

Social network analysis sulle keywords del *The New York Times*

Achenia Antonio
a.achenia@studenti.unipi.it
Student ID: 589563

Di Mauro Gianmarco
g.dimauro3@studenti.unipi.it
Student ID: 587959

Cuozzo Silvia
s.cuozzo1@studenti.unipi.it
Student ID: 587958

Scalisi Marta
m.scalisi2@studenti.unipi.it
Student ID: 587941

ABSTRACT

In questo paper viene analizzata la topologia di rete delle keywords estratte dagli articoli del *The New York Times* pubblicati tra Luglio e Ottobre, nei mesi precedenti alle elezioni americane. Abbiamo esaminato i mesi precedenti alle elezioni presidenziali americane, nel periodo da Luglio ad Ottobre. Nella nostra rete, i nodi rappresentano le keywords e i link gli articoli: due keywords che compaiono nello stesso articolo saranno collegate da un arco. Inoltre, per ogni link è stato calcolato il peso considerando le frequenze delle coppie di keywords.

Il progetto si articola nelle seguenti fasi: *Data collection*, *Basic network analysis*, *Community discovery*, *Spreading*, *Supervised link prediction* e focus sui nodi relativi ai candidati Trump e Biden.¹

KEYWORDS

Social Network Analysis, The New York Times, Elezioni USA 2020, Cronaca, Covid-19.

ACM Reference Format:

Achenia Antonio, Cuozzo Silvia, Di Mauro Gianmarco, and Scalisi Marta. 2020. Social network analysis sulle keywords del *The New York Times*. In *Social Network Analysis '20*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Nel seguente report commenteremo i risultati ottenuti dallo studio della rete costruita attraverso le keywords estratte da un campione di articoli presenti sulla pagina del "The New York Times".² La rete è costituita da 19.025 nodi e 232.168 archi. I nodi sono le keywords e i link gli articoli: due keywords che compaiono nello stesso articolo saranno collegate da un arco. La rete è un grafo indiretto e pesato. Il peso è calcolato a partire dalla frequenza delle coppie di parole. Per la realizzazione del progetto sono stati utilizzati: Python

¹Project Repositories

Data Collection: <https://github.com/sna-unipi/data-collection>
Analytical Tasks: <https://github.com/sna-unipi/analytical-tasks>

Report: <https://github.com/sna-unipi/project-report>

²<https://www.nytimes.com/>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SNA '20, 2019/20, University of Pisa, Italy

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$0.00
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

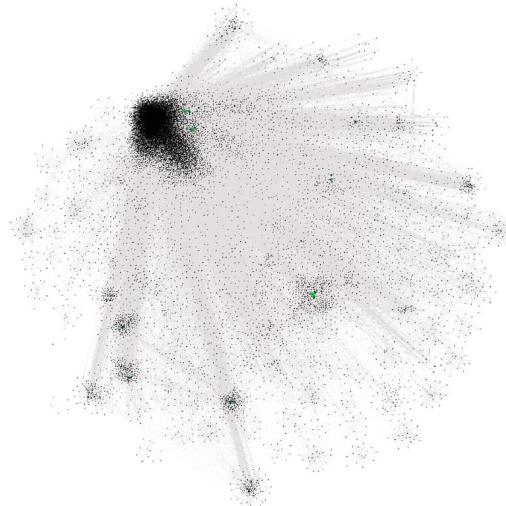


Figure 1: Network visualization

in Google Colab e Ambiente Jupyter, con l'aiuto delle biblioteche NetworkX, CDlib e NDlib; Gephi per la rappresentazione visuale del grafo.

Per l'approfondimento dello studio effettuato, inizieremo descrivendo lo *scraping* dei dati, seguito da una descrizione delle caratteristiche della rete, nello specifico *degree distribution*, *connected components*, *path*, *clustering coefficient*, *density* e *centrality analyses*. Le stesse analisi sono state effettuate anche sui modelli sintetici ai fini di un confronto.

Il lavoro procede con lo studio della community discovery, spreading, supervised link prediction e uno studio focalizzato su una probabile polarizzazione dei nodi di Trump e Biden.

2 DATA COLLECTION

I nostri dati sono stati estratti dal *The New York Times* ricavando le keywords dai metadati del codice HTML di tutti gli articoli degli ultimi quattro mesi. Inizialmente avevamo pensato di considerare l'arco temporale di un anno ma, per motivi computazionali abbiamo optato per conservare il focus su un periodo sensibile come quello delle elezioni presidenziali americane.

Considerando i singoli articoli abbiamo collegato con un link tutte le parole chiave presenti, eseguendo questo processo iterativamente abbiamo costruito la rete e abbiamo assegnato ai collegamenti che occorrevano più volte un peso uguale alla loro frequenza. Per motivi strutturali, la rete risulta *undirected*: non ci sono gerarchie tra i nodi.

2.1 Crawling Methodology and Assumptions

Per lo scraping, i passaggi sono stati i seguenti:

- Inserimento del titolo degli articoli e dei relativi tag in una Dataset;
- estrazione delle keywords dal dataset;
- Trasformazione delle keywords in un dizionario in modo da eliminare i tag ripetuti;
- Assegnazione di un ID univoco ad ogni keywords;
- Creazione della *edgelist*;
- Calcolo delle frequenze e inserimento del peso;
- Rimozione degli archi ripetuti e opposti (ad esempio nel caso di (A,B) e (B,A) abbiamo sommato le frequenze e rimosso una delle due versioni).

3 NETWORK ANALYSIS

Completato lo scraping dei dati, abbiamo utilizzato Gephi, un tool che ha permesso sia di visualizzare il grafo sia di esplorarlo ai fini di uno studio consapevole e preciso della rete.

Parametri	Valori
Num. di nodi	19027
Num. di edges	232169
Weighted	Sì
Directed	No
Density	$6.41 \cdot 10^{-4}$
Num. selfloops	0
Average clustering coefficient	0.428

Table 1: Characteristics of RW network

Le statistiche computate sulla *Real World Network* sono state confrontate con quelle ottenute dai modelli sintetici aventi lo stesso numero di nodi e di archi. I modelli sono stati creati con gli algoritmi predefiniti.

Erdos-Rényi (ER) è un modello randomico, viene utilizzato quando abbiamo una mancanza contingente di dati, ad esempio si conosce soltanto il numero di nodi ed archi, senza nessuna correlazione tra di essi. Il modello ER è caratterizzato da una *Poisson distribution*, da un *path lenght* breve e da un *cluster coefficient* piccolo. Inoltre, è importante sottolineare come in una ER non ci siano outliers.

Il *Configuration Model* è utilizzato per avere una rete con una degree distribution che si avvicini quanto più possibile a quella reale, dà origine ad un grafo direzionato, in cui possiamo definire una *in-degree* e una *out-degree*, in questi casi la degree totale è la somma delle due appena citate. Le proprietà di questo modello sono un path lenght breve e un cluster coefficient piccolo, il quale è indipendente dalla dimensione della rete.

Il *Barabasi-Albert* model (BA) permette di avere una *scale free distribution* ed è l'unico modello ad avere una degree distribution che segue una *power law* in grado di creare *hubs*. Il path lenght e il cluster coefficient sono valori generalmente piccoli. Inoltre, il BA possiede due proprietà delle reti reali ovvero la crescita e il *preferential attachment*.

Infine, l'ultimo modello sintetico generato è il *Watts-Strogatz* (WS),

il quale permette di spiegare i fenomeni de "Small World". In particolare, esso si focalizza sull'*high clustering*, ovvero sul concetto delle reti organizzate in cluster con nodi fortemente connessi e minima connessione con gli altri cluster. Le sue proprietà sono: una degree distribution di tipo poissonian, il path lenght breve ed un cluster coefficient ampio.

	Numero di nodi	Numero di edges
RW	19027	232169
ER	19027	232169
CM	19027	223535
BA	19027	76092
WS	19027	76108

Table 2: Nodes and Edges of the networks

3.1 Degree Distribution

Nella fase della comparazione della degree distribution, notiamo come il modello che più si avvicina alla rete reale sia il Configuration model, questo è facilmente spiegabile dal momento che, come è stato detto in precedenza, è un modello che viene utilizzato per avere una degree distribution quanto più simile a quella reale. Inoltre, un'altra similitudine è riscontrata con il BA model.

Nella nostra rete, vi è un rapporto inversamente proporzionale tra il numero dei nodi della rete e il degree ovvero all'aumentare della degree osserviamo una diminuzione del numero dei nodi. Un ulteriore studio per quanto concerne la degree dei nodi, ci ha permesso di scoprire che il nodo con degree maggiore è *Coronavirus (2019-nCoV)* con un valore pari a 5006; invece il secondo nodo per degree è *USA Politics and Government* con un valore pari a 2834. In seguito, è stato calcolato il numero di edges del path tra i due che risulta essere pari a 2. L'average degree del grafo è 24,40416251 e si ottiene dal rapporto tra il doppio del numero degli edges e il numero dei nodi, in quanto è un grafo undirected.

$$\langle k \rangle \equiv \frac{2L}{N} \quad (1)$$

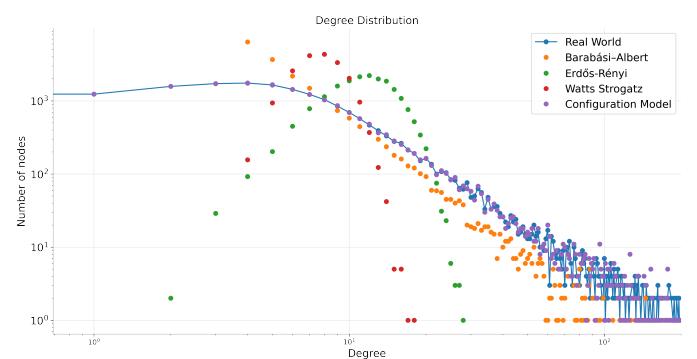


Figure 2: Degree Distribution

3.2 Connected Components

Per poter confrontare le *connected components*, per prima cosa abbiamo convertito i grafi *directed* in *undirected*.

Le componenti dei nostri modelli sono:

- ER network: 1
- BA network: 1
- CM network: 1
- WS network: 1
- Real world network: 4

Nella nostra rete sono state individuate quattro diverse componenti, mentre le altre ne presentano solo una. Inoltre, solo per il *directed CM model* sono state considerate le weakly (1) e le strongly (3595) components.

3.3 Path Analysis

Un *path* è rappresentato da una sequenza di nodi adiacenti. In un grafo directed, il path può seguire soltanto la direzione di una freccia, a differenza del tipo undirected.

L' *average shortest path* è definita come la media dei percorsi più brevi tra tutte le coppie di nodi. Per il modello CM abbiamo considerato le weakly connected components, invece per gli altri, compresa la nostra rete, abbiamo calcolato soltanto le connected components basilari.

	RW	ER	BA	WS	CM
Numero di nodi	19006	19027	19027	19027	19027
Avg short path	2	4	4	5	2
Comp. connesse	4	1	1	1	1

Table 3: Average shortest path

Il numero dei nodi si riferisce a quello delle componenti più grandi. Questa cernita è stata effettuata soltanto per la RW dal momento che ne presenta 4, le altre tre non sono state tenute in considerazione perché hanno un average shortest path pari a 0.

3.4 Clustering Coefficient

Il *Clustering coefficient* può essere sia globale sia locale, in ogni caso è un valore compreso tra [0,1]. Nel primo caso, ci indica quanto è clusterizzata la rete. Nel secondo caso invece, specifica quanto sono connessi i vicini di un nodo. Infine, l' *average clustering coefficient* calcola la media dei LLC presenti nella rete.

I risultati ottenuti dall'analisi di ciascuna rete, calcolando i relativi valori di: minimo, massimo, media e deviazione standard sono i seguenti:

	RW	ER	BA	WS	CM
min	0.000000	0.000000	0.000000	0.000000	0.000000
max	0.500000	0.014286	0.500000	0.800000	1.000000
mean	0.427584	0.000651	0.003533	0.083539	0.058951
stdev	0.123593	0.001148	0.020121	0.078097	0.070937

Table 4: Clustering Coefficients

I valori della nostra rete possono considerarsi accettabili, vicini al modello WS per quanto riguarda il max.

Per quanto riguarda invece i valori del ER model, osserviamo come siano valori molto bassi, ciò è comprensibile considerando che esso

differisce dalle reti reali per il tipo di degree distribution e per il clustering coefficient, inoltre maggiore è la dimensione della rete minore sarà quest'ultimo.

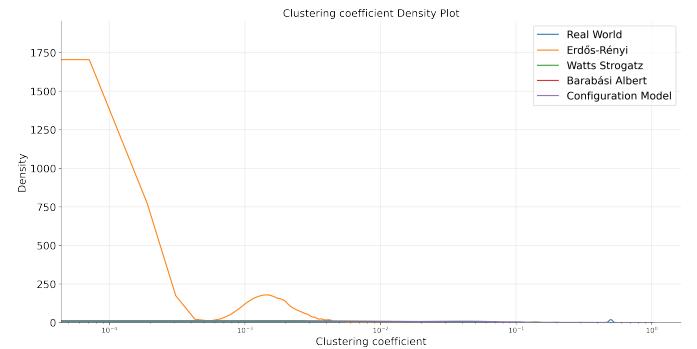


Figure 3: Clustering Coefficients distribution

Dall'analisi della distribuzione del *Clustering Coefficient* mostrata in Figura 3 è possibile osservare che ER presenta un numero molto alto di nodi con degree basso. Infatti, nel range che va da 0 a 0.0001 abbiamo il maggior numero di nodi con il minor clustering coefficient possibile e notiamo che questo cresce rapidamente al diminuire del numero di nodi.

Per questo motivo, abbiamo ritenuto opportuno, al fine di una migliore analisi, rimuovere questo modello i cui valori appiattivano quelli degli altri, in modo da farli emergere e osservarli nello specifico.

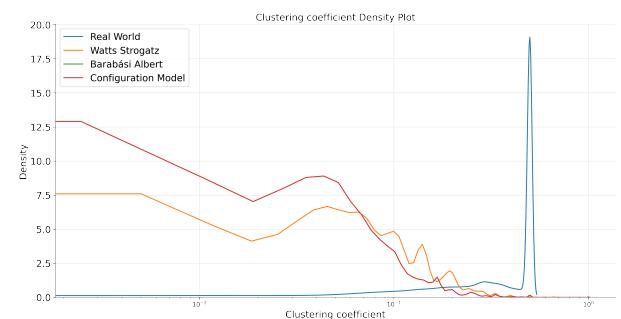


Figure 4: Clustering Coefficients distribution (particular)

La Figura 4 mostra che i modelli WS e CM presentano un andamento simile: al diminuire della densità si ha un aumento del clustering coefficient. La Real World network, invece, presenta un andamento differente. In particolare, inizialmente si ha un livello di densità molto basso e i nodi presentano valori costanti di clustering coefficient (nel range da 0 a 0,01). Da questo punto in poi si verifica una crescita graduale di questo valore all'aumentare della densità con un successivo picco.

3.5 Density analysis

La densità è importante ai fini dello studio di un grafo. Il suo valore è ottenuto mediante la seguente formula:

$$d(G) = \frac{L}{L_{max}} \quad (2)$$

Un grafo è definito denso quando assume un valore pari ad 1, cioè tutti i nodi sono connessi tra loro. La maggior parte delle reti reali sono sparse, infatti anche la nostra rete risulta essere tale. Di seguito, i valori delle densità ottenuti.

	Densità
RW	$6.415 \cdot 10^{-4}$
ER	$6.413 \cdot 10^{-4}$
CM	$6.174 \cdot 10^{-4}$
BA	$4.203 \cdot 10^{-4}$
WS	$4.204 \cdot 10^{-4}$

Table 5: Densità delle reti

La densità della rete RW risulta essere più alta rispetto ai modelli BA e WS, ma molto simile a quella di ER e CM.

3.6 Degree Centrality

Il degree di un nodo è fondamentale per definire la sua rilevanza all'interno della rete: più questo è alto, maggiore sarà la sua importanza. Nella nostra rete il nodo con più alto degree (5006) si è rivelato essere "Coronavirus (2019-nCoV)", seguito da "United States Politics and Government" (2834) e "Books and Literature" (2770).

Dal confronto fatto tra i modelli, presenti in Figura 5, emerge che per i valori più bassi di Degree Centrality i modelli WS e ER presentano i valori più alti di densità.

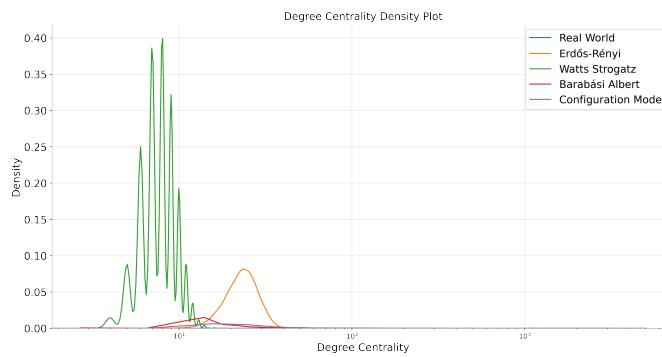


Figure 5: Degree Centrality

3.7 Connectivity - Based centrality

In questa sezione, analizzando i risultati ottenuti sia con l'algoritmo *Eigenvector* sia con l'algoritmo *PageRank*, abbiamo avuto modo di constatare che i valori restituitici seguivano lo stesso pattern: il CM e il BA sono i modelli che approssimano meglio la nostra *RW Network*, mentre ER e WS presentano una distribuzione che si

discosta in modo evidente rispetto agli altri.

Un'ulteriore prova effettuata è stata quella di utilizzare l'algoritmo *Katz*, tuttavia quest'ultimo non riusciva a convergere.

Eigenvector Centrality. I nodi con più alto valore di Eigenvector sono, anche in questo caso "Coronavirus (2019-nCoV)"(0.194), seguito da "United States Politics and Government" (0.158), "Trump, Donald J"(0.155).

Considerando che, secondo la *Recursive Definition*, i nodi importanti sono connessi ad altri nodi importanti ci siamo chiesti se effettivamente se le nostre keywords fossero connesse tra loro e la nostra domanda ha ottenuto un esito positivo.

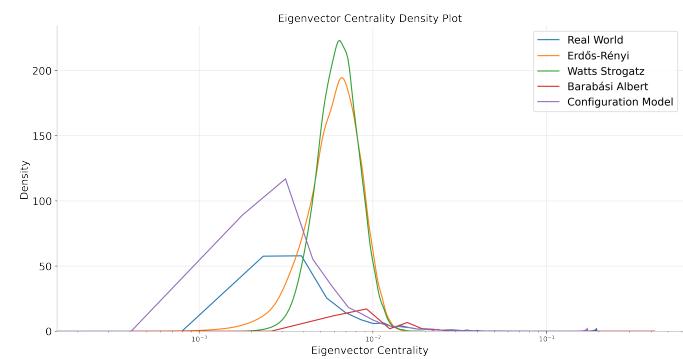


Figure 6: Eigenvector Centrality

PageRank Centrality. Il PageRank è un algoritmo creato da google per valutare l'importanza delle pagine Web. L'algoritmo non considera tutti i vicini di un nodo ma ne sceglie, in modo randomico, un sottoinsieme.

Anche in questo caso, il nodo che ha ottenuto uno score maggiore è "Coronavirus (2019-nCoV)"(0.009), seguito dai nodi che rappresentano le keywords: "Book and Literature" (0.008) e "Movies" (0.006).

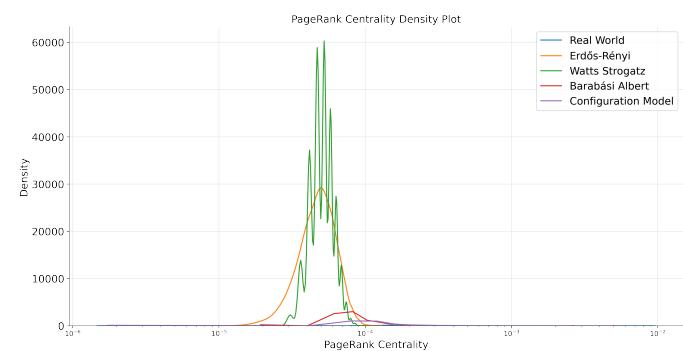


Figure 7: PageRank Centrality

3.8 Geometric centrality

Closeness Centrality. In questo caso, i nodi più centrali all'interno del grafo saranno quelli con più alta closeness, quindi i nodi che hanno una distanza media minore con tutti gli altri nodi. Nel nostro caso, "Coronavirus (2019-nCoV)"(0.572), "United States Politics and

Government "(0.531), "Trump, Donald J" (0.528) sono i nodi che hanno ricevuto lo score più alto.

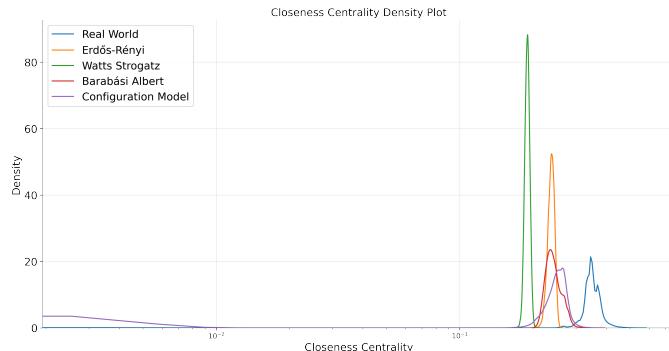


Figure 8: Closeness Centrality

Harmonic Centrality. Calcolando per ogni nodo la media armonica delle shortest paths con gli altri nodi della rete ad essi direttamente collegati, è stato possibile notare che "Coronavirus (2019-nCoV)"(11980.000) è nuovamente il nodo che ha ricevuto lo score più alto. Seguito da "United States Politics and Government"(10831.333) e "Book and Literature"(10758.416). Com'è possibile notare i risultati non differiscono molto da quelli ottenuti nella sezione precedente, con prevalente riferimento alle elezioni politiche americane e coronavirus, due eventi cruciali del 2020.

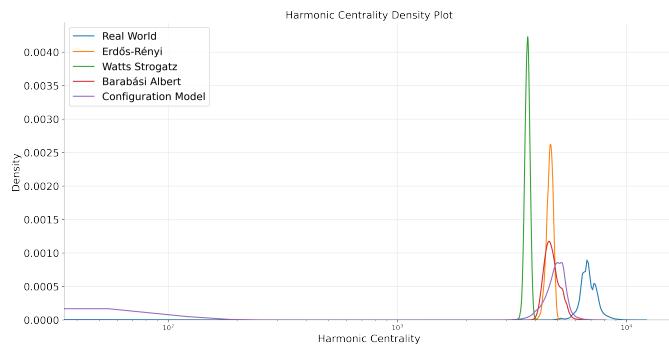


Figure 9: Harmonic Centrality

Betweenness Centrality. L'importanza di un nodo viene determinata dal numero degli shortest path che lo attraversano. I valori più alti di Betweenness Centrality sono stati ottenuti con i seguenti nodi: "Coronavirus (2019-nCoV)"(27097256.839), "Books and Literature"(27097256.839) e "Movies"(13597213.616).

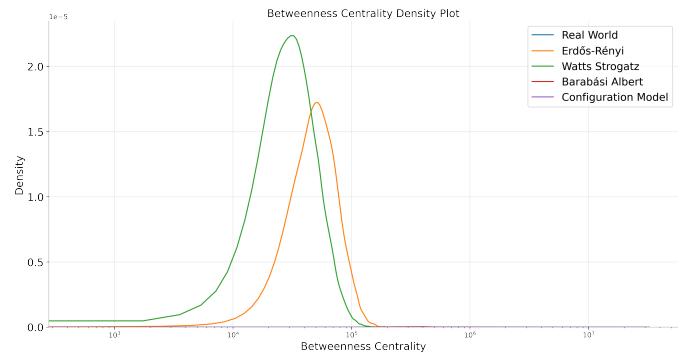


Figure 10: Betweenness Centrality

Confronto Finale. Confrontando la distribuzione delle Degree and Connectivity-Based Centralities, abbiamo avuto modo di constatare che i risultati ottenuti differiscono tra loro.

Nonostante questo, il nodo che si è maggiormente distinto è "Coronavirus (2019-nCoV)", che in ogni algoritmo ha sempre ottenuto il valore più alto. Nelle successive posizioni, invece, nonostante siano state individuate delle keywords differenti, risulta evidente che, nelle maggior parte dei casi, queste facciano riferimento alla politica americana.

In conclusione, è possibile affermare che questi risultati sono prevedibili dal momento che, in questi mesi, gli argomenti maggiormente discussi sui giornali fanno riferimento alla pandemia attualmente in corso e alle elezioni presidenziali americane.

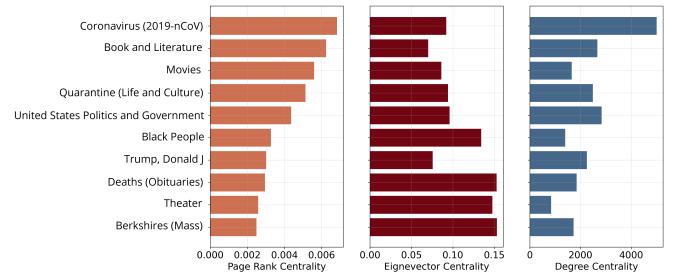


Figure 11: Degree and Connectivity-Based Centralities

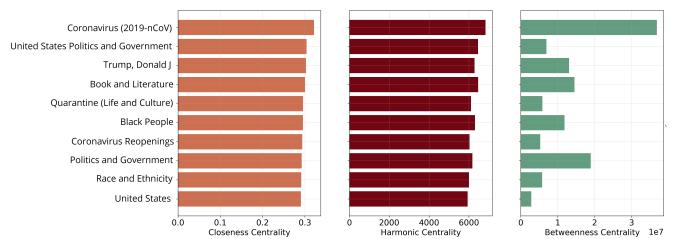


Figure 12: Geometric Centralities

4 TASK 1: COMMUNITY DISCOVERY

La Community Discovery ha lo scopo di identificare comunità, cioè topologie su media scala nascoste nella complessa struttura della rete. Poichè non esiste una definizione univoca di "community", ogni algoritmo modella i cluster secondo proprietà differenti. Per agevolare la computazione del codice non abbiamo considerato i pesi della nostra *undirected network*.

Gli algoritmi da noi utilizzati sono stati:

- **CNM**; un algoritmo proposto da Clauset, Newman e Moore che utilizza il valore di modularità per l'identificazione dei cluster.
- **LPA (Label Propagation Algorithm)**; che assegna ad ogni nodo un identificativo che si propaga attraverso la rete. Ad ogni iterazione dell'algoritmo, il nodo aggiorna il suo valore con quello maggioritario rispetto ai nodi vicini, fino a convergenza.
- **Louvain**; algoritmo che massimizza la modularità delle comunità.
- **Demon**; l'unico algoritmo utilizzato che individua cluster non *overlapping*.

Tutti gli algoritmi sono stati importati dalla libreria *CDlib*, per python. In prima istanza è stato eseguito anche l'algoritmo K-Clique, che utilizza invece la percolazione, ma data la complessità dell'algoritmo il tempo di analisi risultava troppo elevato.

	GM	Labels propagation	Louvain	Demon
Comunità	279	675	21	421
Nodi	19025	19025	19025	18626
Degree	5.72	4.14	12.37	28.07
Density	0.22	0.23	0.07	0.04
Size	68.18	28.18	905.95	690.42
Edges	638.06	329.25	6865.28	15965.99
Cut ratio	0.000164	0.000106	0.000318	0.007518
Modularity	0.1109	0.1384	0.0898	0.8577
Conduct.	0.380143	0.379267	0.270942	0.779915

Table 6: Evaluation of algorithms (mean values)

Dalla tabella si può evincere che il valore più alto di modularità è stato ottenuto dall'algoritmo Demon, possiamo definire questa partizione ottimale; con Labels propagation il valore risulta essere medio, gli altri due hanno registrato valori negativi.

I primi tre algoritmi hanno utilizzato tutti i nodi per individuare le comunità, invece Demon ne ha esclusi circa 500, probabilmente per questo motivo mostra un degree medio più elevato.

Osserviamo che all'aumentare della dimensione media dei cluster diminuisce sensibilmente la densità.

4.1 Confronto tra le partizioni

Considerando che *Normalized Mutual Information* (NMI) necessita una copertura totale dei nodi, è stata applicata solo sugli algoritmi di Greedy Modularity, Label Propagation e Louvain.

	GM	LP	Louvain
GM		0.44	0.39
LP	0.44		0.27
Louvain	0.39	0.27	

Table 7: Normalized Mutual Information

Il NMI ci restituisce un valore compreso tra [0,1] quindi più le partizioni sono simili maggiore è il suo valore. Dal confronto delle partizioni, visibile in Tabella 7 è possibile affermare che le partizioni sono abbastanza dissimili.

	GM	LP	Louvain	Demon
GM		$4.4 \cdot 10^{-2}$	$8.1 \cdot 10^{-3}$	$1.9 \cdot 10^{-2}$
LP	$4.4 \cdot 10^{-2}$		$5.0 \cdot 10^{-4}$	$3.6 \cdot 10^{-2}$
Louvain	$8.1 \cdot 10^{-3}$	$5.0 \cdot 10^{-4}$		$1.1 \cdot 10^{-2}$
Demon	$1.9 \cdot 10^{-2}$	$3.6 \cdot 10^{-2}$	0.07	

Table 8: NF1 score matrix

I risultati ottenuti con la NF1 score invece, sono visibili nella Tabella 8. Da questi valori possiamo vedere che, confrontando gli algoritmi di Label Propagation, Louvain e Greedy Modularity visti in precedenza, i risultati sono molto diversi da quelli che ci si sarebbe dovuti aspettare leggendo i valori ottenuti con la NMI. Considerando infine che tutti i valori di NF1 score sono prossimi allo zero, possiamo affermare che non abbiamo ottenuto risultati simili con nessun algoritmo utilizzato.

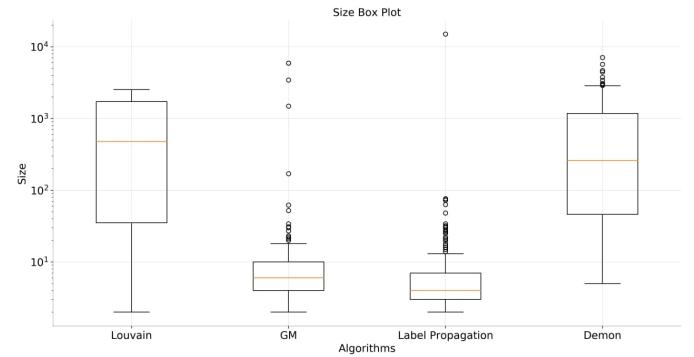


Figure 13: Boxplot of sizes

Il boxplot in Figura 13 mostra la grandezza media delle comunità trovate dagli algoritmi di *Community Discovery*. Le comunità trovate in Louvain e Demon presentano in media un numero di nodi maggiore rispetto a quelli ottenuti tramite gli algoritmi GM e Label Propagation, che invece presentano un minor numero di nodi distribuiti in piccoli range.

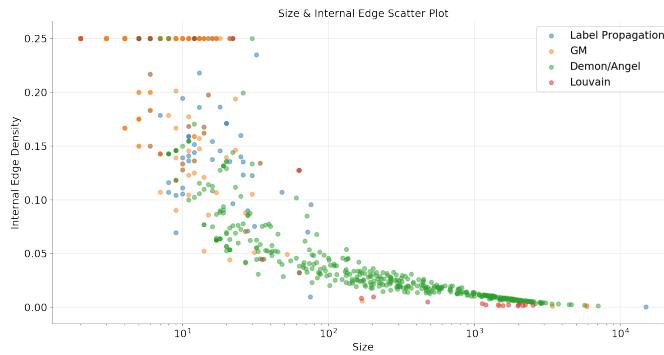


Figure 14: Scatter plot Size vs Internal Edge Density

La Figura 14 mostra che la grandezza di una comunità è inversamente proporzionale alla densità degli archi.

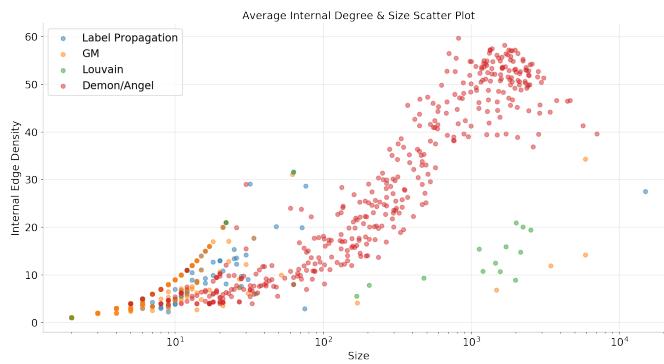


Figure 15: Scatter plot Size vs Average Internal Degree

In questo caso vediamo invece che: quando si hanno un numero basso di nodi all'interno delle comunità, si ha anche un basso valore di *average degree*; quando invece il numero di nodi al loro interno cresce, si ha anche un aumento dei valori di *average degree* che, in alcuni punti, tende a distribuirsi in modo irregolare.

4.2 Analisi semantica dei cluster

Da un punto di vista meno tecnico, è interessante osservare la composizione dei cluster individuati per comprendere gli aspetti "non-matematici" delle relazioni tra i nodi. Nel dettaglio, nel nostro caso è lecito chiedersi se all'interno dei cluster si sia mantenuto un legame semantico tra le parole chiave presenti.

Data la grande mole di cluster individuati, abbiamo deciso di analizzare velocemente soltanto quelli ottenuti dagli algoritmi Greedy Modularity e Label Propagation. Abbiamo escluso dalla nostra analisi quelli ottenuti con Louvain, per il basso valore di modularità, e quelli ottenuti con Demon perché si ha *overlapping* tra i cluster, e quindi idealmente meno propensi a mantenere una stretta correlazione semantica.

Come si evince già dal boxplot in Figura 13 i cluster ottenuti secondo questi due algoritmi seguono una distribuzione similare: poche comunità molto grandi, con più di mille keywords e moltissime con

meno di 10 componenti. In entrambi i casi il cluster più grande contiene al suo interno molte delle parole "centrali" alla rete, come "Covid", "Trump", "Black People", "USA Politics and Government", "Quarantena" e "Razza ed etnia": sebbene queste parole non appartengano tutti allo stesso campo semantico, potremmo considerarle come appartenenti al macrosettore della "Cronaca" o "Attualità". Sono i nodi all'interno del "*nucleo*" della nostra rete, le parole più ricorrenti tra gli articoli presi in considerazione, i temi caldi degli ultimi mesi. Altre parole all'interno di questi cluster sono: "private equity", "veterani", "Bangladesh" e "Association for the Advancement of Colored People".

I cluster successivi ricavati dal Label Propagation sono invece molto meno numerosi: il secondo conta appena 76 nodi. In generale, queste comunità raccolgono le parole chiave di due o tre articoli, collegati tra loro da una parola più ricorrente. Ad esempio, sono presenti un piccolo cluster di nomi di scrittori e uno al cui centro vi è la keyword "parole crociate" collegata con diverse altre parole che supponiamo essere le soluzioni. In questi casi delle strutture semantiche potrebbero essere individuate, ma facendo riferimento ad un numero così ristretto di articoli l'associazione non può essere che banale e insoddisfacente. Anche la maggior parte dei cluster ottenuti con il Greedy Modularity presentano queste caratteristiche - ovvero l'essere molto piccoli e riconducibili a pochi articoli - eccetto che per quei pochi che contano al loro interno più di 100 nodi, appartenenti tutti ai medesimi campi semantici.

Nel secondo cluster ottenuto, di dimensione 5.681 nodi, tutte le keywords fanno riferimento al campo dell'arte e dell'intrattenimento. La parola con il grado di centralità più alto è "Libri e letteratura", ma sono presenti anche "Film", "Teatro", "Arte", "Pop and Rock Music", "Musica", "Danza", "Musica classica" fino a enumerare diversi film, festival e attori vari.

Il terzo, invece, ha come tema principale lo sport. Le parole più centrali sono tutte squadre di football americano - "Tampa Bay Buccaneers", "Dallas Cowboys", "San Francisco 49ers" - ma all'interno della comunità troviamo anche "sci", "olimpiadi 2022" e anche il nome di un cavallo da corsa, "Swiss Skydiver". Separatamente troviamo anche un ennesimo cluster a tema sportivo, il cui argomento principale è però esclusivamente il calcio. In particolare, il nodo più centrale è "UEFA Champions League", contornato da nomi di squadre di calcio - "Real Madrid (Soccer Team), Barcelona(Soccer Team), Bayer Monaco (Soccer Team)" - e di nomi di allenatori, come "Bartomeu Josep Maria".

5 TASK 2: SPREADING

L'andamento dei contagi durante un'epidemia può rivelare molte informazioni utili riguardo una rete sociale e il tipo di relazioni peculiari che la contraddistinguono.

Storicamente le epidemie sono state riconosciute e studiate analiticamente assimilando l'evolversi del contagio all'andamento della curva disegnata da una funzione logistica.

La portata e la velocità del contagio possono avere connotazioni di significato più ampie di quella biologica e medica; le epidemie sono in primis legate a fenomeni biologici come la diffusione di virus e batteri, parassiti, trasmessi con contatti fisici o per via aerea, ma possono in parallelo essere accostate anche all'universo informatico, con la diffusione di virus, malware ed exploit di bug, e all'universo

socio-mediatico, con lo scambio e la circolazione di informazioni, notizie e codici tra canali principali e minori, persone ed organizzazioni.

L'analisi dello spreading ha come obiettivo inquadrare un'eventuale relazione tra struttura delle reti e la probabilità di contagio tra i nodi della rete con la portata dell'epidemia, espressa in numero totale di contagiati, attraverso l'applicazione di quattro modelli di diffusione sulla nostra rete e i due modelli sintetici ER e BA.

I grafici e i risultati sono stati ottenuti e comparati grazie alla libreria Python NDLIB.

I modelli in questione, che verranno singolarmente presi in esame, sono:

- SI, susceptible-infected;
- SIS, susceptible-infected-susceptible;
- SIR, susceptible-infected-recovered;
- Threshold model.

5.1 SI: susceptible-infected

Il modello SI ha due stati caratteristici, Suscettibile (S) e Infetto (I); nel modello si utilizza il parametro β , atto a rappresentare il tasso di contagio, ovvero come l'infezione si diffonda tra la popolazione al variare del tempo (t).

Nel modello in questione la crescita epidemiologica sarà inizialmente esponenziale poiché tutti i nodi vengono considerati come suscettibili, e col passare del tempo, tutta la popolazione risulterà contagiata/infetta.

Al crescere di (t), poiché sempre meno individui suscettibili possono essere contagiati rispetto alla parte già infetta della popolazione, la curva di contagio cresce sempre più lentamente, evidenziando il comportamento logistico del fenomeno, fino al cessare dell'infezione che coincide con il completo contagio della popolazione in esame se si estendesse all'infinito.

Parametri: $\beta: 0.015$; %infected: 0.05; tp rate: 1; fraction infected: 0.05

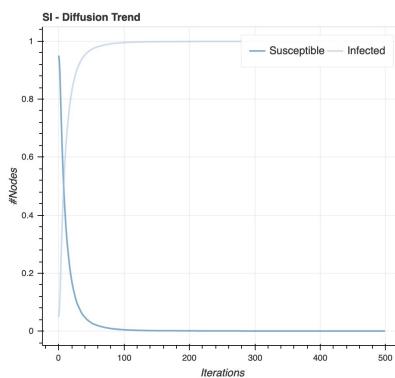


Figure 16: SI: Real Network

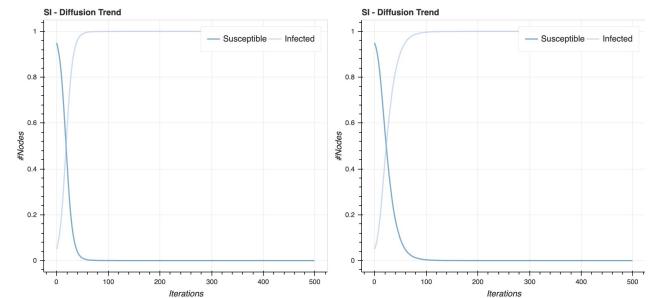


Figure 17: SI: ER and BA

5.2 SIS: susceptible-infected-susceptible

Il modello SIS ha due stati; la differenza rispetto al modello precedente è che lo stato Infetto ha la possibilità di ripresentarsi come Suscettibile dopo il contagio. Oltre al parametro β che indica il tasso di virulenza, è presente anche il parametro μ , che rappresenta il tasso di guarigione.

Dal rapporto tra il tasso di contagio e il tasso di guarigione β/μ , si può ricavare il parametro λ che rappresenta il numero basico di riproduzione del virus, anche definibile come il numero medio di contagiati generato da un singolo individuo infetto presente all'interno di una popolazione considerata completamente suscettibile. Il parametro λ indica la pericolosità di diffusione del virus in relazione ai parametri di infezione e guarigione. Quando $\lambda > 1$, si verificherà l'epidemia, diversamente con $\lambda < 1$ il contagio si estingue.

I contagi crescono esponenzialmente nel brevissimo periodo, più velocemente nel caso della rete NYT2k20, ma in maniera molto simile agli altri due casi.

Parametri: $\beta: 0.008$; $\lambda: 0.01$; %infected: 0.05; tp rate: 1; fr. infected: 0.05

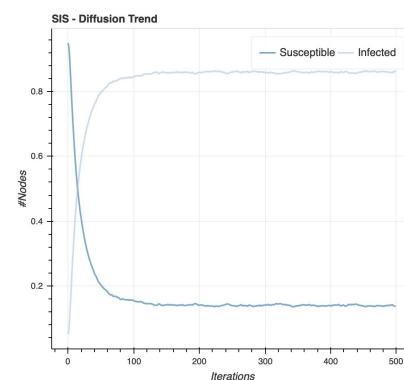


Figure 18: SIS: Real Network

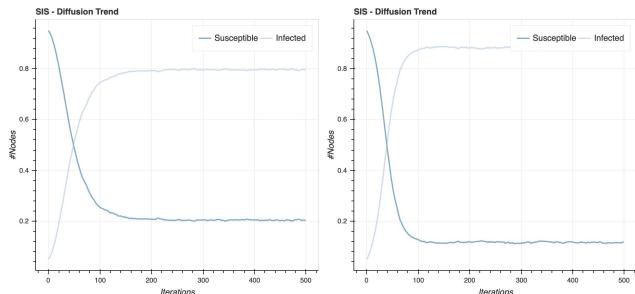


Figure 19: SIS: ER and BA

5.3 SIR: susceptible-infected-recovered

Il modello SIR ha tre stati, con lo stato R, ovvero Recovered (o Removed) che indica il cambio di stato di un nodo infetto grazie ad immunità naturale, un vaccino somministrato, oppure un conseguente decesso per malattia.

Dai grafici si evince come inizialmente, al presentarsi di popolazione suscettibile, l'epidemia cresca esponenzialmente nel breve periodo, per poi attenuarsi nel lungo periodo al crescere della curva R, che di fatto "sottrae" ai virus individui contagibili, con un comportamento del fenomeno che tende alla saturazione come visto nel modello SI.

Parametri: $\beta: 0.01$; $\gamma: 0.006$; %infected: 0.05; tp rate: 1; fr. infected: 0.05

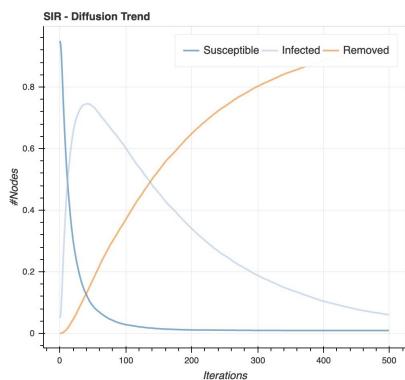


Figure 20: SIR - Dual Network

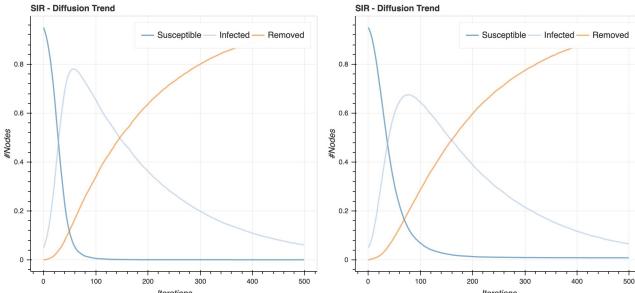


Figure 21: SIR: ER and BA

5.4 Threshold model

L'ultimo modello testato è il modello Threshold, in cui ogni nodo ha una soglia caratteristica che, se superata, rende il nodo suscettibile "infetto". È presente una percentuale di nodi inizialmente infetti, a rappresentanza dell'inizio della diffusione epidemiologica: il parametro α indica i primi individui di popolazione che infettano i vicini al superamento del valore soglia impostato.

L'andamento delle curve ottenute si è rivelato pressoché analogo a quello visto nel modello SIS.

Parametri: %infected: 0.3; fraction infected: 0.3

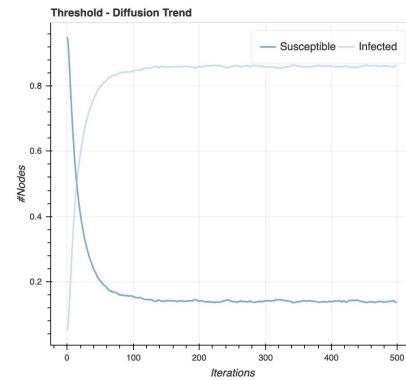


Figure 22: Threshold: Real Network

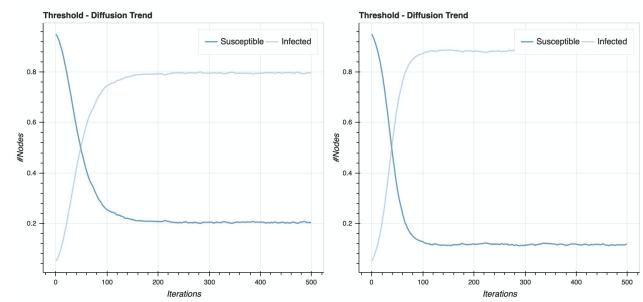


Figure 23: Threshold: ER and BA

6 TASK 3: SUPERVISED LINK PREDICTION

L'obiettivo del *link prediction* è quello di fare predizioni sui link che, all'interno di un grafo, potrebbero apparire in futuro.

In questo task, è stato utilizzato un approccio *supervised* in quanto usare degli algoritmi di apprendimento supervisionato permette di ottenere un'accuratezza molto più alta rispetto agli algoritmi non supervisionati. I primi, infatti, nonostante abbiano dei costi computazionali elevati, tengono conto della complessa struttura della rete.

Proprio per questo motivo, abbiamo creato un sottografo della rete prendendo in considerazione soltanto i link con frequenze maggiori di uno e considerando solo la più grande *connected component*.

In seguito, abbiamo creato una matrice di adiacenza in modo da poter controllare quali nodi fossero connessi tra loro. Questo ci ha permesso di capire quali link potessero essere rimossi, in modo da evitare che il grafo si suddividesse in *multiple connected components*. Abbiamo in questo modo ottenuto un grafo in cui tutti i link omissibili sono stati rimossi e su questo abbiamo applicato *node2vec*³ per estrarre le *features* dei nodi. Il nostro modello è stato addestrato sul training (70%) e validato sul test (30%).

Infine, il *Logistic Regression model* è stato utilizzato per fare predizioni sui link della rete.

Per quanto concerne il valore dell' *average ROC AUC*, esso è pari a 0.96, ciò implica che le performance del nostro modello sono migliori rispetto a quelle di un modello randomico, cioè un modello in cui ciascun arco ha la stessa possibilità di apparire in futuro.

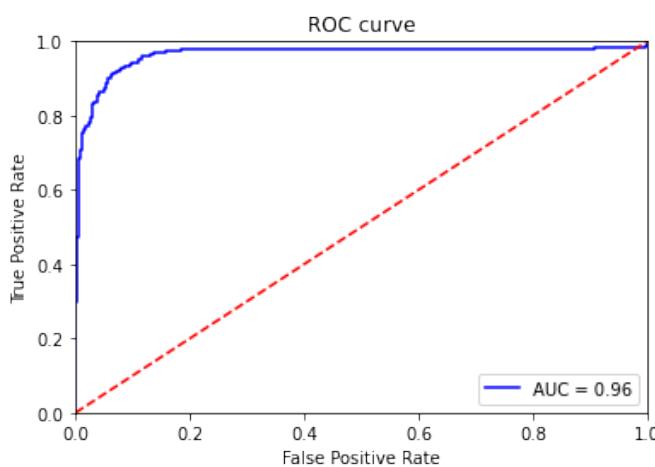


Figure 24: ROC AUC

7 OPEN QUESTION

La spinta all'azione dietro questo studio arriva primariamente dalla nostra personale volontà di visualizzare in maniera reticolare le parole "più rappresentative" dei mesi appena precedenti all'inizio del nostro lavoro e utilizzare il grafo ottenuto come cartina di tornasole delle nostre personali considerazioni e sensazioni del periodo che avevamo appena vissuto. Sapevamo più o meno cosa aspettarci, quali sarebbero stati gli hubs e quali i link più significativi, ma ci chiedevamo se così facendo fosse possibile far risaltare scorciatoie mentali e collegamenti nascosti tra i temi caldi di attualità. Quello del "The New York Times" ci sembrava un buon "punto di vista" da adottare per la nostra rete avendo una visione sufficientemente globale e di comprovata qualità.

Come è noto, però, intenzioni e quesiti cambiano man mano che ci si addentra nel campo della propria ricerca e solo in un secondo momento ci siamo resi conto dell'occasione ghiotta che ci si era posta davanti: l'imminente inizio delle elezioni americane ci permetteva di studiare per tempo l'immagine mediatica creatasi intorno ai due candidati e i contesti in cui i loro nomi venivano nominati maggiormente. Ci siamo chiesti dunque se fosse possibile individuare dei

³Node2Vec project description: <https://pypi.org/project/node2vec/>

cluster distinti centrati sui due candidati, e in caso affermativo, se a partire da ciò fosse possibile tracciare un modello di come il NYT avesse rappresentato i due candidati.

Allora, abbiamo iniziato a interrogarci su quale fosse la strategia migliore per rispondere a questi quesiti. L'esecuzione degli algoritmi di Community Discovery sull'intera rete restituiva un numero troppo grande di cluster per permetterci di analizzarli e studiarli tutti, ed inoltre non assicurava l'effettiva relazione tra il nostro nodo centrale (Trump o Biden) e le parole più decentrate rispetto al cluster, cioè, più specificatamente alle parole a distanza maggiore di 1 dal nodo centrale. Ipotizziamo ad esempio la presenza all'interno del nostro cluster del nodo "Cultura" a distanza 1 dal nodo "Trump": questo indicherebbe necessariamente una stretta correlazione tra i due argomenti e una forte presenza nell'immagine mediatica di Trump del discorso intorno alla sfera culturale, ovviamente declinabile sia in positivo che in negativo. Ma se all'interno del cluster fosse presente a distanza 1 dal nodo "Cultura" il nodo "Arte egizia", che quindi si trova a distanza 2 dal nodo "Trump", non avendo con esso un link diretto, quale sarebbe l'effettiva relazione tra i due nodi? A nostro parere nessuna.

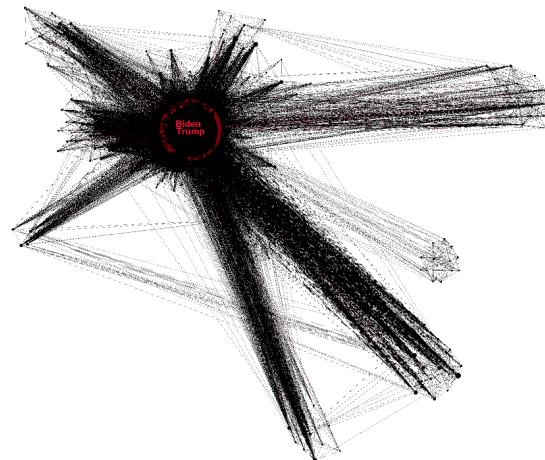


Figure 25: Subgraph of Biden and Trump's neighbors

Per risolvere entrambi questi problemi abbiamo deciso di fare lo studio di Community Discovery esclusivamente sul sottografo dei neighborhood dei nodi "Trump" e "Biden". Così facendo avremmo infatti ridotto considerevolmente il numero di nodi e link, e quindi di cluster ottenuti, e avremmo evitato la presenza di collegamenti fuorvianti, avendo preso in considerazione per la creazione del sottografo esclusivamente i nodi a distanza 1.

Per l'individuazione dei cluster la nostra preferenza è invece inizialmente ricaduta su quegli algoritmi che permettono una partizione netta del grafo, quindi senza overlapping dei cluster, ovvero *Louvain*, *Label propagation* e *Greedy modularity*. In questo modo la nostra analisi sarebbe eventualmente stata più lineare: non avendo parole chiave in comune tra Biden e Trump sarebbe stato più semplice tracciare un profilo netto dei candidati. Abbiamo tuttavia deciso di

utilizzare anche l'algoritmo *Demon*, valutando la possibilità di fare uno studio semantico più approfondito su come parole presenti in entrambi i cluster si collegassero diversamente con gli altri termini presenti.

Abbiamo ottenuto i risultati migliori proprio con questo algoritmo, che ci ha restituito una modularità di 0.35 e 11 diversi cluster. Per individuare quali di questi effettivamente rispondessero alle nostre esigenze abbiamo studiato la centralità di tutti e undici, sperando di trovarne almeno uno che avesse come nodo principale *"Trump"* e un altro con *"Biden"*. Abbiamo dunque scelto le tipologie di centralità che ci sembravano più adeguate al nostro studio: essendo partiti da una rete costruita *ad hoc* partendo dai vicini di Trump e Biden, ci aspettavamo che questi fossero i nodi con degree più alto, dunque *Degree Centrality*; che facessero un po' da spartiacque tra i nodi relativi a Trump e quelli relativi a Biden, quindi *Betweenness Centrality*; che fossero tendenzialmente molto vicini a tutti i nodi presenti, *Harmonic Centrality*.

Con tutte e tre le misure di centralità, in tutti gli undici cluster, i nodi più rappresentativi erano sempre gli stessi 10, senza nessuna differenza sostanziale. E *"Biden"* non saliva mai neanche sul podio dei primi tre. Prima di lui *"Trump, Donald J."*, in tutti i cluster il nodo con il più alto score di centralità; a seguire *"USA Politics and Government"*; *"Elezioni presidenziali 2020"* al terzo posto; *"Covid-19"* inatteso quarto posto e solo quinto, per centralità, *"Biden, Joseph R Jr"*. A seguire *"USA"*, *"Repubblicani"*, *"Black People"* e parole semanticamente correlate a queste.

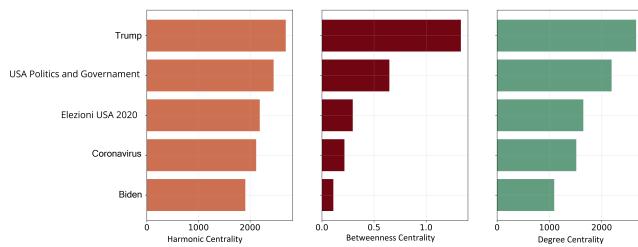


Figure 26: Centralities of clusters

Risalta dunque come il candidato del partito repubblicano abbia dominato la discussione pubblica rispetto al suo concorrente, probabilmente per via del suo ruolo di Presidente degli Stati Uniti d'America, sia nel settore politico (*"repubblicani"* appare tra le 10 parole più centrali, a differenza di *"democratici"*), che in quello culturale (in relazione all'argomento del razzismo). Si evidenzia anche come in linea generale il ruolo del nodo *"Biden"* all'interno della rete sia quasi esclusivamente quello di "vicino" del nodo *"Trump"*: possiamo supporre che, all'interno dei vari articoli, venisse citato solo in relazione al suo diretto contendente e raramente come argomento di discussione collegato ad altri argomenti.

A sostegno di questa tesi riportiamo come fattore interessante la suddivisione fatta dall'algoritmo *Greedy Modulation* che, a differenza di *Demon*, ha individuato solo 2 cluster: il primo contenente il 99% dei nodi del sottografo, il secondo invece contenente appena 13 nodi. Ricercandoli velocemente all'interno del dataset abbiamo notato come tutte queste 13 parole chiave appartenessero al medesimo

articolo e non comparissero in nessun altro: tema dell'articolo una biografia su Joe Biden.⁴

Per fare un confronto abbiamo deciso dunque di prendere tutti i vicini di Biden e osservare quanti di questi non fossero anche vicini di Trump: su 1200 nodi, solo poco più di 100 sono nodi collegati esclusivamente a Biden. La maggior parte di questi fanno riferimento a Stati e personaggi politici, e quindi collegati indirettamente a Trump da forti legami con parole chiave quali *"USA Politics and Government"* e *"USA relazioni interne"*, mentre le altre sono parole chiave relative ad argomenti vari e causali, con connessioni deboli con il nodo *"Biden"*, come, per esempio, *"ciclismo"* e *"orecchini"*. Per completezza, riportiamo che nonostante quanto scritto fino ad ora, Biden è attualmente il nuovo presidente degli Stati Uniti d'America.

8 DISCUSSION

Questo progetto ci ha permesso di approfondire e mettere in pratica le conoscenze di social network analysis apprese durante il corso universitario.

Un aspetto aggiuntivo molto interessante è stato il crawling dei dati: non solo dal punto di vista tecnico della scrittura del codice, ma soprattutto per la sfida mentale di riuscire a "pensare" una rete e individuare i possibili utilizzi pratici di essa.

REFERENCES

- A. Grover, J. Leskovec, Node2vec: Scalable Feature Learning for Networks, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2016.
- G. Rossetti, L. Milli, R. Cazabet. CDlib: a Python Library to Extract, Compare and Evaluate Communities from Complex Networks Applied Network Science Journal. 2019. DOI:10.1007/s41109-019-0165-9.
- G. Rossetti, L. Milli, S. Rinzivillo, A. Sirbu, D. Pedreschi, F. Giannotti, NDlib: a Python Library to Model and Analyze Diffusion Processes Over Complex Networks, Journal of Data Science and Analytics. 2017. DOI:0.1007/s41060-017-0086-6.
- P. Joshi, A Guide to Link Prediction – How to Predict your Future Connections on Facebook⁵

⁴<https://www.nytimes.com/2020/10/27/books/review/new-this-week.html>

⁵<https://www.analyticsvidhya.com/blog/2020/01/link-prediction-how-to-predict-your-future-connections-on-facebook/>.