

Quando gli utenti diventano giudici: il processo Depp-Heard

Alice Isola
a.isola@studenti.unipi.it
Student ID: 545007

Eleonora Rossi
e.rossi41@studenti.unipi.it
Student ID: 562337

Lorenzo Casarosa
l.casarosa4@studenti.unipi.it
Student ID: 582647

ABSTRACT

In estrema sintesi vengono riportate i task svolti durante la realizzazione di questo lavoro. Nello specifico, dopo la creazione della rete reale sono state studiate le comunità in cui gli utenti sono stati raggruppati da diversi algoritmi, per poi passare all'andamento delle opinioni degli utenti e alla loro diffusione, per poi finire con la parte più peculiare e forse importante: l'analisi dei sentimenti più rilevanti che il processo nelle sue diverse fasi ha suscitato nel pubblico che lo ha seguito con interesse. Degni di rilievo sono anche i vari argomenti che sono emersi nelle discussioni avvenute su Twitter.

Occorre precisare che l'analisi è stata condotta senza alcun parteggiamento, ma basandosi esclusivamente sui dati, le comunità create e sui risultati, anche grafici, dell'analisi¹

KEYWORDS

Social Network Analysis, Amber Heard, Johnny Depp, Trial, comunità Sentiment Analysis, Topic Modeling, utenti, tweet

ACM Reference Format:

Alice Isola, Eleonora Rossi, and Lorenzo Casarosa. 2022. Quando gli utenti diventano giudici: il processo Depp-Heard. In *Social Network Analysis '22*. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

L'intera analisi è basata sul processo che vede coinvolti l'attore statunitense Johnny Depp e l'attrice Amber Heard.

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>

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 '22, 2021/22, University of Pisa, Italy

© 2022 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>

Il processo è stato intentato dall'attore contro l'ex moglie per accuse di diffamazione; nello specifico, la donna, durante un suo intervento alla NBC, aveva denunciato di aver subito violenza domestica dall'attore. Tali accuse, a detta dell'attore stesso, avrebbero compromesso la sua carriera. Quindi questo processo racchiude al suo interno molte tematiche rilevanti e attuali, che affondano le proprie radici nella cultura di questo secolo. Tra queste si annoverano: la violenza domestica, discriminazione di genere, violenza psicologica, confine sempre più labile tra libertà di espressione e diffamazione. Per tutte queste motivazioni, tale processo ha avuto una valenza mediatica notevole, ancor più accresciuta dalla notorietà dei soggetti coinvolti.

Occorre precisare che l'obiettivo sotteso a questa analisi non sarà un'interpretazione degli aspetti giudiziari della vicenda o una ricerca delle ragioni che hanno spinto i due protagonisti ai loro interventi, ma piuttosto ci focalizzeremo sulle opinioni espresse liberamente dalla comunità sui social networks.

2 DATA COLLECTION

Selected Data Sources

In prima istanza si è resa necessaria la fase di **Data Collection** durante la quale sono state raccolte le risorse occorrenti per procedere poi alla creazione della rete. Si è deciso di utilizzare come fonte di dati il social network *Twitter*, in quanto risulta essere molto più dinamico e immediato nell'esprimere opinioni su argomenti di tendenza. Ciò favorisce l'interazione tra utenti, la discussione e la diffusione di opinioni anche tra utenti molto polarizzati. Il dinamismo e la velocità con cui le informazioni vengono diffuse sono favorite anche dalla quasi totalità dei profili pubblici, aspetto che incarna perfettamente la logica dietro questo Social Network (lo stesso non si può dire, invece, per social più comuni come *Instagram* e *Facebook*). Questa decisione è stata ulteriormente avvalorata da una maggiore facilità nell'effettuare la fase di *scraping* per cui è stato adottato come strumento la libreria *Twint*².

Crawling Methodology and Assumptions. L'arco temporale durante il quale sono stati raccolti i dati ha come data di

²Disponibile all'interno del seguente Repository: <https://github.com/twintproject>

inizio il 12 aprile 2022 che coincide con l'avvio del processo e si conclude il 13 giugno 2022. Dato l'arco temporale molto ristretto si è deciso di raccogliere tutti i tweet contenenti quattro hashtags differenti di natura più neutrale possibile: *DeppHeard*, *DeppHeardTrial*, *DeppVsHeard*, *JohnnyDeppAmberHeardTrial*. Al fine di raggiungere un sufficiente numero di dati non è stato applicato alcun vincolo sulla ricerca, né sul numero di tweets ammessi.

I dati sono stati raccolti in quattro file csv ciascuno relativo ad uno degli hashtag appena menzionato. Alla fine del processo sono stati scaricati in totale 271352 tweets in lingua inglese. Abbiamo così ottenuto un dataset composto da 36 *features*, molte delle quali sono state eliminate poiché contenevano valori nulli o non erano funzionali al fine della nostra analisi (Tabella 1). In seguito a questa fase di *preprocessing*, il dataset risulta ora composta da 15 colonne. Tra le variabili mantenute le seguenti sono state utilizzate per la creazione della rete: *hashtags*, *user_id*, *mentions*, *reply_to*, mentre le altre verranno utilizzate per successive analisi. Segue una lista del numero di tweets ottenuti per ciascun hashtag:

- (1) **#DeppVsHeard**: 124027
- (2) **#DeppHeardTrial**: 69817
- (3) **#JohnnyDeppAmberHeardTrial**: 58630
- (4) **#DeppHeard**: 24136

Prima di creare la rete è stato ritenuto opportuno studiare l'andamento dei tweet in corrispondenza dei giorni in cui il processo si è svolto (Figura 1).



Figure 1: Andamento temporale dei tweets

L'andamento nel grafico mostra quattro fasi principali del processo mediatico. All'inizio assistiamo ad un numero basso di tweets pressoché stabile, forse a causa dello scarso interesse che il processo destava al suo inizio negli utenti di Twitter. Entrati nella fase principale del processo, è possibile assistere ad una crescita molto ripida della curva, a testimonianza di una crescita esponenziale nel numero di tweets (picco relativo al periodo 15/05 - 18/05). La terza fase corrisponde a marcata discesa della curva causata dallo stop del processo per il 22 maggio. L'ultima fase, che è in prossimità

della fine del processo, conferma un numero di tweets alto e piuttosto stabile. Questi aspetti sono perfettamente in linea con le aspettative che avevamo circa il comportamento degli individui sui social: l'interesse mediatico è più alto, quanto è più cruciale è la fase dell'evento.

Table 1: Valori eliminati dal corpus durante la fase di pulizia

Feat. valori nulli	Feat. inutili	Feat. mantenute
near	place	id
geo_id	thumbnail	conversation_id
source	language	date
user_rt_id	video	time
retweet_id	photos	user_id
user_rt	created_at	username
retweet_date	timezone	name
translate	cashtags	tweet
trans_src	urls	mentions
	retweet	replies_count
	quote_url	retweets_count
		likes_count
		hashtags
		link
		reply_to

Preso come punto di partenza il dataset appena descritto, ne è stato creato un altro che sarà funzionale alla creazione della rete. Tale dataset descrive le relazioni tra utenti; relazioni che consistono in rispondere ad un tweet o l'esser menzionato da un altro utente. In particolare il dataset è costituito da 21713 righe e 5 colonne dove:

- *source* e *id_source* corrispondono a *username* e *user_id* rispettivamente, dell'autore dei tweet
- *target* e *id_target* si riferiscono a *username* e *user_id* rispettivamente, dell'utente che ha risposto o che è stato menzionato

Successivamente, è stata aggiunta una *feature* contenente il peso di ciascuna interazione; in particolare, tale peso è determinato studiando il numero di volte che egli/ella ha assistito ad una risposta oppure ad una menzione al proprio tweet da parte di un utente. Tale peso è poi stato normalizzato in un range da 0 a 1. Questa versione del dataset sarà quella adottata per condurre tutte le successive analisi e tasks.

3 NETWORK CHARACTERIZATION

A partire dal dataset così configurato è stata creata la rete tramite la funzione *from_pandas_edgelist* della libreria *networkx*³. Ne risulta un multigrafo, diretto, avente 18258 nodi e

³<https://networkx.org/documentation/stable/index.html>

21680 edges e 40 *self-loops* (utenti che menzionano o rispondono al loro stesso tweet).

Tale struttura è giustificata dal fatto che stesse coppie di utenti possono avere più interazioni fra di loro; inoltre le interazioni sono di tipo diretto in quanto un utente potrebbe essere in relazione ad un altro ma quest'ultimo potrebbe non avrebbe avere alcun tipo di collegamenti con il primo.

Per confrontare al meglio le caratteristiche della rete reale, si è scelto di confrontarla con quattro **modelli sintetici**, i quali sono stati creati con caratteristiche (numero di nodi e di *edges*) più simili possibili a quelle della rete reale, impostando i seguenti parametri.

- **Modello Erdos-Renyi (ER)**: con una probabilità $p = 0.00011$;
- **Modello Barabasi-Albert (BA)**: con $m = 1$;
- **Modello Watts-Strogatz (WS)**: con $k = 2$ e $p = 0$;
- **Configuration Model (CM)**: impostando per ciascun nodo della rete il grado del nodo corrispondente nella rete reale.

Nella tabella 2 sono riportate le informazioni di base della rete reale (RN) e dei quattro modelli sintetici.

Table 2: Informazione di base della rete reale e dei quattro modelli sintetici

Modello	Nodi	Edges	Avg. Degree	L_max
RN	18246	21686	2.37	166449135
ER	18246	24924	2.73	166449135
BA	18246	18245	2	166449135
WS	18246	18246	2	166449135
CM	18246	21674	2.37	166449135

Caratteristiche RN

Una delle metriche più importanti di cui ci avvaleremo nel corso della successiva analisi è la distribuzione dei gradi che i nodi della rete reale generano. Dal seguente grafico (Figura 2), si evince un fenomeno abbastanza prevedibile: al diminuire del numero dei nodi, la *degree* tende ad aumentare in maniera monotonica fino ad attestarsi intorno al valore di 10^2 . Ricordando la definizione di degree centrality e il fatto che ciascun nodo è collegato ad un altro secondo un criterio specifico descritto nella sezione 3, non è sorprendente che una forte diminuzione della quantità di nodi presente nella rete generi un valore di *degree* abbastanza alto. Il numero di gradi di ciascun nodo influenza inoltre la presenza di componenti tra loro fortemente/debolmente connesse; In particolare ci si aspetta che una degree centrality alta sia associata ad una rete con componenti fortemente connesse tra di loro. Questo aspetto è maggiormente rafforzato nel caso in cui il numero di nodi sia molto basso.

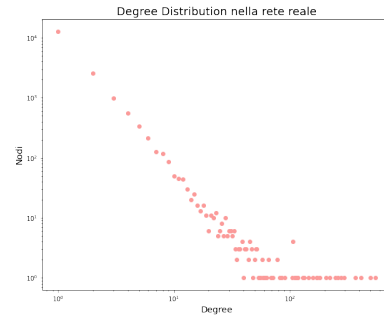


Figure 2: Distribuzione dei gradi nella rete reale

Per completezza prenderemo in analisi altre metriche, tra cui il coefficiente di Clustering locale e globale, la densità della rete, le misure di centralità e l'analisi dei percorsi o *paths*.

Table 3: Densità, average clustering coefficient, average shortest path e numero di hubs della rete reale e dei modelli sintetici

	Densità	Avg. clust. coef.	Avg. short. path	Hubs
RN	1.3×10^{-4}	0.009	5.36	40
ER	1.5×10^{-4}	7.03	9.57	34
BA	1.1×10^{-4}	0.0	9.39	11
WS	1.1×10^{-4}	0.0	4561.75	0
CM	1.2×10^{-4}	0.007	4.83	40

Confronto con ER

Confrontando la distribuzione dei gradi che i nodi presentano all'interno della rete reale (Figura 2) e del modello Erdos-Renyi (Figura 3), si può osservare come il valore dei gradi cresca in maniera molto simile a mano a mano che il numero di nodi diminuisce.

Tuttavia, c'è una differenza nella gradualità con cui ciò avviene: la rete random infatti non avendo alcun criterio specifico nell'unire due nodi decresce più velocemente e più ripidamente rispetto a quella della rete reale. A quest'ultimo aspetto, si lega anche il maggior numero di componenti fortemente connesse tra loro all'interno della rete randomica: infatti, due nodi verranno collegati tra loro in maniera completamente casuale senza il bisogno di soddisfare i criteri menzionati nella sezione 3.

Confronto con BA

La peculiarità del modello Barabasi Albert consiste nella presenza di un'unica componente che connette tutti i nodi della rete. La presenza di un'unica *Giant Component* tende a far

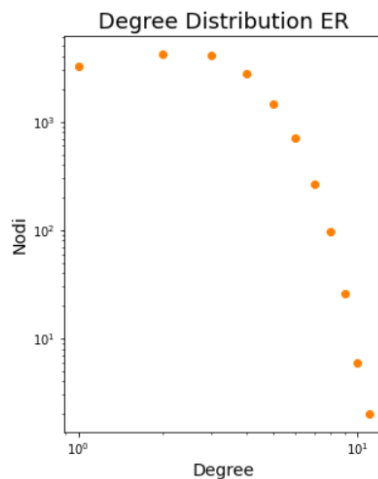


Figure 3: Distribuzione dei gradi nella rete random

diminuire il valore complessivo della Closeness Centrality. Infatti, ciascun nodo dovrà essere collegato a un nodo molto distante perché in un'unica componente; nella rete reale invece, essendoci numerose componenti (numero di componenti fortemente connesse: 2408) aumenta la probabilità di avere minori distanze tra i nodi. Ciò spiega il più alto valore di questa metrica nella rete reale rispetto al modello Barabasi. Segue la Closness Centrality dei primi 4 nodi con centralità maggiore di ciascun modello:

- Rete reale:
 - ('deppheardtruth', 0.2096847344352438),
 - ('mensurviveto', 0.19311074013869345),
 - ('thirdgirl', 0.1929779364858902),
 - ('theemilydbaker', 0.18855523599767265),
- Rete BA:
 - (0, 0.19822256988581424),
 - (2, 0.1974545730024567),
 - (7, 0.1763705085696057),
 - (18, 0.17214375347920027),

Analoghe considerazioni per la Betweenness Centrality, in quanto con componenti più piccole è più probabile che ciascun nodo sia sul cammino più breve per collegarne altri.

- Rete reale:
 - ('deppheardtruth', 0.2096847344352438),
 - ('mensurviveto', 0.19311074013869345),
 - ('thirdgirl', 0.1929779364858902),
 - ('theemilydbaker', 0.18855523599767265),
- Rete BA:
 - (0, 0.19822256988581424),
 - (2, 0.1974545730024567),

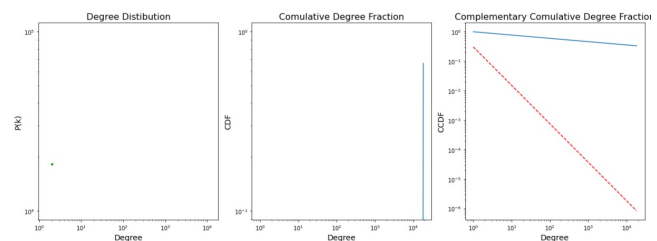


Figure 4: Distribuzione dei gradi e cumulative degree function della rete WS

(7, 0.1763705085696057),
(18, 0.17214375347920027),

Confronto con WS

In relazione alla rete generata con il modello, ci è sembrato interessante analizzare le *Cumulative Degree Functions* (Figura 4) che, tramite lo studio dell'Area Sotto la curva definiscono la probabilità che un punto casuale non appartenente ad essa abbia valore più piccolo del punto corrispondente sulla curva.

A tal riguardo, notiamo una peculiarità nella rete WS, la quale non è in grado di stabilire con affidabilità tale questione (retta parallela all'asse y e quindi area sotto la curva pari a 0). Altra particolarità di questo modello, si evidenzia nella degree distribution nella quale non si osserva un andamento crescente o decrescente della curva, bensì si ha solamente un punto nel piano, ad evidenza del fatto che tutti i punti della rete potrebbero avere la stessa degree centrality (2).

Confronto con CM

In termini di numero di *edges*, distribuzione di gradi (Figura 5) e delle relative metriche, queste due reti non presentano differenze sostanziali, ma presentano molte affinità. Questo perché nella costruzione del Configuration Model, si prendono in considerazione i gradi dei nodi delle rete reale, associando a ciascun nodo della rete sintetica il grado del nodo nella rete reale.

Confrontando infatti la degree centrality dei nodi di entrambe le reti, essa risulta uguale sia nella rete reale che nel Configuration Model.

4 TASK 1: COMMUNITY DISCOVERY

Come primo task si è deciso di analizzare la distribuzione dei nodi all'interno di *comunità*, utilizzando i seguenti **Algoritmi di Community Discovery**: *Louvain*, *K-clique* e *Label Propagation*. La scelta di algoritmi appartenenti a famiglie differenti permette di ottenere risultati confrontabili e che provano a definire in maniera completa le caratteristiche della rete. In particolare, il primo appartiene alla *Internal density family*, il K-clique alla *Structure definition family* ed

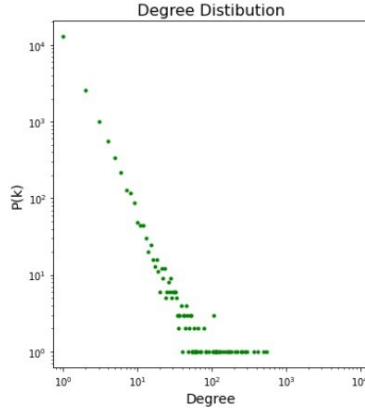


Figure 5: Distribuzione dei gradi nella rete CM

infine l'ultimo appartiene alla *Percolation*. Per tutte le analisi condotte in questo task verrà utilizzata la libreria **CDlib**⁴ in Python.

Negli algoritmi in cui era necessario scegliere dei parametri è stata effettuata un'operazione di *Parameter Turing* utilizzando il metodo *random search*.

Inoltre, è stata effettuata per ciascun algoritmo una **valutazione interna** che, mediante dei punteggi valuta la bontà delle comunità all'interno delle singole partizioni in termini di *Internal Edge Density*, *Modularity*, *Average Node Degree* (valori che devono essere massimizzati nel confronto delle partizioni), *Conductance* (valore che deve essere minimizzato nel confronto delle partizioni).

Occorre precisare che poiché molti algoritmi possono operare solo in presenza di grafi non diretti e non pesati, si è reso necessario trasformare la rete adoperata nelle precedenti analisi, in una *undirected, unweighted graph*. Ciò ha permesso di ottenere dei valori attendibili e confrontabili.

Dalla prospettiva della valutazione interna, è possibile osservare come gli algoritmi *Louvain* e *Label Propagation* abbiano performance molto simili tra loro. Nello specifico i punteggi che dovrebbero essere massimizzati, in questi due algoritmi risultano essere i più alti e di valore simile tra loro; a dimostrazione dell'efficienza dei due per la rete adottata.

Considerazioni molto differenti possono essere fatte nei confronti del terzo algoritmo implementato, ossia il *K-Clique*: esso presenta infatti i peggiori valori registrati in termini di *Internal Edge Density*, *Average Node Degree*, *Modularity* e *Conductance*. Di seguito viene riportata la tabella riepilogativa delle metriche riscontrate.

Table 4: Valutazione interna dei tre algoritmi di CD utilizzati

	Modularità	IED	AND	Conductance
Louvain	0.79	0.85	1.15	0.03
Label Prop.	0.65	0.79	1.21	0.20
K-Clique	0.01	0.90	2.38	0.76

Analoghi risultati si riscontrano nell'ambito della valutazione esterna con cui è possibile osservare la similarità nei comportamenti degli algoritmi individuati in termini di **Normalized Mutual Information (NMI)**, metrica usata per valutare la similarità tra le partizioni individuate come output dagli algoritmi testati, e **F1 score**, calcolata come la media armonica tra Precision e Recall moltiplicata per due. Entrambe sono considerate buone se hanno valori alti.

Per quanto riguarda la NMI, considerando il fatto che tale metrica per generare i risultati ha bisogno di confrontare partizioni aventi la stessa copertura in termini di nodi, la nostra analisi si è limitata ad un confronto tra Louvain e Label Propagation (Figura 6), da cui è possibile riscontrare un valore di NMI molto alto, pari a 0.81 a indicare un'alta similarità tra le comunità individuate dai due algoritmi.

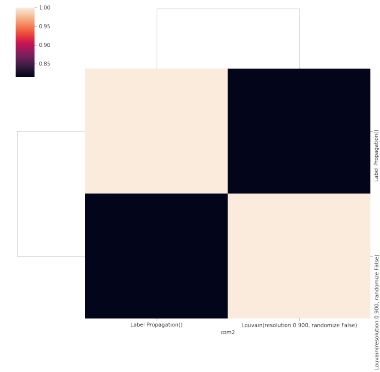


Figure 6: Matrice di similarità tra Louvain e Label Propagation rispetto alla NMI

Successivamente, per favorire un maggior numero di comparazioni e un confronto tra tutti e tre gli algoritmi implementati, a tale metrica è stata affiancata la F1 score (Figura 7).

⁴<https://cdlib.readthedocs.io/en/latest/overview.html>

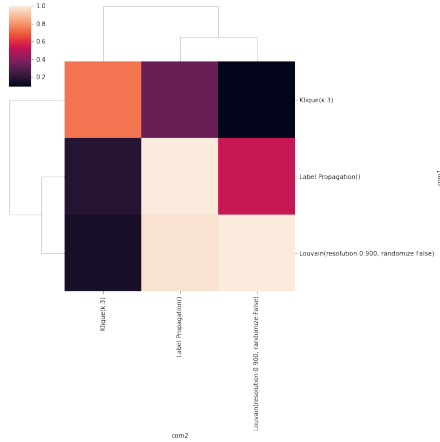


Figure 7: Matrice di similarità tra Louvain, Label Propagation e K-Clique rispetto al F1 score

Il valore alto di tale metrica corrisponde ad un'alta similarità tra le comunità che i due algoritmi generano. Nello specifico la combinazione tra gli algoritmi di Louvain e Label Propagation mostra il più alto valore di F1 score (0.53), come già dimostrato con il valore di NMI, mentre il primo insieme al K-Clique dà luogo al più basso valore (0.089). Si ottiene un valore intermedio considerando il K-Clique insieme alla label propagation (0.33).

Tali risultati non sono affatto sorprendenti se si considera la logica dietro al funzionamento dei tre algoritmi di Community Discovery; infatti Louvain e la Label Propagation formano comunità basandosi su criteri ben predefiniti, l'uno massimizzando il valore della modularità (Tabella 4), l'altro assegnando etichette a ciascun nodo sulla base dei nodi a lui più vicini in termini di caratteristiche (considerato un valore α della probabilità). La generazione delle comunità nel K-clique avviene invece unendo gruppi di nodi che ne condividono uno o più; ne consegue che si formeranno delle comunità contenenti nodi molto differenti tra di loro.

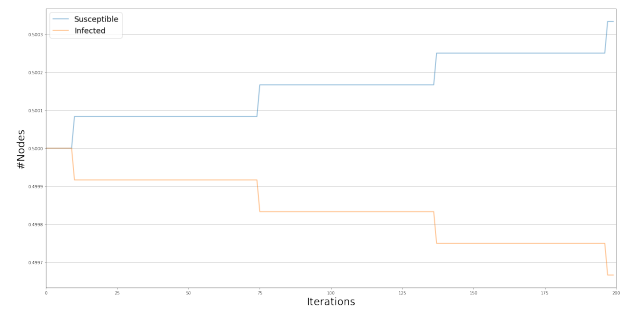
5 TASK 2: DIFFUSIONE - OPINION DYNAMICS

Il successivo step consiste nella diffusione delle opinioni all'interno della rete considerando la sua giant component e poi per successive comparazioni è stato considerato un grafo completo avente come numero di nodi lo stesso della Giant Component (11974). Tali confronti sono stati effettuati adottando 4 differenti modelli discreti di diffusione delle opinioni: **Voter**, **Majority Rule**, **Sznajd** e **Q-Voter**.

Modello Voter

Il modello **Voter** sceglie casualmente ad ogni iterazione un agente i , la cui opinione verrà influenzata dal suo vicino j . Ciò spiega l'andamento delle due curve relative al *complete graph* (Figura 8b) tra loro inversamente proporzionale:

all'aumentare del numero di nodi infetti diminuisce quello dei nodi suscettibili, a dimostrazione del cambio di opinione che la vicinanza di un vicino j provoca nell'agente i (che diventa infetto, cioè acquisisce una sua opinione). Inoltre, possiamo desumere che, all'aumentare dell'iterazione, il numero degli utenti suscettibili decresce fino a raggiungere una forte divergenza tra le due curve: ciò significa che una volta raggiunto un numero di iterazioni molto alto non è più possibile uno scambio di status che trasformi l'agente i scelto di volta in volta casualmente in un individuo suscettibile. Nella rete reale (Figura 8a) si osserva il caso estremo rispetto a quello appena descritto, ovvero la divergenza delle curve è totale; ciò significa che non esiste un'iterazione tale per cui un individuo suscettibile acquisisca un'opinione e diventi quindi agente. Dal punto di vista del numero di nodi è evidente come l'aumentare del numero di iterazioni provochi un andamento decrescente nella curva dei nodi infetti.



(a)



(b)

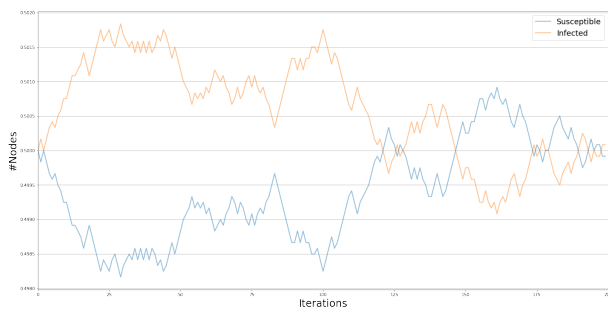
Figure 8: Diffusione delle opinioni nel modello *Voter* su rete reale (8a) e su rete completa (8b)

Modello Majority Rule

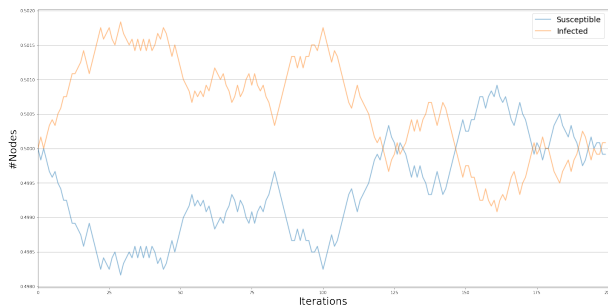
Il modello **Majority Rule** condivide con **Voter** l'idea di estrarre non uno ma un gruppo di agenti i dai nodi della rete. In questo caso però il cambio di opinione degli agenti nel gruppo è dettato dall'opinione che domina all'interno del gruppo stesso (come succede in un dibattito con un oratore

e una platea di uditori). Il forte carattere casuale con cui l'algoritmo estrae i gruppi di agenti, sembra in qualche modo motivare l'andamento delle curve riportato nelle seguenti curve (Figura 9a): infatti un agente i può ad un'iterazione j diventare infetto con una certa opinione, mentre ad una iterazione z lo stesso agente potrebbe diventare suscettibile e successivamente infetto con l'opinione opposta. Infatti questo algoritmo è quello che presenta varie intersezioni tra le due curve ad ulteriore dimostrazione della forte suscettibilità a cui sono soggetti gli utenti.

Non si riscontra nessuna differenza tra il modello completo (Figura 9b) e quello generato con la rete reale a dimostrazione del fatto che questo algoritmo non è influenzato da forti connessioni tra i nodi appartenenti alla rete.



(a)



(b)

Figure 9: Diffusione delle opinioni nel modello *Majority Rule* su rete reale (9a) e su rete completa (9b)

Modello Sznajd

Il modello *Sznajd* è sicuramente quello che conduce ai risultati più inaspettati: le due curve parallele infatti mostrano che non c'è mai uno scambio tra lo status di nodo infetto e quello di nodo suscettibile. Tutti i nodi suscettibili, già alla prima iterazione sono tutti diventati nodi infetti. Questo è spiegato dal funzionamento che c'è dietro a tale algoritmo: dopo aver selezionato in maniera casuale un agente i e un suo vicino j , entrambi aventi una propria opinione, l'algoritmo

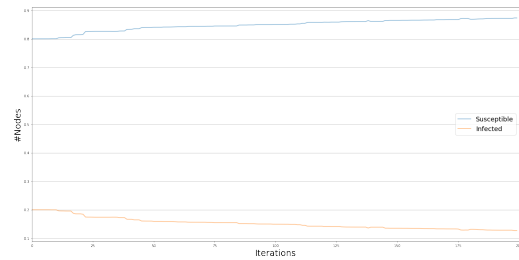
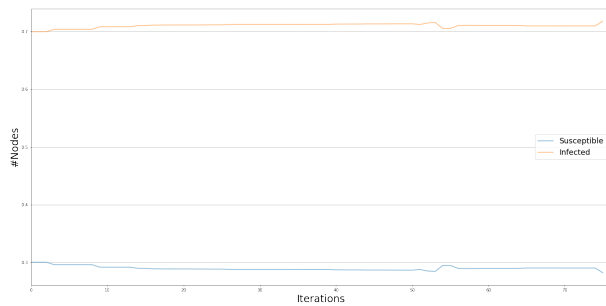


Figure 10: Diffusione delle opinioni nel modello *Sznajd* su rete reale (11a) con soglia di infetti pari a 0.20

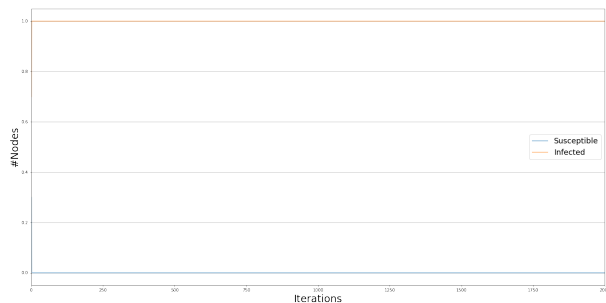
verifica la coincidenza o meno delle due. Se coincidono tutto il vicinato prende tale opinione condivisa dai due (dunque o restano infetti mantenendo la stessa opinione condivisa da i e j oppure diventano infetti ma con l'opinione opposta rispetto a quella originaria).

Giungiamo a risultati più interessanti se si prende in considerazione la rete completa, ma con una soglia di infetti molto più bassa (pari allo 0.20) di cui è riportato il grafico di seguito (Figura 10) in cui osserviamo una divergenza tra il numero di nodi che si infetta e il numero di utenti suscettibili.

La peculiarità di questa situazione si ha nell'osservare un aumento del numero di nodi suscettibili che sembra andare contro quello che è la logica dell'algoritmo che stiamo utilizzando. La spiegazione di questo fenomeno risiede però nel fatto che un nodo suscettibile non è in grado di influenzare l'opinione altrui; se a questo aggiungiamo che la soglia di nodi suscettibili è molto più alta di quella dei nodi della rete infetti, la probabilità di selezionare casualmente due nodi suscettibili oppure due nodi infetti va a gravare negativamente sul numero finale di nodi infetti molto più di quanto la probabilità di estrarre un nodo infetto e un nodo suscettibile come suo vicino riesca a contribuire positivamente su tale numero.



(a)



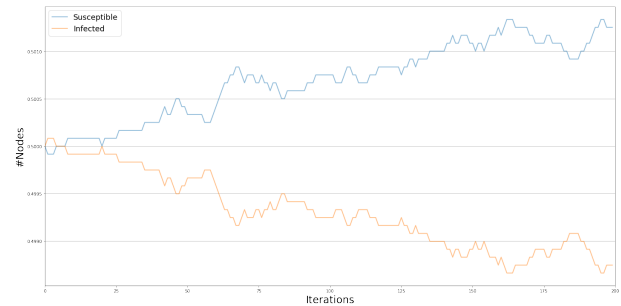
(b)

Figure 11: Diffusione delle opinioni nel modello *Sznajd* su rete reale (11a) e su rete completa (11b)

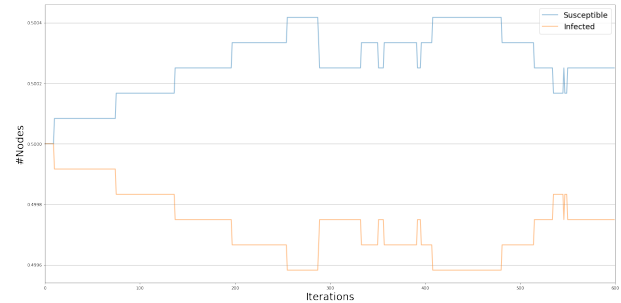
Ricordiamo, infatti, che se estraiamo due nodi infetti, uno dei due verrà computato come suscettibile perché costretto dall'algoritmo a cambiare la sua opinione. Anche in questo caso non emergono forti differenze tra le configurazioni ottenute con la rete completa e la rete reale.

Modello Q-Voter

Come generalizzazione di Voter e Sznajd, il modello **Q-Voter** tende a combinare le caratteristiche peculiari di questi due algoritmi. Nello specifico un ruolo chiave è giocato dagli agenti scelti casualmente, ma stavolta viene introdotta una probabilità che essi diventino suscettibili e quindi cambino opinione rispetto a quella che già hanno (ricordiamo che nel Voter la probabilità è del 100%). In aggiunta a ciò, occorre considerare se i q vicini estratti siano o meno d'accordo; in caso affermativo il fenomeno di propagazione delle opinioni è il medesimo visto nello Sznajd. In caso contrario subentra la probabilità ϵ del cambio di opinione. Nel nostro caso specifico, l'andamento simmetrico delle due curve della rete completa denota una probabilità pari al 50 %: ciò significa che ogni due iterazioni il soggetto i torna ad essere suscettibile e si infetta dell'opinione che aveva due iterazioni prima.



(a)



(b)

Figure 12: Diffusione delle opinioni nel modello *Q-Voter* su rete reale (12a) e su rete completa (12b)

L'unica differenza riscontrata nella rete reale rispetto a quella completa si ha nell'accentuazione che la prima manifesta nel cambiamento di status dei nodi. Nella rete completa notiamo infatti tratti di curva costanti.

6 TASK 3: OPEN QUESTION

L'ultima fase della nostra analisi si è incentrata sull'utilizzo di tecniche di *Natural Language Processing* (NLP) combinate con metodi di analisi delle reti complesse al fine di studiare la percezione che gli utenti hanno avuto nel corso del processo di Johnny Depp e Amber Heard.

La seguente analisi consta dell'utilizzo delle tecniche di Opinion mining, **Sentiment Analysis** e **Topic modeling**, le quali estraggono a partire da risorse testuali i sentimenti degli utenti e gli argomenti più frequenti, rispettivamente. Le suddette tecniche verranno applicate sulle comunità identificate tramite l'utilizzo di due algoritmi di *Community Discovery*: Louvain e K-Clique. È stato ritenuto consono confrontare questi due algoritmi in virtù del fatto che, come visto nella sezione 4, essi generavano risultati completamente diversi in termini di numero di comunità e di valore di modularità. La Label Propagation non è stata considerata in questa sede in quanto molto simile a Louvain. In particolare, la nostra analisi si è concentrata sulle prime quattro comunità più popolate, ma verranno riportati i risultati di quelle comunità

che, comparate tra loro, danno luogo a fenomeni più interessanti o più peculiari. Prima di poter utilizzare le suddette metodologie è stato necessario eseguire alcune operazioni di preprocessing dei tweet e la creazione dei dataset con le comunità. Per approfondimento, si rimanda alle sezioni successive.

Preparazione dei dati: pulizia dei tweet e creazione del dataframe con le comunità

In prima istanza, è stato necessario effettuare alcune operazioni di *preprocessing* sui tweet contenuti all'interno del dataset individuato alla fine della fase di scraping.

- ripulire i tweet da elementi superflui (come simboli, lettere maiuscole, doppi spazi, link ecc.) tramite l'utilizzo delle espressioni regolari (re);
- lemmatizzare le parole, ossia riportarle al lemma originale;
- eliminare le *stopwords*, ossia le parole più frequenti all'interno di una lingua, ma che non aggiungono significato o informazione alle frasi.

In seconda istanza, è stato creato un dataset le cui colonne corrispondono alle quattro comunità prese in esame. Ciascuna comunità ha al suo interno gli utenti che la compongono identificati dal loro username. Questo dataset è stato poi utilizzato per filtrare il dataset originale selezionando solo i tweet scritti dagli utenti appartenenti a quelle comunità.

Sentiment Analysis

I tweet di ciascuna comunità sono stati poi passati alla funzione *sentiment.polarity_scores* del modello **VADER** (*Valence Aware Dictionary for Sentiment Reasoning*). Tale metodo assegna a ciascun tweet un punteggio, in un range compreso tra -1 e 1 in base alla natura del sentimento che l'utente ha trasmesso tramite il suo tweet. Questo punteggio è stato poi categorizzato con l'ausilio di un valore soglia scelto arbitrariamente; in particolare, il valore discriminante è zero (al valore > 0 corrisponde un'emozione positiva manifestata dall'utente). Mediante l'ausilio delle seguenti rappresentazioni grafiche è stato possibile stabilire il peso che ciascun tipo di emozione ha in relazione a tutti gli utenti della comunità analizzata. In particolare è stata scelta da prima la comunità 0 generata dall'algoritmo di Louvain (Figura 13) che rappresenta una situazione piuttosto equilibrata, nella quale le emozioni coesistono in egual misura.

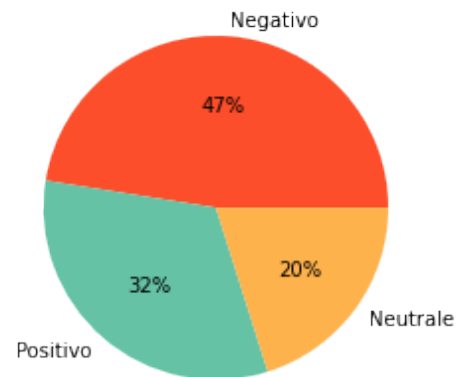


Figure 14: Percentuali di tweet neutrali, positivi e negativi nella prima comunità (community_1) identificata con l'algoritmo K-clique

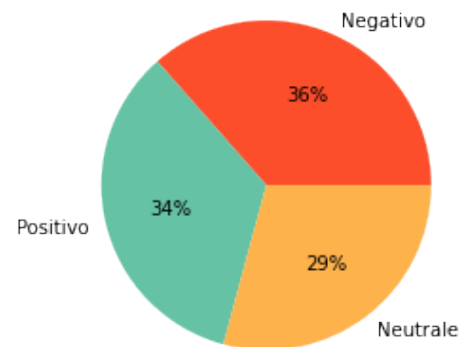


Figure 13: Percentuali di tweet neutrali, positivi e negativi nella prima comunità (community_0) identificata con l'algoritmo Louvain

Il diverso funzionamento dei due algoritmi visti in Community Discovery Analysis riaffiora nella suddivisione delle emozioni all'interno della comunità 1 generata dal k-Clique. Il grafico in figura 14, infatti, mostra un forte sbilanciamento delle emozioni di natura negativa che predominano sugli altri due tipi.

Per approfondire ulteriormente la questione ed avere una prospettiva ancor più completa, abbiamo affiancato a tale grafico la rappresentazione *Word Cloud* (Figura 15a) dalla quale è possibile evincere le parole più frequentemente usate e la loro accezione negativa. Interessante è stato notare come la valenza fortemente negativa di alcune parole presenti nella comunità di Louvain sia molto attenuata se si considerano le medesime parole nella comunità generata da k-Clique (e



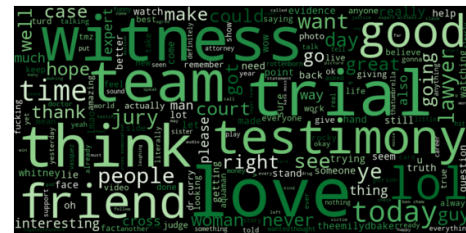
(a)



(a)



(b)



(b)

Figure 15: Word Cloud delle parole più frequenti nei tweet classificati come *Negativi*

viceversa). Alcuni esempi sono costituiti dalla parola *witness* o *testimony*. Se contestualizziamo queste osservazioni e le mettiamo in relazione al processo che è stato preso in esame, è possibile associare questo cambiamento delle due parole alle diverse personalità che sono state chiamate in tribunale a testimoniare.

A corroborare la nostra ipotesi prendiamo in considerazione la figura 16 ed in particolare la parola *testimony*, che passa da un’accezione negativa, come prima descritto, ad una completamente positiva. Un’altra caratteristica da tenere in considerazione è la grandezza delle parole rappresentate: essa è direttamente proporzionale alla frequenza con cui la parola viene utilizzata nei vari tweet.

Topic Modeling

L'ultimo task affrontato consiste nell'applicazione di una tecnica di elaborazione automatica del testo impiegata per l'estrazione degli argomenti presenti all'interno di risorse testuali, raggruppandoli in clusters o gruppi di argomenti; tale tecnica prende il nome di **Topic Modeling**. Tale analisi è stata condotta mediante l'utilizzo di una particolare tecnica di *Topic Embedding* derivata dal BERT (Bidirectional Encoder Representation), ossia il **BERTopic**. Esso è stato applicato sulle quattro comunità più popolose individuate dall'algoritmo Louvain, già studiate nella sezione relativa alla Sentiment Analysis (sezione 6). È stato usato come riferimento tale algoritmo in quanto massimizzava il valore della modularità (indice di bontà nella suddivisione in comunità),

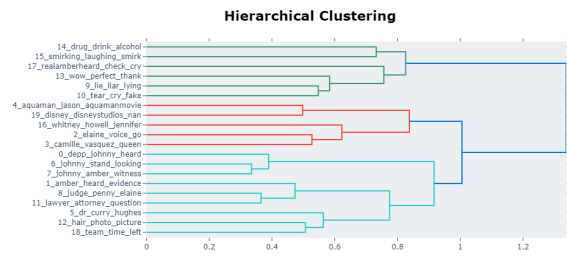
Figure 16: Word Cloud delle parole più frequenti nei tweet classificati come *positivi*

e al contempo garantiva un sufficiente numero di tweet in ciascuna delle quattro comunità; quest’ultimo aspetto non si verifica, invece, se consideriamo il K-Clique dove una comunità presenta solo 3 utenti.

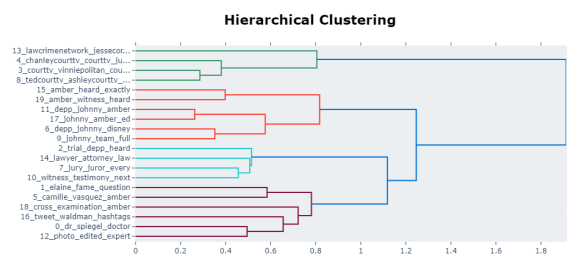
Prima di entrare nel merito dell'analisi vera e propria è stato necessario rimuovere da ciascuna comunità i tweet duplicati e isolare solamente i tweet che soddisfano un valore minimo di interazione (1 like e 1 retweet). Da ciascuna comunità, usando il modello BERTopic sono stati estratti i venti argomenti più rilevanti all'interno di ciascuna comunità; l'interazione tra gli argomenti sarà il criterio discriminante per valutare la bontà dei clusters individuati. Le tecniche di visualizzazione di tali clusters sono: il dendrogramma, la matrice di confusione, la rappresentazione grafica sul piano cartesiano della distanza che separa i topic e la distribuzione temporale degli argomenti.

Le raffigurazioni mediante dendrogramma del raggruppamento dei tweets nelle varie comunità mostrano tutte suddivisioni abbastanza marcate; ciò denota una buona somiglianza degli argomenti discussi dagli utenti all'interno delle singole comunità. Le seguenti figure (17) mostrano che tagliando il dendrogramma ad una certa altezza è possibile ottenere clusters ben separati e distinti tra loro. A titolo esemplificativo, riportiamo di seguito le due comunità che hanno ottenuto risultati migliori in termini di cluster, ossia la comunità 2 (figura 17a) e 3 (figura 17b)

È stato ritenuto interessante affiancare a questa rappresentazione l'andamento che i tweet hanno avuto nell'arco



(a)

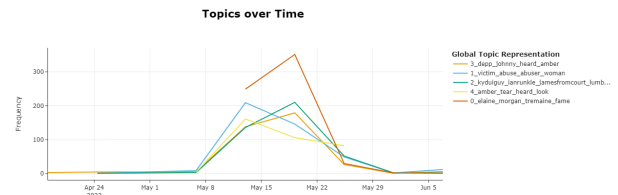


(b)

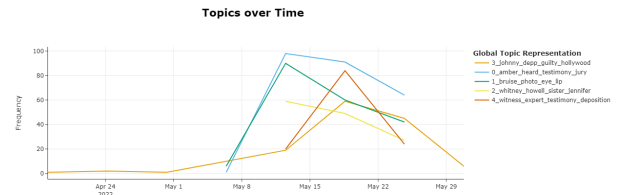
Figure 17: Rappresentazione grafica del raggruppamento dei tweet nelle comunità individuate da Louvain: comunità 2 17a, comunità 3 17b

temporale considerato (inizio processo fino al 12 giugno). In entrambi i casi, prevalgono argomenti riferiti ad Amber Heard e ad aspetti a lei strettamente correlati (come ad esempio le sue lacrime, il suo avvocato Elaine, i testimoni da lei chiamati ecc.). Inoltre, è evidente come la frequenza dei tweet riferiti ad uno specifico tema sia molto influenzata dalla fase in cui si trova il processo; a titolo esemplificativo, un testimone è al centro dell'attenzione mediatica nel giorno della sua deposizione, ma col passare del tempo perde sempre più rilevanza.

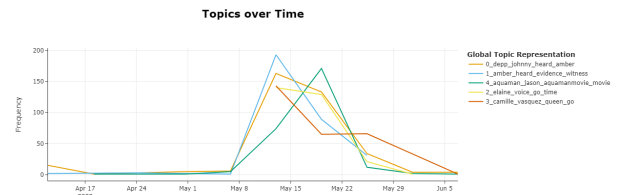
Infine, per cogliere la correlazione tra i vari topic all'interno delle comunità è stata usata una rappresentazione basata sulle distanze che ricorda molto l'algoritmo di clustering (K-means). Ciò ci ha permesso di osservare l'implicazione che gli argomenti hanno l'uno sull'altro. Molti temi risultano essere ricorrenti in tutte e quattro le comunità. Strettamente collegato a questo c'è un altro tipo di raffigurazione (bar-charts Figura 19) che abbiamo ritenuto esemplificativi nella probabilità di trovare una parola all'interno di un tweet di una comunità specifica.



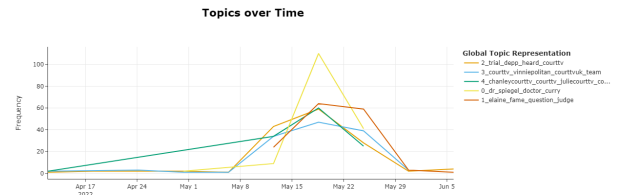
(a)



(b)



(c)



(d)

Figure 18: Andamento temporale dei tweet delle comunità estratte da Louvain: comunità 0 18a, comunità 1 18b, comunità 2 18c, comunità 3 18d

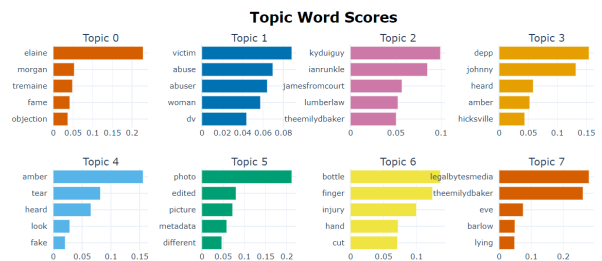


Figure 19: Frequenza degli argomenti nelle comunità estratte da Louvain nella comunità 0

Per esempio, il topic 0 identificato nella comunità 0 (Figura 19) vede come parola prevalente Elaine Bredehoft, lasciando pensare che l'argomento in questione riguardi Amber e le persone a lei vicine in questo processo; infatti Elaine è la sua avvocatessa.

7 DISCUSSION

Sin dall'inizio, l'intento di tale lavoro era quello di studiare le emozioni e le idee che trapelavano dal tweet scritti dai vari utenti in merito al processo e alle due personalità principali coinvolte: Johnny Depp e Amber Heard. È stata quindi posta l'attenzione sulla polarizzazione di opinioni fortemente sbilanciata verso Amber Heard; in particolare, la maggioranza dei tweet riguardavano le sue bugie, le sue lacrime, i testimoni da lei chiamati, gli avvocati che l'hanno difesa, fino a passare a temi più profondi come l'abuso domestico e la violenza di genere. Si può quindi affermare senza timore di essersi schierati da una parte piuttosto che da un'altra e che il processo ha fatto emergere delle tematiche che vanno al di là della disputa interna alla coppia Depp e Heard. Questo ci ha permesso di adottare tecniche per studiare la comunità

mediatica attorno a questo processo e la natura degli utenti che emerge da questi tweet.

Possibili sviluppi futuri potrebbero riguardare lo studio nel tempo delle comunità, in relazione alle diverse fasi, anche temporali, del processo.

REFERENCES

- [1] Albert-László Barabási. [n.d.]. *Network Science by Albert-László Barabási*. <http://networksciencebook.com/>
- [2] David Easley and Jon Kleinberg. [n.d.]. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press.
- [3] Maarten Grootendorst. [n.d.]. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv:2203.05794 [cs] <http://arxiv.org/abs/2203.05794> type: article.
- [4] Manika Lamba and Madhusudhan Margam. [n.d.]. Sentiment Analysis. ([n. d.]), 191–211. https://doi.org/10.1007/978-3-030-85085-2_7
- [5] Giulio Rossetti, Letizia Milli, and Rémy Cazabet. [n.d.]. CDLIB: a python library to extract, compare and evaluate communities from complex networks. 4, 1 ([n. d.]), 1–26. <https://doi.org/10.1007/s41109-019-0165-9> Number: 1 Publisher: SpringerOpen.
- [6] Giulio Rossetti, Letizia Milli, Salvatore Rinzivillo, Alina Sirbu, Dino Pedreschi, and Fosca Giannotti. [n.d.]. NDlib: a python library to model and analyze diffusion processes over complex networks. 5, 1 ([n. d.]), 61–79. <https://doi.org/10.1007/s41060-017-0086-6>