*Progetto per l'esame di **Social Media Analytics***
Corso di Laurea in Data Science (a.a 2018/2019)

Lavoro a cura di:

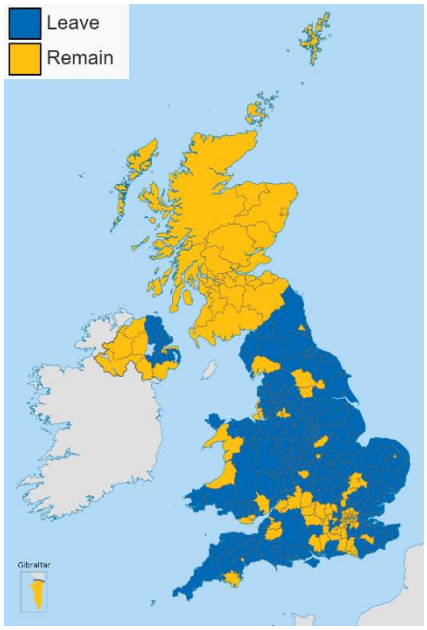
Luca Gabellini (777786)
Matteo Provasi (782922)
Pierluigi Tagliabue (835211)

BREXIT TWEETS' ANALYSIS

BACKGROUND

Il 23 giugno 2016 si è tenuto il referendum popolare nel Regno Unito per decidere la permanenza o l'uscita dall'Unione Europea (EU). A dispetto di quelli che erano i sondaggi, il fronte dell'uscita (*leave*) ha prevalso con il 51.9% delle preferenze. Questo risultato inaspettato ha portato alla dimissione, fra diverse cariche dello stato, del Primo Ministro David Cameron succeduto dalla Premier Theresa May.

Il Regno Unito è il primo paese membro dell'EU ad iniziare le trattative per un'uscita dall'Unione, la novità e l'unicità di questa situazione ha portato a delle lunghe trattative (non ancora completamente concluse in data odierna) per regolare le future relazioni e la posizione del Regno Unito nei confronti degli altri stati membri.



La fase travagliata della brexit (**Br**itain **exit**) è amplificata dalle forti divisioni interne al Regno (mappa con le preferenze distrettuali sul voto **Fig.1** a lato): l'Inghilterra è favorevole all'uscita, con l'eccezione della città di Londra mentre Irlanda e Scozia sono estremamente a favore per restare nell'EU (*remain*). Quest'ultima ha visto il referendum sull'indipendenza dal Regno Unito tenutosi nel 2014 respinto proprio per i timori di un'esclusione della Scozia dall'EU.

In data odierna l'accordo formale (deal) per l'uscita del Regno Unito è stato trovato ma non è stato approvato dal Parlamento inglese. Parte degli accordi stipulati prevedono il pagamento di 40 miliardi di sterline per assolvere agli impegni del Regno Unito già intrapresi con l'EU, l'agevolazione per gli stranieri europei presenti nel territorio almeno fino al 2020 (data di fine della transazione della brexit) che non subiranno ripercussioni per l'uscita dall'Unione. Questo risultato non era certo quello sperato dai sostenitori della brexit, in quanto non si concretizza la rottura netta con l'EU annunciata nel referendum ed è una delle ragioni per cui non è stato ratificato.

Altro punto fondamentale è quella del confine tra Irlanda del Nord (parte del Regno Unito) ed Irlanda: il confine è stato oggetto di una guerra civile fino a due decenni fa fino a quando non si è deciso un confine senza dogana e libero scambio. L'uscita del Regno Unito comporterebbe complicazioni al confine che non presenta neanche delle barriere naturali. Vista la delicatezza del confine sia il governo di Londra che l'EU cerca di evitare la condizione di *hard border* mediante un meccanismo di scambio doganale dettato dal *backstop*. L'EU ha accettato l'idea di un'unione doganale dopo il periodo di transizione ma con le sue condizioni, tra cui l'accettazione delle leggi di concorrenza europee per il Regno Unito.

Questi accordi dovevano essere votati in Parlamento ma hanno ottenuto una secca respinta da parte dei parlamentari. Delle tensioni si sono create all'interno del partito della May che ha anche proposto una mozione di sfiducia che non è passata. L'UE non vuole ritrattare gli accordi raggiunti e Theresa May deve trovare una soluzione per scongiurare l'ipotesi di uscita con la mancanza di accordi bilaterali (*no deal*) che provocherebbe danni ingenti all'economia britannica.

Nella serata del 29 gennaio 2019 si è tenuta una riunione parlamentare su diversi emendamenti relativi alla brexit. Gli emendamenti che sono passati riguardano la possibilità di trovare accordi alternativi con l'EU al posto del backstop e il governo si impegna ad evitare il no deal; d'altro canto viene esclusa la possibilità di slittare la brexit oltre la data prefissata dal 29 marzo 2019.

OBIETTIVI

Vista l'importanza della giornata per il Regno Unito si è deciso di scaricare i tweet postati durante il 29 gennaio relativi alla brexit, su cui sono state effettuate una sentiment analysis e una network analysis. Gli obiettivi del progetto sono:

- valutare se sussistono variazioni significative del sentimento a seconda delle diverse finestre temporali in cui sono stati postati i tweet (in particolare prima e dopo la votazione parlamentare);
- analizzare il network degli utenti per comprendere quali siano i nodi più importanti e ricercare l'eventuale presenza di comunità nella rete.

DATI

Per svolgere la nostra analisi è utilizzata una API messa a disposizione da Twitter che consente agli utenti di effettuare lo streaming. Per effettuare lo streaming si è utilizzato sia Python (sentiment analysis) sia R (community detection). In Python i tweet sono stati raccolti in *live streaming* nella giornata del 29 gennaio prima e dopo la votazione (a partire dalle 17:50 UTC circa fino alle 23:35 UTC), per quanto riguarda R i tweet sono stati scaricati i giorni successivi in *Batch.* Per delle problematiche relative all'API i dati in un intervallo di tempo fra le 18:00 e 18:10 UTC non sono stati scaricati.

Streaming in Python

Per accedere alle API Twitter, è stata utilizzata `Tweepy`, una libreria Python che supporta l'autenticazione OAuth e ne consente il collegamento grazie a quattro token personali (consumer key, consumer secret, access token e access secret).

Quest'ultimi sono messi a disposizione dell'utente direttamente da Twitter dopo la creazione di una nuova applicazione e vengono utilizzati attraverso la creazione dell'istanza OAuthHandler. Per effettuare lo streaming dei tweets si è deciso di ricorrere al metodo "on status", offerto dalla libreria `Tweepy`, definito all'interno della classe "`listener`" ereditata da `StreamListener`.

La funzione restituisce un elevato numero di variabili relativi ai tweet ed agli utenti, è stato scelto un sottoinsieme degli attributi che si è ritenuto essere utile ai fini del progetto. I tweets vengono scaricati in formato *.json* e poi convertiti in stringa attraverso il comando "`json.dumps`" .

Le *keywords* inserite all'interno del filtro per lo streaming sono state:

```
brexit
#brexit
#hardbrexit
```

```
hardbrexit
#stopbrexit
stopbrexit
```

```
#TheresaMay
TheresaMay
```

È importante dire che lo streaming è stato effettuato su due diversi computer per evitare eventuali perdite di dati (si è comunque verificata una perdita di dati in un intervallo di 10 minuti).

Per salvare i tweets uno ad uno si è fatto riferimento prima ad un dizionario appositamente creato e successivamente i tweets sono stati salvati all'interno di un file *.csv*. Una volta ultimato il processo di streaming dei tweets i diversi file *.csv* ottenuti sono stati successivamente concatenati e manipolati nella fase di *pre_processing.*

La struttura di ogni record presente nel file comprende:

- **Author**: nome dell'utente Twitter, autore del tweet.
- **Username**: username univoco dell'utente Twitter, autore del tweet.
- **Text**: testo presente all'interno del tweet.
- **Hashtag**: Hashtag presenti nel testo del tweet.
- **Location**: luogo indicato dall'utente come residenza.
- **Data**: data e orario di pubblicazione del tweet.
- **Retweet**: variabile booleana che indica se un tweet è un retweet o no.
- **Name_Author_Original_Tweet**: nome dell'utente Twitter che è stato retwettato.
- **Username_Original_Tweet**: username univoco dell'utente Twitter che è stato retwettato.
- **Text_Original_Tweet**: testo originale del retweet.
- **Original_Hashtag**: Hashtag presenti nel testo originale del retweet.

Pre-processing

Una volta concluso il processo di streaming i diversi file *.csv* precedentemente realizzati su Python sono stati caricati e tramite la libreria `Pandas` concatenati creando così il data frame finale nominato *"DB Finale"* contenente tutti i tweets scaricati. Si è poi proceduto eliminando i duplicati ottenendo così un totale di $225'235$ tweets.

È stato poi risolto un problema relativo al testo dei retweet superiori a 140 caratteri, i quali risultavano troncati e quindi non particolarmente significativi per le nostre analisi. Per aggirare questo problema è stato necessario utilizzare il testo originale del retweet, presente all'interno della colonna "Text_Original_Tweet", collegando all'inizio del testo il tag "`@ RT Username:`" dove `Username` era la username dell'utente che ha retweettato. In questo modo siamo riusciti ad ottenere il testo completo dei retweet superiori a 140. Una volta ottenuti tutti i tweet aventi corretta lunghezza li abbiamo sostituiti agli originali troncati.

Per effettuare la sentiment analysis si è fatto riferimento solamente alla variabile indicante il testo del tweet. Sono state applicate le principali tecniche di pre-preocessing per documenti testuali:

- **Tokenizzazione**: divisione della frase in parole singole, token.
- **Normalizzazione**: divisa in lower case (parole in minuscolo) e rimozione dei segni di punteggiatura e di caratteri speciali; prima di questo passaggio sono state rimosse le parole che iniziavano con "#" o "@" in quanto hashtag o citazioni di altri utenti.
- **Lemmatizzazione**: le parole sono portate alla loro radice, sono rimosse le inflessioni del linguaggio come coniugazioni verbali, declinazioni, plurali.
- **Rimozione stopwords**: libreria tratta dal pacchetto `nltk` che contiene un elenco di parole altamente frequenti nel linguaggio parlato, principalmente articoli e preposizioni più alcuni verbi molto utilizzati. Questa lista predefinita solitamente è utile per analisi del testo, tuttavia per l'analisi del sentiment è consigliabile tenere più parole possibili per poter ottenere dei risultati più accurati. Sono state rimossi solamente le parole come le preposizioni o articoli

ininfluenti per il sentimenti, mentre alcuni avverbi e forme verbali negative (che ribaltano il sentiment di pozioni di frasi) sono stati tenuti.

Fig. 2 *Wordcloud basata sui tweet*

Da questo punto si sono sistemati manualmente alcuni termini specifici che non sono stati trattati correttamente dalle fasi di pre-processing elencate in precedenza, inoltre acronimi di organizzazioni ed aziende sono state portate in maiuscolo.

Per avere un'idea generale di quelli che erano i termini globalmente più frequenti nei tweet si è creata una world cloud (figura a lato a destra). Come era lecito attendersi i termini "brexit" e "Theresa" sono fra i più frequenti (senza includere nel conteggio queste parole come hashtag), di grande rilevanza anche le parole sul voto, i membri del parlamento, elementi riguardanti le trattative come "backstop" e altri nomi di politici.

SENTIMENT ANALYSIS

Il dataset ottenuto dalle fase di pre-processing è stato esportato da Python ed importato in R per la sentiment analysis. È risultata necessaria un'ulteriore modifica al dataset in quanto la data salvata nel file *.csv* era letta correttamente, ma non nel formato per poter implementare rappresentazioni basate sul tempo, è stata quindi necessaria una trasformazione con la funzione `as.POSIXct`.

Da queste date si sono ottenute delle sequenze temporali dall'ampiezza di 5 minuti, la prima che parte dalle ore $17{:}50$ UTC e l'ultima fino alle $23{:}35$ UTC. Vista la finestra temporale limitata dei tweet si è resa necessaria una segmentazioni ad intervalli così brevi per poter analizzare meglio delle variazioni di sentiment. L'interesse non era tanto quello di osservare globalmente quale fosse il sentiment generale della gente riguardo al topic della brexit, anche perché non è immediata la connessione fra un sentiment positivo o negativo ad una delle due posizioni. Quello che ci interessava analizzare era un eventuale cambio repentino nel sentiment in un breve intervallo di tempo che coincidesse con l'ora in cui il Parlamento inglese votava i vari emendamenti presentati in giornata.

Si sono create delle liste che per ogni intervallo di tempo indicava quali osservazioni del dataset, in base al loro ordine, dovevano essere connesse a quel determinato intervallo. A parte due intervalli di tempo che non presentavano osservazioni per problemi con l'API già citati in precedenza, per ogni intervallo di 5 minuti si trovavano almeno 1000 osservazioni. Si è ritenuto che questo fosse un numero consistente di osservazioni da non rendere la sentiment valutata su questi intervalli affetta da problemi di undersampling.

Per ogni porzione del dataset si è effettuata una sentiment analysis con quattro diverse tecniche:

- nrc: comprende una lista di parole collegata a 8 emozioni differenti (rabbia, anticipazione, disgusto, paura, gioia, tristezza, sorpresa, confidenza) e due sentimenti (negativo e positivo), le attribuzioni sono state effettuate manualmente tramite un programma di crowdsourcing.
- Afinn: utilizza un dizionario più ristretto in cui le parole hanno associato un sentiment che varia nel range $(-5,5)$. Fra tutti i dizionari utilizzati è quello che ammette un range di valori più ampio.
- Bing: a differenza dei dizionari precedenti comprende una serie di parole nel che sono volutamente scritte in maniera errata simulando il comportamento degli utenti all'interno dei social, presenta inoltre slang e abbreviazioni. Sebbene dovrebbe essere un vantaggio sugli altri metodi, ci sono delle note negative, come il fatto di non riuscire a dare un sentiment adeguato al variare della lunghezza della frase.
- Syuzhet: è un metodo che è stato studiato per poter rilevare il sentiment su testi lunghi composti da più frasi. Nel nostro caso purtroppo non si poteva sfruttare la particolarità di questo metodo in quanto ogni tweet è composto da una frase o un insieme di frasi breve non paragonabili ad un testo letterario, quindi è paragonabile ad un metodo *Afinn* ma con dizionario diverso ed un sistema di pesi diverso (è l'unico metodo che per ogni parola assegna anche punti decimali e non interi). Presenta delle limitazioni per quanto riguarda le ripetizioni di termini o emozioni ravvicinate, in questi casi sia senza che con ripetizioni il valore di sentiment assegnato tiene in considerazione uno solo dei termini ripetuti.

Per tutti i metodi sono stati salvati i risultati ad ogni iterazioni riguardanti il sentiment, per il metodo *nrc* sono state salvate anche le emozioni.

Il sito online de The Guardian [1] ha seguito live la seduta parlamentare provvedendo a dare informazioni in tempo reale sulle votazioni dei diversi emendamenti. Il sito è stato monitorato durante la giornata del 29 gennaio ed è stato molto utile per poter individuare il momento preciso in cui un determinato emendamento veniva approvato o respinto. Il seguente grafico rappresenta il sentiment medio per ogni intervallo di 5 minuti in relazione alle emozioni di rabbia, anticipazione, tristezza e fiducia.

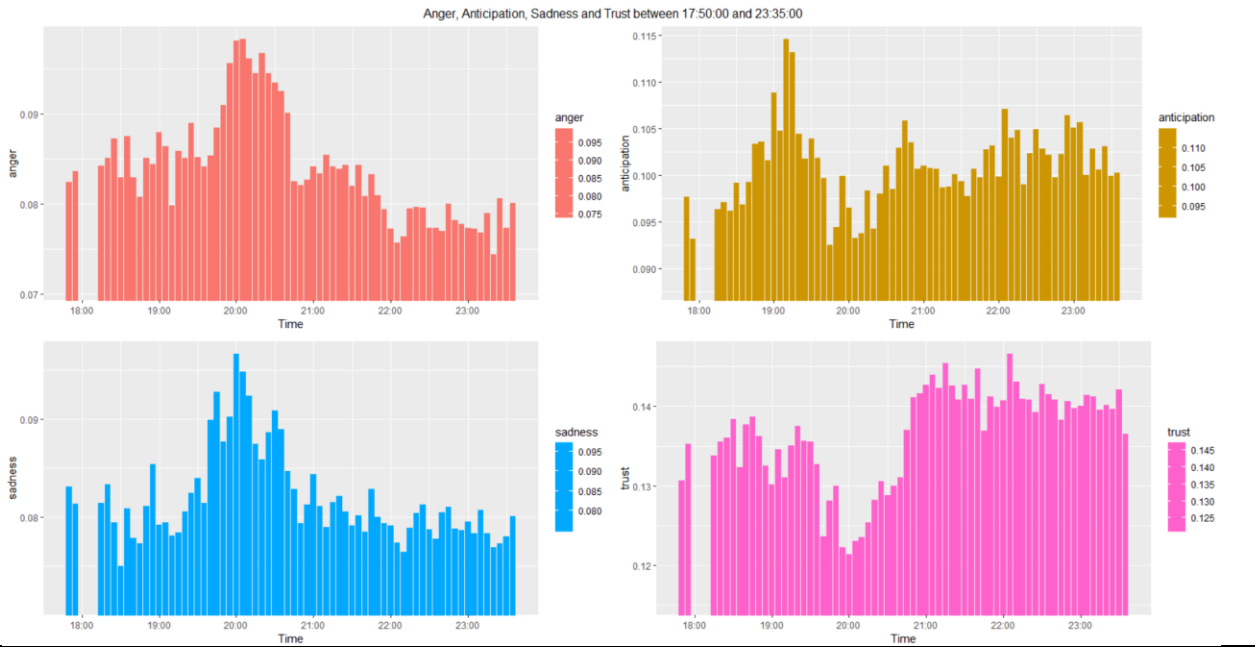


Fig. 3 *Emozioni ottenute mediante il metodo nrc*

[1] https://www.theguardian.com/politics/live/2019/jan/29/brexit-vote-commons-latest-news-developments-liam-fox-says-may-now-saying-withdrawal-deal-text-must-be-rewritten-politics-live?page=with:block-5c50a92ee4b037f7f77a339d#liveblog-navigation

Secondo il sito The Guardian le votazioni sono iniziate alle ore 19:06 UTC e terminate intorno alle 20:30 UTC. L'emendamento più importante, quello di un possibile slittamento delle brexit oltre il 29 marzo, è stato a sorpresa bocciato (alle ore 19:56 UTC), aumentando notevolmente le probabilità di un no deal. Dalle quattro emozioni precedenti è possibile notare come si verifichino dei picchi in negativo o in positivo intorno all'orario delle votazioni. Per *anticipation* il massimo è raggiunto poco dopo l'inizio delle votazioni, rabbia e tristezza hanno dei massimi intorno alle 20:00 UTC stessa ora in cui l'emendamento considerato chiave della giornata è stato bocciato, analogamente in questo orario il *trust* crolla a picco.

I trend evidenziati suggeriscono quindi delle possibili relazioni fra la variazione delle emozioni e le votazioni nel Parlamento. Variazioni così marcate suggeriscono che si sia sviluppato un malcontento generale dettato dai risultati delle votazioni indipendentemente dalla posizione politica dei soggetti. Questo aspetto è spiegabile in termini tecnici in quanto il tanto atteso slittamento del termine delle brexit avrebbe aperto le porte a delle nuove trattative che sarebbero potute continuare per più mesi portando a dei cambiamenti sostanziali (posizione sperata dal fronte del *leave*) o si sarebbe avuto il tempo per sfiduciare il governo o arrivare addirittura ad un secondo referendum (posizione sperata dal fronte del *remain*). L'impossibilità di estendere il limite aumenta notevolmente le probabilità di un'uscita senza accordo che non è visto neanche bene dal fronte del *leave* (tranne la parte più estrema, che rappresenta comunque una piccola minoranza) in quanto con questo scenario tutti gli indicatori economici indicano un repentino peggioramento dell'economia britannica.

Passando all'analisi del sentiment i risultati con i diversi metodi sono stati i seguenti:

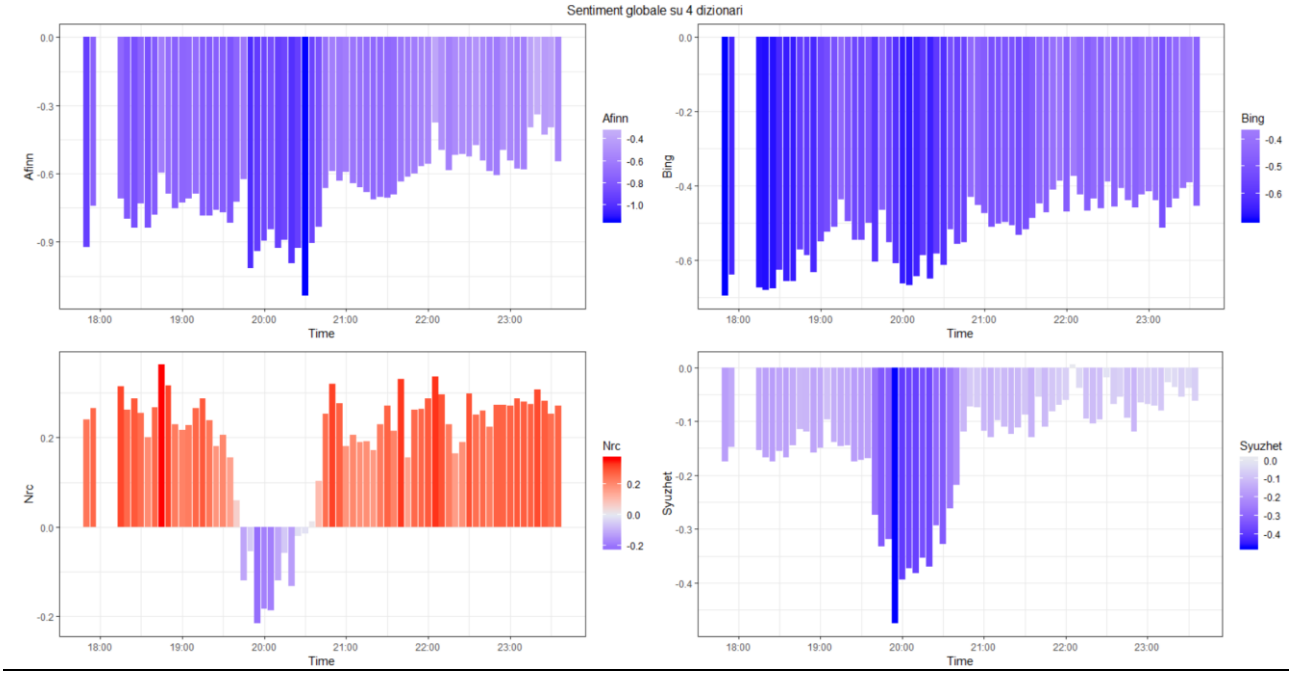


Fig. 4 *Sentiment a confronto fra i diversi metodi*

Tutti tranne il metodo *bing* identificano un picco in basso con aumento notevole di un sentiment negativo in corrispondenza dell'orario delle votazioni, a conferma ancora una volta che la reazione degli utenti su Twitter è stata prevalentemente di malcontento. Il metodo *Afinn* è quello che attribuisce valori inferiori in termini assoluti, probabilmente per il range di valori con cui è costituito il metodo, *syuzhet* ha valori molto contenuti tranne per il picco intorno alle 20:00 UTC, potrebbe

essere dato dal fatto che il metodo non coglie ripetizioni di termini all'interno dei tweet. Caso particolare è il metodo *nrc* che è l'unico che presenta valori positivi per tutto il tempo durante le votazioni. Come già detto nell'introduzione della sentiment analysis la parte più importante era quella di trovare un pattern ben definito in corrispondenza dell'evento di interesse, un fatto che si è decisamente verificato. Resta comunque interessante notare come il sentiment dopo un picco in negativo risale immediatamente per tutti i metodi ai livelli precedenti (lo stesso si verifica per le emozioni). L'aspettativa dato un picco negativo era quella di osservare un trend di calo del sentiment più esteso nel tempo di quello che in realtà è stato.

NETWORK ANALYSIS - BREXIT RETWEETS

Come analisi supplementare si è scelto di rappresentare in forma di grafo i retweet relativi alla Brexit. I tweet di interesse in questo caso sono stati ottenuti con il package rtweet del software R, con le stesse opzioni scelte in Python (tweet contenenti hashtag #brexit e tweettati nella stessa finestra temporale).

Dei 18000 tweets ottenuti con la funzione search tweets() ne sono stati utilizzati solamente 500: questa scelta ha permesso di condurre l'analisi in tempi di elaborazione accettabili, e inoltre ha reso possibile una chiara ed efficace visualizzazione grafica della rete. Inizialmente i 500 tweet vengono filtrati in modo tale da ottenere i soli retweet, in tutto 389.

Per la creazione del grafo viene creato un dataset di archi ottenuto dai retweet, composto da due colonne:

- retweeter: screen name dell'utente che ha retweettato
- original_user: screen name dell'utente che è stato retweettato.

Per identificare gli utenti, e quindi i nodi del grafo, vengono utilizzati gli username piuttosto che gli Author's name: i primi infatti identificano univocamente l'utente, mentre per i secondi l'univocità non è garantita. Dal dataset viene in seguito costruito un grafo pesato con la funzione graph_from_data_frame(). Il peso associato agli archi indica il numero di volte in cui un utente è stato retweettato dall'altro utente.

Il grafo ottenuto presenta 306 nodi, pari al numero degli utenti del campione scelto, e 361 archi. La densità (proporzione di archi esistenti su tutti gli archi potenziali del grafo) è molto bassa, per la precisione di 0.004.

A causa della natura dei dati grezzi, il grafo presenta molte componenti sconnesse: non sembra appropriato calcolare la closeness centrality come misura di centralità node-level in questo contesto. Risulta decisamente più opportuno adottare come misura di riferimento per la centrality il grado del nodo (In-degree, Out-degree).

I nodi che presentano un In-degree elevato corrispondono agli utenti più retweettati, gli utenti che hanno retweettato molto presenteranno un elevato Out-degree, risultano essere rispettivamente:

Utente	In-degree
CarolineLucas	31
RepBrendanBoyle	27
simoncoveney	26
joannaccherry	22
Anna_Soubry	16
nickreeves9876	9
GavNewlandsSNP	7

Utente	Out-degree
GNaomiMartin	10
GraceGrace901	9
FBPETrundlelin	8
LaughingSmurf	8
PaulK1954	8
Weesem7	8
NatashaPolitics	7

Il discorso fatto per la closeness centrality può essere esteso alla betweenness centrality: i valori di questa misura risultano nulli per quasi tutti i nodi, eccezion fatta per l'utente *TerryEnga*, che presenta una betwenness pari a 8, e per *STEFLEX_AHM* ($betweeness = 1$).

Analizzando il caso particolare di *TerryEnga*, utente molto attivo su Twitter e dichiaratamente europeista, si può notare come retweetti 4 differenti utenti, tra cui Caroline Lucas e Anna Soubry, entrambe contrarie alla Brexit; inoltre l'utente viene retweettato due volte, e rappresenta uno dei pochi utenti che retweetta e viene retweettato allo stesso tempo: è verosimilmente per questa particolarità che *TerryEnga* presenta una betweenness non nulla.

La seguente immagine rappresenta la rete che si sviluppa intorno all'utente *TerryEnga:*

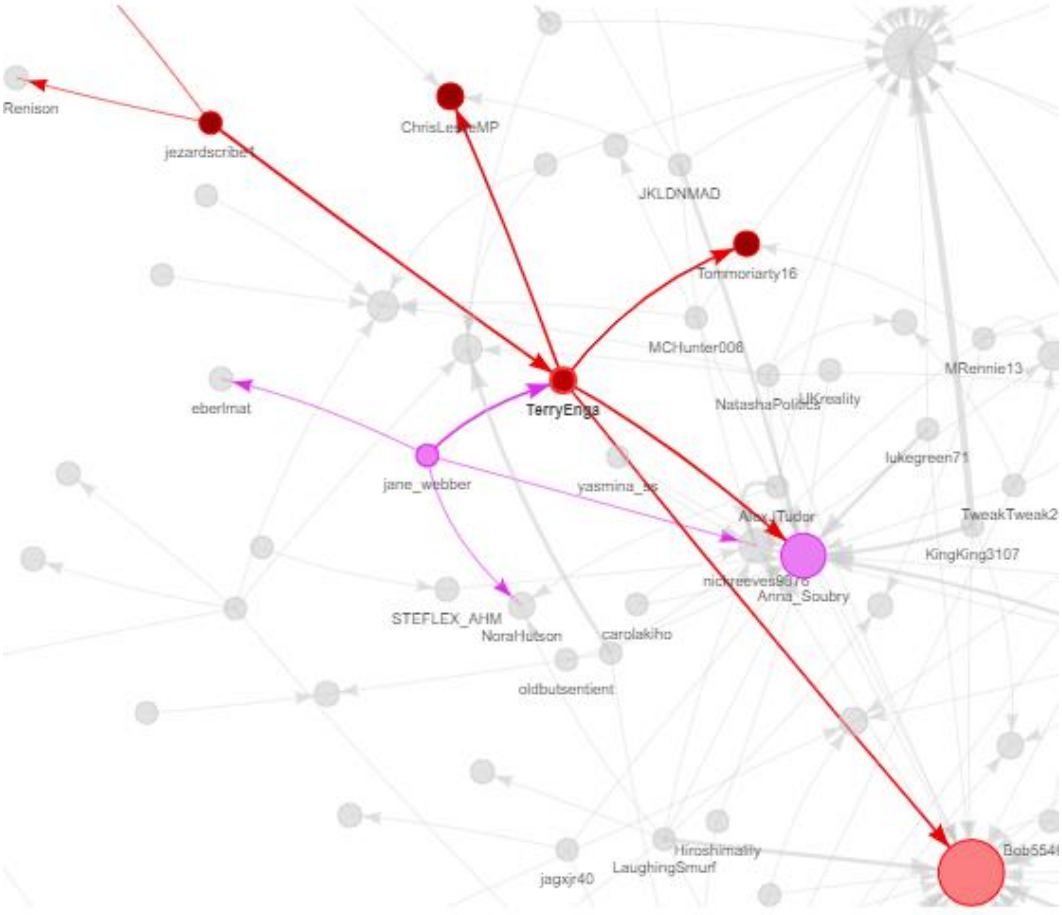


Fig. 5 *Community intorno all'utente TerryEnga*

Dato che la In/Out-degree centrality non tiene conto di connessioni ripetute tra coppie di nodi, vengono proposte come integrazione le misure di **Authoritativeness** e **Hubness**, calcolate utilizzando rispettivamente le funzioni `authority_score()` e `hub_score()` di R. In particolare:

- Gli authority scores dei vertici corrispondono agli autovalori principali di $t(A) \times A$, dove A è la matrice di adiacenza del grafo, mentre $t(A)$ è la trasposta di A^2;
- Gli hub scores dei vertici corrispondono agli autovalori principali di $A \times t(A)$.

I nodi più autorevoli, che corrispondono agli utenti più retweettati e i nodi con Hubness più elevata, ovvero gli utenti che hanno retweettato di più in questo caso risultano essere rispettivamente:

Utente	Authoritativeness
CarolineLucas	1.000
joannaccherry	0.793
euronews	0.672
Anna_Soubry	0.626
ACatInParis	0.448
simoncoveney	0.380
RepBrendanBoyle	0.295

Utente	Hubness
FBPETrundlelin	1.000
KingKing3107	0.658
LaughingSmurf	0.483
lukegreen71	0.421
NatashaPolitics	0.393
GraceGrace901	0.358
HWinckelmann	0.345

Per meglio comprendere il ruolo e lo schieramento di alcuni degli utenti più autorevoli, vengono presentate alcune informazioni salienti su di loro:

- **Caroline Lucas**: politica britannica, leader del Partito Verde di Inghilterra e Galles. Contraria alla Brexit, il suo tweet più retweettato tra quelli da noi analizzati è stato:

  ""“Let them go to the chippy instead” - DUP MPs muttering behind me when @IanBlackfordMP mentioned food prices rising after No Deal #Brexit.It won't be MPs who have to cope with worst impacts of No Deal. @duponline should be ashamed of their disregard for people they represent."

- **BrendanBoyle**: politico statunitense del Partito democratico, membro della Camera dei Rappresentanti per lo stato della Pennsylvania. Boyle in un’intervista ha mostrato preoccupazione per un eventuale ritorno di un hard border tra Irlanda e Irlanda del Nord. Il suo tweet più retweettato è stato:

  'After agreeing to the Irish backstop, Theresa May’s government has now reneged on it. Why would anyone negotiate with her now? #Brexit'

- **Simon Coveney**: politico irlandese che ha lavorato come Tánaiste (vice primo ministro irlandese) dal mese di novembre 2017. Coveney fu nominato Ministro degli Affari Esteri e del Commercio, con responsabilità speciali per la Brexit. È contrario a una rinegoziazione dei termini che includono il backstop, per evitare che si torni all’hard border. Il suo tweet più retweettato è stato:

  'Backstop was agreed by UK/EU as the insurance policy to avoid a hard border in all scenarios. We hope it will never be used, or be replaced quickly by a future relationship agreement. But it is necessary and tonight’s developments at Westminster do nothing to change this. #Brexit'

[2] J. Kleinberg, Authoritative sources in a hyperlinked environment. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998.

- Joanna Cherry: politico membro dello Scottish National Party (SNP). Cherry ha criticato a più riprese l'operato di Theresa May, ribadendo come il Primo Ministro e il suo governo abbiano trattato la Scozia in maniera irrispettosa. Il suo tweet più retweettato è stato:

  'Today I asked what kind of a PM drives a coach horses thru her own #Brexit agreement? Now we know. She'll need to relax her red lines to get a new agreement. It's hard to care. It will soon be time for #Scotland to take leave of this madness #indyref2'

- Anna Soubry: politico del British Conservative Party, avvocato e giornalista. Soubry è una forte sostenitrice dell'Unione europea e ha sostenuto la campagna *remain* durante il referendum sulla permanenza in UE del 2016. Il suo tweet più retweettato, con 304 retweet è stato:

  'The only MPs cheering tonight are members of the #ERG And a lot of #Labour MPs held thei heads in shame and scuttled off home #Brexit'

COMMUNITY DETECTION

Per ricercare l'eventuale presenza di comunità presenti nella rete, il package `igraph` di R mette a disposizione dell'utente un'ampia varietà di algoritmi. La maggior parte di questi però richiede in input un grafo non direzionato. Di conseguenza gli approcci idonei per la casistica di interesse, ovvero un grafo direzionato (retweet → tweet originale) con archi pesati, si riducono notevolmente.

L'algoritmo di community detection da noi utilizzato è `cluster_infomap()` [3]. L'idea che sta alla base dell'algoritmo è la seguente: se si vuole comprendere al meglio la struttura di un network bisogna analizzare i flussi informazionali che caratterizzano la rete. Un gruppo di nodi in cui l'informazione fluisce rapidamente e in quantità maggiore può essere visto come una well-connected component, e quindi come una comunità. L'algoritmo utilizza dei random walks come indicatori di flussi informazionali del sistema; comprime l'informazione derivante dai random walks per identificare le strutture basilari candidate per essere considerate comunità di nodi.

Più formalmente l'algoritmo infomap si basa sull'ottimizzazione della seguente map equation:

$$L(M) = q_{\curvearrowright} H(Q) + \sum_{i=1}^{m} p_{\circlearrowright}^{i} H(P^{i})$$

che ricerca una partizione M di n nodi suddivisi in m moduli per minimizzare la lunghezza della descrizione attesa di un random walk. Il primo termine dell'equazione equivale al numero medio di bits necessario per descrivere il movimento tra differenti comunità; il secondo termine equivale al numero medio di bits utili per descrivere il flusso informazionale tra nodi di una stessa community.

Per una descrizione più dettagliata dell'algoritmo si rimanda al paper di Martin Rosvall e Carl T. Bergstrom, ideatori dell'algoritmo [3].

L'algoritmo ha partizionato i 306 nodi in ben 62 comunità, molte delle quali composte da non più di due nodi. La modularità calcolata dall'algoritmo è di 0.672, indice del fatto che i vertici sono ben separati tra i vari moduli e si allontanano significativamente dalla distribuzione attesa di archi.

[3] M. Rosvall and C. T. Bergstrom, Maps of random walks on complex networks reveal community structure.

	title <fctr>	group <dbl>
CarolineLucas	CarolineLucas	3
joannaccherry	joannaccherry	6
euronews	euronews	14
Anna_Soubry	Anna_Soubry	2
ACatInParis	ACatInParis	14
simoncoveney	simoncoveney	4
RepBrendanBoyle	RepBrendanBoyle	1

Fig. 6 *Utenti e il loro gruppo di appartenenza*

Dalla tabella, e analizzando il grafo interattivo riportato nel file `html`, emerge una tendenza: le comunità più numerose sono caratterizzate da un singolo nodo autorevole (con alto numero di nodi in ingresso) e dai suoi seguaci, ovvero gli utenti che lo hanno retweettato. Nell'immagine successiva uno screen del grafo interattivo:

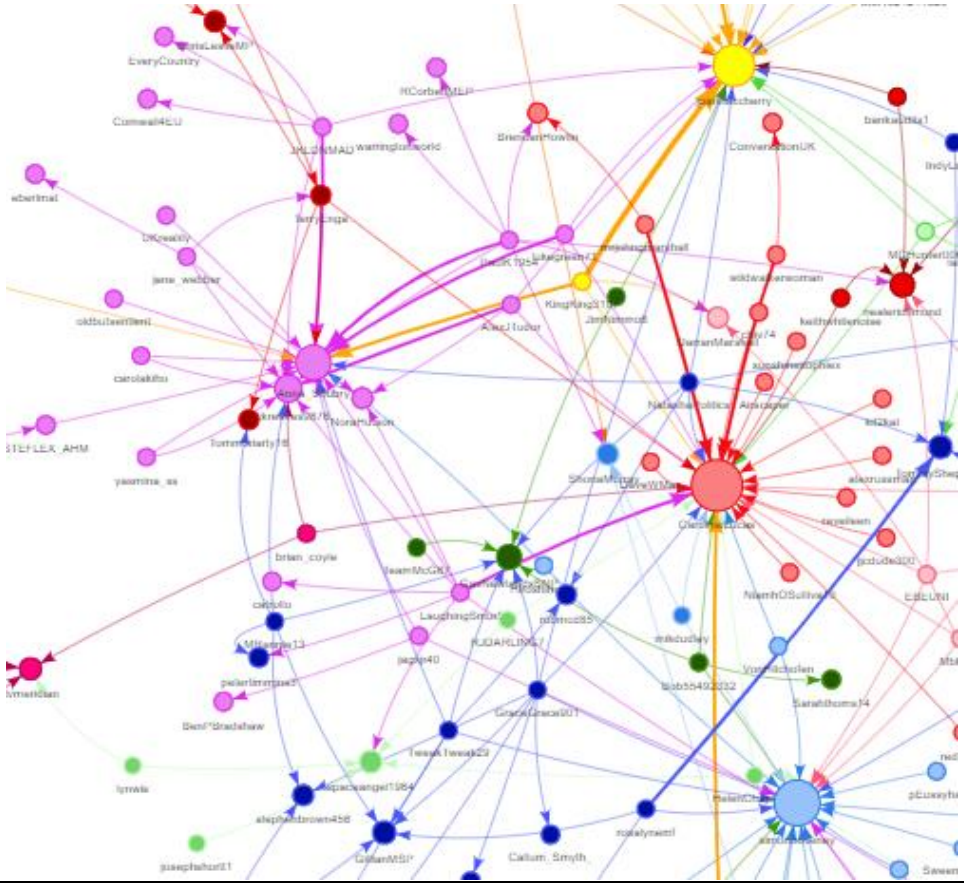


Fig. 7 *Porzione del grafo interattivo creato*

Il campione di 500 tweets considerato per la network analysis è stato ritenuto rappresentativo del network più esteso relativo ai tweet sulla brexit ottenuti da python, su cui è stata condotta l'analisi del sentimento; di conseguenza non è stato necessario effettuare un campionamento stratificato o di altro tipo.