' titles and values are the articles' body [2]

. We have dumped each dictionary in "json" format for two reasons: to have them handy in order to analyze the future results and to have the same information during the elaboration of this work.

Constructing the nodes

Each node in a network represent an article, so when we created our "corpus" dictionary we created in parallel an identifier dictionary [3] whose keys are the IDs of each article and the values are their titles. In this manner, the network drawn can have these IDs as labels for the nodes and we can easily identify the articles by these labels.

Constructing the edges

We treated each article as a bag of words. For this purpose we created a special "tokenizer" function that removes punctuation and stop words from the text, then it lemmatizes the words and it returns a list of tokens. We used the "TfidfVectorizer" function from the sklearn library [4] to create the tf-idf vector for each article. For the tokenizer parameter, we used the already mention tokenizer function we created. Once we had the tf-idf vectors we computed the cosine similarity matrix by using the built cosine similarity function in sklearn [5]. Finally, with the cosine similarity matrix we constructed our network by using the Networkx [6] package in Python3, please notice that for our networks we are using undirected weighted edges, where the weight of each edge is the cosine similarity measure among two nodes, i.e. they represent the similitude between two articles.

Analysis of the Networks

For inspecting and interpreting the network analysis measures in our graphs, we have used the packages Netowrkx, community[7] and sklearn in Python 3.

Since we are dealing with a lot of articles from different media sources and we attempt to discover communities by categories and by news, we decided to apply different cosine threshold to our networks, i.e. if the cosine similarity measure between two articles is less than a certain threshold we dropped that edge.

Table I, table II and table III summarize the network analysis measures for the different cosine thresholds of the Network 1, 2 and 3 respectively. In these tables, we can see that when we do not use any threshold, both the clustering coefficient and the average degree are high, this can be explained because

[1]https://github.com/codelucas/newspaper/tree/python-2-head

[2]You can find these dictionaries in json format at one driver. The links are provided inside of the deliverable folder

[3]You can find these dictionaries in json format at one driver. The links are provided inside of the deliverable folder.

[4]http://scikit learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectori

[5]http://scikit-learn.org/stable/modules/metrics.html

[6]https://networkx.github.io/

[7]https://pypi.python.org/pypi/python-louvain/0.3

we are adding edges even when the similarity between two articles is very small (like 0.0003), thus it results in a very dense network with nodes very connected among them. In the news articles we can have some words that tend to appear in many articles regardless of the topic and sometimes can be more related with the category, this behavior also can explain the small shortest paths. So, for finding community units for categories we used a small threshold or even no threshold at all, and for finding communities for specific news and topics we use a bigger threshold. We would like to recall that for the Network 1 we gather the information among 2026 articles, for Network 2 among 1179 articles and for Network 3 among 627 articles.

Please notice, that for the network 1 we could not obtain the clustering coefficient nor the average degree when we did not use threshold, because of the amount of edges and nodes.

TABLE I
SUMMARY OF NETWORK ANALYSIS MEASURES FOR DIFFERENT COSINE THRESHOLDS IN THE NETWORK 1

Cosine Threshold	**Nodes**	**Edges**	**Average Shortest Path**	**Clustering Coefficient**	**Avg Degree**
0	2026	2021078	1.01	-	-
0.1	1811	29737	3.04	0.45	16.42
0.15	1322	13200	3.88	0.56	10
0.2	897	6960	4.43	0.65	7.75
0.25	622	3870	1.52	0.69	6.22

TABLE II
SUMMARY OF NETWORK ANALYSIS MEASURES FOR DIFFERENT COSINE THRESHOLDS IN THE NETWORK 2.

Cosine Threshold	Nodes	Edges	Average Shortest Path	Clustering Coefficient	Avg Degree
0	1179	681007	1.01	0.98	577.61
0.1	999	16355	2.96	0.5	16.37
0.15	676	7785	3.7	0.63	11.51
0.2	460	4137	3.42	0.7	8.99

TABLE III
SUMMARY OF NETWORK ANALYSIS MEASURES FOR DIFFERENT COSINE THRESHOLDS IN THE NETWORK 3.

Cosine Threshold	**Nodes**	**Edges**	**Average Shortest Path**	**Clustering Coefficient**	**Avg Degree**
0	627	195480	1	0.99	311.81
0.1	569	9633	2.59	0.59	16.92
0.15	480	3724	3.6	0.63	7.75
0.2	357	1834	5.09	0.67	5.13

Finding communities

For clustering the networks and finding their communities, we opted to use the "Louvain" method because it is proved to be efficient in networks with thousands nodes and can be applied in graphs with weighted edges as ours, we implemented this method with the help of the Community Package in Python3.

In each network, we got different number of partitions depending on the cosine threshold as we can seen in tables IV, V and VI. It is worth mentioning that as we are increasing the cosine threshold we are also getting more number of partitions, so it allows us to find communities more related to topics rather than specific categories.

TABLE IV
NUMBER OF PARTITIONS FOR EACH COSINE THRESHOLD IN NETWORK1

Cosine Threshold	**Partitions**
0	8
0.1	20
0.15	34
0.2	67
0.25	69

TABLE V
NUMBER OF PARTITIONS FOR EACH COSINE THRESHOLD IN NETWORK2

Cosine Threshold	**Partitions**
0	6
0.1	17
0.15	28
0.2	44

TABLE VI
NUMBER OF PARTITIONS FOR EACH COSINE THRESHOLD IN NETWORK3

Cosine Threshold	**Partitions**
0	7
0.1	13
0.15	18
0.2	22

RESULTS

Figures 1, 2 and 3 summarize our main results for each network. Among these networks, we can easily see that when we used a low threshold it was possible to get communities mostly related to categories whereas with a higher threshold more partitions were gotten, resulting in more topic-related communities.

Network 1:

For cosine threshold of 0.1, the main communities found were:"Sports", "Health", "Terrorism, Police and Shootings", "Traveling", "Christmas Holidays Topics", "Politics and Tech and Gadgets".

For cosine threshold of 0.25, the main communities found were: "Entertainment News: Miley Cirrus, Start Trek, Star Wars, Christmas topics ", "US Regions News", "Overweight, eat less meat, global warming", "Opinion Articles", "Financial, Funding and Wealth".

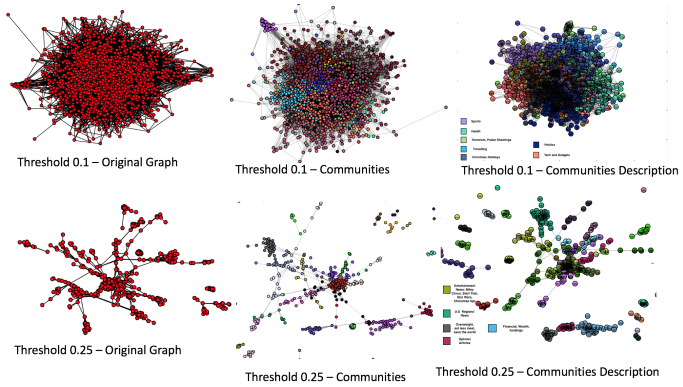

Fig. 1. Network 1 Results. From left to right, in the first row we have the results when we used a threshold of 0.1 whereas in the second row when we used a threshold of 0.25.

Network 2:

For cosine threshold of 0, the main communities found were:"Spanish News about entertainment and politics ", "English entertainment news", "Traveling", "Politics, Economics and Education", "War, ISIS, Syria".

For cosine threshold of 0.2, the main communities found were: "Arms, Weapons, Fires, killing", "Police News ", "Hollywood news, viral news traveling", "Pakistan, ISIS, Wars, Bombs", "Opinion Articles: Gay Mariage, New Years' eve', "Technology and science", "International News about money and agreements", "Expected events to happen in 2016 in the are of: Space launches (NASA), politics (mostly USA's New President) and Technology (New APPs to be launched) ", "International Opinion articles" and "Self-helping articles".

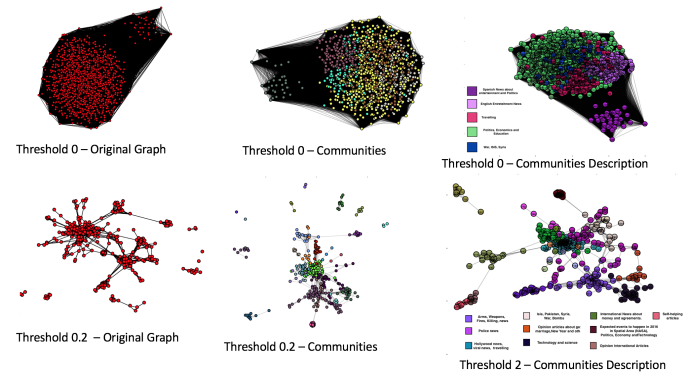

Fig. 2. Network 2 Results. From left to right, in the first row we have the results when we don't use threshold whereas in the second row when we used a threshold of 0.2

Network 3:

For cosine threshold of 0, the main communities found were:"Boxing", "NFL", "News about problems in sports such as immigrant players, doping and crisis".

For cosine threshold of 0.2, the main communities found were: "Novel news about soccer and football", "Tribute articles to players like Michael Jordan", "The best and more commented in sports during the 2015 year", "Articles about important athletes like Lionel Messi, Pau Gasol, Tiger Woods and John Lloyd " and "Sports Cups like Ryders cup, Davis Cup, Champions League".

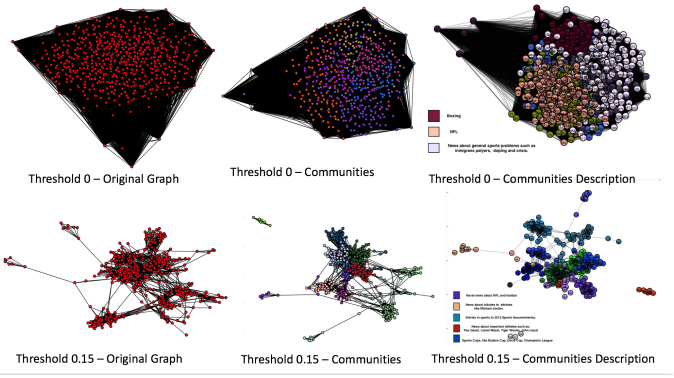

Fig. 3. Network 1 Results. From left to right, in the first row we have the results when we don't use threshold whereas in the second row when we used a threshold of 0.15

More plots and graphs are included in the delivered package into the folder "Figures".

CONCLUSIONS

In this lab-work we have demonstrated that by using together two different techniques in an information network, such as cosine threshold and Louvain method, we could get two different types of communities: category-related communities and topic-related communities. We have shown that as we increase the cosine threshold we get more topics-related communities rather than category-related communities, this can be easily shown in network 3, when the threshold is 0.15 in a community we got articles from famous athletes of different sports like Lionel Messi (football), Tiger Woods (Golf) and Pau Gasol(Basketball) whereas when the threshold was 0 we got specific categories as communities like soccer, NFL and so on. The same behavior is presented on networks 1 and 2.

It is also worth mentioning that as we have more nodes and more information in our network, the cosine threshold has to be increased for both: for getting the category-communities and for getting the topic-communities. Network 1 contains more than 2000 nodes, thus the threshold used for finding the category communities was 0.1 whereas in the other two networks was 0 and the threshold for finding the topic-communities was 0.25 meanwhile in the other two was 0.20 and 0.15.

For concluding, we would like to emphasize that one drawback of using the Louvain method for finding communities in our networks was that it is a non-overlapping community finding algorithm and in our case would fit better an algorithm that allows to overlap communities. Nevertheless, we could obtain clear results that allowed us to examine and analyze the behavior of our networks.

.