Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea in Informatica

Progettazione e Sviluppo di una Web Application per l'Analisi di Reti Sociali in ambito Forense

Relatore:
Prof. Danilo Montesi

Correlatore:
Dott. Flavio Bertini

Presentata da:
Leonardo Lucente

Sessione III
Anno Accademico 2019/2020

Abstract

L'obiettivo di questo lavoro è di progettare e sviluppare una applicazione web il cui utilizzo sia quello di poter condurre analisi sociali andando ad elaborare dati provenienti da diverse fonti sociali quali: Facebook, Twitter e mail box. Per rendere possibili queste analisi abbiamo focalizzato la nostra attenzione, non solo sulla costruzione visiva di un reticolo sociale, ma anche sulla diffusione dei contenuti nel tempo e nello spazio e sulla loro estrazione testuale. Siamo partiti definendo il concetto di rete sociale ed abbiamo proseguito evidenziando come l'analisi sociale possa essere di supporto alle analisi forensi, fornendo alcune tecniche specifiche per questo ambito. Abbiamo, infine, messo a paragone alcuni strumenti di analisi forense attualmente presenti sul marcato. Questa panoramica ci ha permesso di definire il nostro campo di interesse e le motivazioni dietro il progetto che hanno portato allo sviluppo di un'applicazione web in grado di analizzare e rappresentare visivamente i dati in ingresso. Andremo a definire l'architettura utilizzata e presenteremo un caso d'uso reale utilizzando i dati provenienti dai miei canali social.

Indice

Capitolo 1

Introduzone

1.1 Panoramica

I social network ricoprono, ormai da anni, un ruolo di rilievo nella vita di ciascuno di noi. Questo grazie alla sempre crescente proliferazione e all'accessibilità economica di dispositivi abilitati alla connessione Internet. Con il termine "Social Network" siamo soliti riferirci, ad esempio, ai più comuni Facebook e Twitter che rappresentano gli esempi più lampanti di "rete sociale", ma spesso tendiamo a dimenticare che anche le nostre caselle di posta elettronica rappresentano una di queste reti. Ognuno di questi sistemi rappresenta un mezzo per rimanere in contatto con i nostri familiari, con i nostri amici o conoscenti e con cui siamo soliti condividere indifferentemente situazioni della nostra vita lavorativa o della nostra sfera privata. Proprio per le interazioni che andiamo a costruire attraverso i social network, risulta interessante un'analisi degli stessi, in quanto questo ci permette di indagare in maniera approfondita su diversi aspetti riguardanti la vita di ciascuno di noi.

1.2 Motivazioni e obiettivi

Indifferentemente dal modello sociale che si va ad analizzare, ciascuno dei contenuti che condividiamo attraverso la piattaforma, viene salvato all'interno del sistema

con metadati che possono essere utili per effettuare una prima basilare analisi della rete sociale stessa. Basti pensare alla possibilità di poter estrarre la lista di amici per quanto riguarda Facebook o la lista di follower per quanto riguarda Twitter, o, in maniera del tutto simile, la lista dei contatti per un mailbox. Questa operazione, per quanto lunga possa essere da effettuare a mano, è relativamente semplice, mentre, ad esempio, estrarre tutti i messaggi contente una specifica parola può essere un'operazione complessa, soprattutto se l'insieme dei dati di partenza è molto grande.

Il problema principale di trattare questo genere di dati, è, infatti, il volume dei dati stessi. Da questa problematica ne deriva che diventa complicata anche una visualizzazione che possa risultare esplicativa per l'utente finale.

Il nostro obiettivo è quello di creare un applicativo che sia in grado di elaborare una grande quantità di dati, ma anche di rappresentare in maniera semplice e chiara questi dati, in modo tale che possano essere utili ai singoli utenti per avere una panoramica dei loro contenuti condivisi nel tempo, ma anche che sia utile anche in campo giudiziario per poter estrarre risultati utili ai fini di un'indagine. Al termine, avremo creato un framework che ingloberà sia caratteristiche di applicazioni già esistenti, ma anche e soprattutto nuove funzionalità utili al nostro fine. Questo sarà la combinazione di diverse tecniche quali: estrazione testuale, analisi forense, elaborazione dei dati e ovviamente visualizzazione dei dati

1.3 Strumenti di ricerca

Per la realizzazione di un applicativo che sia di utilizzo generico, quindi non circoscritto all'utilizzo di un solo social network o di un solo compito avremo bisogno di far confluire nel progetto finale diverse tecniche. In primo luogo, come già evidenziato, la nostra applicazione deve essere in grado di elaborare diverse reti sociali, nello specifico Facebook, Twitter e mailbox, e quindi avremo bisogno di strumenti di elaborazioni dei dati per questi social, ma anche di metodologie che, una volta analizzati questi dati, siano in grado di elaborarli sia in forma visiva, ma anche

testuale. L'analisi testuale stessa ricoprirà un ruolo fondamentale all'interno della nostra applicazione proprio perché un analisi forense e giudiziaria passa in maniera obbligatoria da un analisi testuale. Affronteremo prima un confronto con altri strumenti simili attualmente disponibili e concluderemo fornendo una panoramica sull'implementazione del nostro applicativo e su un possibile esempio di utilizzo.

Capitolo 2

Stato dell'arte

Per svolgere una corretta analisi che possa avere risvolti anche in campo legale avremo bisogno di esaminare diversi aspetti che, mescolati insieme, possano risultare utili ai fini del nostro applicativo.

Prenderemo in considerazione diversi settori. Partiremo dall'acquisizione ed elaborazione dei dati, continueremo con l'estrazione o *mining* sia dei dati stessi, ma anche e soprattutto dei contenuti testuali che vengono condivisi dall'utente attraverso i vari social network e concluderemo con la loro visualizzazione.

Passeremo in rassegna, inoltre, anche altri lavori che abbracciano il medesimo campo di studio. Questo risulterà importante per sottolineare l'importanza che ricopre un'analisi sociale e ci permetterà di evidenziare le differenze tra gli applicativi già esistenti e il nostro.

Prima di iniziare, però, illustreremo cosa si intende con *social netowork analysis* e come questo concetto sia strettamente correlato a quello di grafo. Andremo ad indicare cosa si indica con il termine "grafo", e come questi siano di assoluta utilità nell'analisi sociale e, nello specifico, anche al nostro obiettivo. Il concetto di grafo sarà alla base del nostro lavoro e per questo motivo inizieremo fornendo delle informazioni preliminari su questo argomento che aiuteranno il lettore a inquadrare meglio il nostro campo di ricerca.

2.1 Social network e social network analysis

L'obbiettivo del nostro lavoro, come già detto, è quello di sviluppare un applicativo che possa essere di supporto alle attività forensi nelle analisi delle reti sociali. Anche se il termine *social network* è diventato di uso corrente e il suo significato ci può sembrare banale, è bene sottolineare che cosa si intenda per rete sociale. Come descritto in [1] un *social network* è concepito come una rete di interazioni o relazioni, dove i nodi sono costituiti da attori, e gli archi consistono nelle relazioni o interazioni tra questi attori. Spesso, con questo termine, ci si riferisce erroneamente alle sole applicazioni basate su internet, come ad esempio lo è Facebook, tuttavia possiamo estendere il suo significato a qualsiasi interazione generica tra qualsiasi gruppo di attori. Queste relazioni non devono essere per forza avvenire per via telematica attraverso lo scambio di *"post"* o *"tweet"*, ma una rete sociale è costituita da qualsiasi genere di relazione sia essa avvenuta personalmente, per via telematica, via posta tradizionale o via e-mail.

2.1.1 SNA

Sebbene gli studi riguardanti le reti sociali siano antecedenti ai social network stessi, è con l'affermarsi di questi ultimi, grazie alla diffusioni di dispositivi di telecomunicazione, che è sorta anche la necessità di svolgere attività di analisi supportate da applicativi che permettessero un'elaborazione automatica dei dati. Attraverso queste, infatti, diventa possibile cogliere spunti interessanti di ricerca per quanto riguarda la vita di una persona. Attraverso un'analisi di una rete sociale, difatti, si potrebbero ricavare dati personali riguardanti un utente o identificare le attività da esso svolte online. Da questa necessità si è diffusa la **social network analysis** o più brevemente SNA che, come suggerisce il nome, si occupa di studiare la reciproca influenza tra individui tramite processi di comunicazione. La SNA si rileva utile qualora si voglia analizzare la rete sociale di una persona per poterne ricavare, ad esempio, eventuali schemi comportamentali o altre informazioni legate all'attività sociale online.

Per quanto ciascuna rete sociale sia completamente differente rispetto alle altre, tutte sono accumunate da alcuni aspetti trasversali che riassumono la struttura di qualsiasi social network:

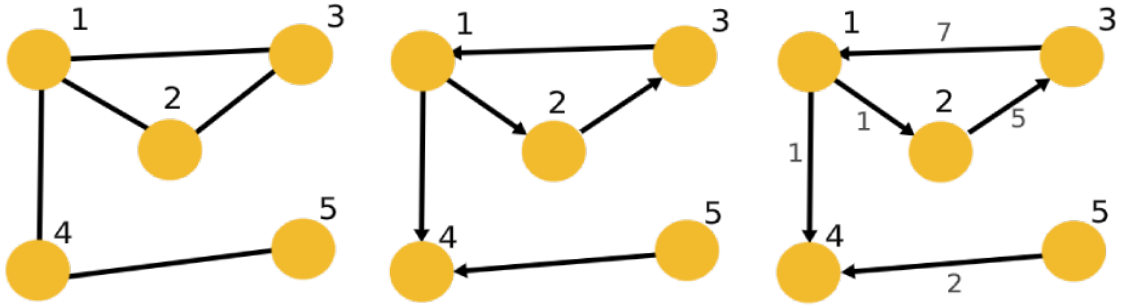
- **attori**: gli attori rappresentano tutti gli utenti appartenenti ad una specifica rete
- **relazioni tra attori**: indica la possibilità di instaurare rapporti tra i vari utenti
- **condivisione di contenuti**: indica la possibilità di condividere messaggi, foto o video con la nostra rete

La maggior parte degli studi riguardanti la social network analysis si sono concentrati sull'esplorare nuove modalità di visualizzazione per rappresentare una rete sociale e rendere la stessa il più esplicativa possibile. Lavori come [16] si basano su di una visualizzazione che fa uso dei diagrammi di Eulero in combinazione con una struttura ad albero e che si concentra sul rappresentare interessi e rapporti tra i vari attori, mentre altri come [20] utilizzano una visualizzazione temporale, dove i vari termini dei contenuti condivisi con la rete vengono posti su di una linea temporale. Ciononostante, il modello più immediato per rappresentare una rete sociale è tramite l'utilizzo dei grafi. Quest'ultimi, infatti, mostrano la naturale struttura di una rete sociale e costituiranno il punto fulcro del nostro lavoro. Per questo motivo, nelle prossime sezioni, continueremo con una presentazione dei grafi, che si rivelerà utile al conseguimento del nostro obiettivo finale.

2.2 Introduzione ai grafi

Nella sezione precedente abbiamo affermato che una rete sociale altro non è che una rete di relazioni tra attori. Ecco, quindi, che questa nostra presentazione può essere ridotta al semplice concetto di grafo, così come affermato in [3]. In questo contesto gli attori della rete sociale diventano i nodi (chiamati anche vertici) all'interno del

grafo, mentre le relazioni tra di essi vengono chiamate archi. Gli archi a loro volta, nel caso in cui abbiano un verso, sono definiti orientati, al contrario vengono definiti non orientati.



(a) Grafo non orientato (b) Grafo orientato (c) Grafo orientato pesato

Figura 2.1: Esempi di grafi

I grafi dove gli archi sono privi di verso vengono definiti grafi non orientati, al contrario vengono definiti orientati se gli stessi archi presentano un verso. In quest'ultimo caso va fatta una distinzione qualora gli archi abbiano dei pesi. Nel caso in cui sia importante l'intensità delle relazioni si da luogo al grafo in figura *2.1 (c)*, in caso contrario si origina un grafo come in figura *2.1 (b)*

Indipendentemente dal fatto che il grafo sia diretto o indiretto, si definisce grafo $G = (V, E)$ la coppia dove:

- V costituisce l'insieme dei nodi del grafo G

$$V = \{v_1, v_2, ..., v_i\} \tag{2.1}$$

- E costituisce l'insieme degli archi del grafo G

$$E = \{e_1, e_2, ..., e_j\} \tag{2.2}$$

Due nodi collegati da un arco vengono definiti **adiacenti**, l'insieme dei nodi adiacenti ad uno specifico nodo v_i rappresenta il **vicinato** di quel nodo, mentre la somma dei nodi adiacenti al nodo v_i costituisce il **grado di connessione** di v_i. Da questa

affermazione ne consegue che il grado di un nodo è il valore numerico che esprime la dimensione del suo vicinato e che ad un aumento di tale valore corrisponde un maggiore connessone della rete.

Queste proprietà appena elencate, sebbene ad una prima lettura potrebbe risultare banali, troveranno un riscontro pratico all'interno del nostro lavoro come vedremo più avanti.

2.3 Misurazione dei grafi

La rappresentazione più semplice ed immediata di un grafo è la visualizzazione che avviene su un piano raffigurando i nodi come delle circonferenze collegati tra loro da delle linee. Questo tipo di visualizzazione è di un grande impatto visivo in quanto permette di identificare a colpo d'occhio le relazioni tra i vari attori e come questi interagiscano tra di loro andando a vedere se, ad esempio, ci siano dei raggruppamenti tra i nodi stessi.

Come abbiamo già avuto modo di constatare, esistono tra differenti tipologie di grafo. Dal punto di vista illustrativo, questi vengono rappresentati in maniera analoga, senza grandi differenze tra l'uno e l'altro. La tipologia del grafo, tuttavia, influisce in maniera considerevole su di un secondo tipo di visualizzazione, ovvero, quella matriciale.

2.3.1 Matrice di adiacenza

Sempre basandoci sul lavoro di [3], vediamo ora come è possibile passare da una rappresentazione grafica a quella analitica. Questa rappresentazione, che prende il nome di **matrice di adiacenza**, permette di rappresentare qualsiasi grafo finito sotto forma di una matrice quadrata dove gli indici delle righe e le colonne corrispondono ai nodi del grafo. Continuando nella riga (v_i, v_j) troveremo il valore 1 se il grafo presenta un arco che collega il nodo v_i al nodo v_j, altrimenti troveremo uno 0 qualora i nodi non dovessero essere in relazione tra loro.

Consideriamo il grafo presente in *fig.2.1(a)* costituto dall'insieme dei nodi $V = \{1, 2, 3, 4, 5\}$ e dall'insieme degli archi $E = \{(1, 2), (1, 3), (1, 4), (2, 3), (4, 5)\}$. La sua matrice di adiacenza sarà:

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Basandoci sull'esempio riportato qui sopra e dai concetti introdotti precedentemente, inoltre, si può dedurre che il vicinato del nodo 4 è dato dall'insieme dei nodi $\{1, 5\}$ e che il suo grado è 2.

Come già affermato, il valore 1 determina l'esistenza di una relazione tra v_i ed v_j. Ne consegue che la matrice di adiacenza varia sensibilmente a seconda della morfologia del grafo stesso, ovvero se i suoi archi presentano un verso. Questo dipende dal fatto che si potrebbe avere un arco per la coppia di nodi (v_i, v_j) ma, lo stesso arco, in un grafo orientato, non è percorribile in senso contrario. Riferendoci sempre alla *fig.2.1(b)*, la matrice risultante sarà:

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

In questo caso per il calcolo del del vicinato di un nodo e del su grado bisogna distinguere gli archi in entrata da quelli di usciti. Il nodo 2 avrà un *in-degree* ed un *out-degree* pari ad 1 in quanto il suo vicinato sia i entrata che in uscita e costituito da un solo nodo, rispettivamente $\{1\}$ e $\{3\}$.

Sebbene l'utilizzo della matrice di adiacenza nel campo dell'analisi sociale possa risultare dispendiosa poiché un grafo sociale, anche se finito, è caratterizzato da un

numero elevato di archi e nodi, i concetti appena introdotti risulteranno essere molto utili nel conseguimento del nostro obiettivo e troveranno un riscontro pratico più avanti nei prossimi capitoli.

2.3.2 Altre proprietà dei grafi

Per misurare correttamente un reticolo sociale e più in generale un grafo, non possiamo basarci interamente sulla natura delle relazioni tra i vari vertici. Risulta essere importante, anche la struttura stessa del grafo. A questo proposito, introdurremo una serie di nozioni che ci permetteranno di comprendere meglio il nostro lavoro.

La misurazione strutturale più semplice riguarda la **dimensione** del grafo G. Banalmente, la dimensione di un grafo è data dalla cardinalità dei suoi nodi, quindi un grafo avente n nodi, avrà dimensione N.

Questi nodi, inoltre, possono essere connessi o essere isolati. Bisogna, infatti, distinguere tra nodi che compongono la rete e nodi connessi alla rete. Questi ultimi, infatti, comprendono solo i nodi aventi almeno un arco incidente ad essi. Se il numero dei nodi connessi è minore della cardinalità dei nodi, il grafo presenterà dei nodi isolati. Questa misura viene chiamata **indice di inclusività** ed è data dalla relazione:

$$I = \frac{N_c}{N} \tag{2.3}$$

Dove N_c rappresenta i nodi connessi e N il numero dei nodi del greafo.

Un'ultima proprietà importante riguarda la densità. Questa si ricava dalla proporzione tra gli archi presenti nel grafo e quelli possibili. La densità, infatti, rappresenta il rapporto tra gli archi effettivamente presenti al tempo t e quelli potenzialmente sviluppabili. Tale formula dipende dal tipo di grafo e quindi si avrà:

- per un grafo non orientato

$$I = \frac{e_t}{\frac{n_t*(n_t-1)}{2}} \tag{2.4}$$

- per un grafo orientato

$$I = \frac{e_t}{n_t * (n_t - 1)} \tag{2.5}$$

Dove e rappresenta il numero degli archi al tempo t ed n il numero dei nodi. Questo rapporto è sempre compreso tra 0 ed 1 dove 0 rappresenta un grafo vuoto ed 1 un grafo completo.

La densità e fortemente connessa al concetto di **connettività**. Una grafo, infatti, è tanto più connesso, tanto più gli archi che lo compongono sono connessi tra loro. Questo vuol dire che ad una maggiore densità, corrisponde una maggiore connettività. Quest'ultimo concetto è molto importante per quanto riguarda le analisi delle reti sociale e troverà un approfondimento nei prossimi paragrafi.

2.4 Cenni sulla teorica dei grafi all'interno dell'analisi della SNA

Il concetto di rete sociale è piuttosto eterogeneo. Basandoci sul lavoro di [1] possiamo affermare che sebbene la sua definizione più classica sia legata al campo della sociologia che si basa, quindi, univocamente sulle relazioni umane, il concetto di rete sociale, al giorno d'oggi, trova adito anche nel campo delle telecomunicazioni, dove sia la posta elettronica che i vari client di messaggistica possono essere considerati una forma indiretta di social network in quanto vengono modellati come comunicazione tra diversi attori. Per ultimo sono sorti alcuni siti che modellano in maniera esplicita le iterazioni fra i vari utenti. Tra questi possiamo collocare Facebook, Twitter, ma anche tutti quegli applicativi di condivisione testuale e multimediale che permettono uno scambio di interazioni tra utenti come YouTube. Tutti questi siti permettono un analisi davvero ricca, data dalla varietà di contenuti che quali testo, immagini, audio, video e dati riguardanti la geolocalizzazione. È importante capire, quindi, che non solo Facebook o Twitter sono dei social network solo perchè ci vengono pubblicizati come tali, ma qualsiasi relazione o applicazione che permette un'esperienza sociale mediante interazioni con altri utenti si può considerare come

social network [1]. Sebbene il campo di applicazione dei social network sia molto ampio, il nostro campo di ricerca di estenderà a solo tre tipologie di grafi sociali ovvero quelli prodotti da Facebook, Twitter e dalle e-mail.

2.4.1 Costruzione del reticolo sociale

Ora che abbiamo circoscritto il concetto di rete sociale, passiamo a descrivere come sia possibile una sua rappresentazione.

Come evidenziato in [16], la visualizzazione di un social network deve avere l'obiettivo di identificare, in maniera chiara, gli attori di una rete e comprendere le relazioni tra gli stessi. Nello specifico, un'utente deve essere in grado di distinguere e conteggiare in maniera semplice gli attori della rete e stabilire come una persona di sua interesse sia collegata alle altre. Per questo motivo, il primo passo è costituito dal rilevamento della posizione dei vari attori all'interno della rete. Due attori, quindi due nodi del grafo, hanno la stessa posizione se questi hanno schemi di legami identici con gli altri attori anch'essi appartenenti alla medesima rete. La problematica del rilevamento delle classi di attori prende il nome di **assegnazione di ruoli** o **blockmodeling**. Partendo dagli schemi di legami tra i vari attori della rete, si potrebbero avere differenti tipi di blockmodeling. Lasciamo al lettore la possibilità di approfondire questo argomento ampiamente discusso in [3] e che esula dalle nostre competenze. Ai fini del nostro lavoro ci basterà sapere che una volta fissati uno schema di legami tra i vari attori, e identificato i ruoli comuni tra i vari nodi della nostra rete, potremo procedere a rappresentare la nostra rete. Nel capitolo 5 illustreremo nel dettaglio lo schema stabilito per rappresentare nella forma più chiara possibile i vari social network e le relazioni tra i vari elementi che li compongono.

È importante sottolineare, tuttavia, come sia erroneo pensare che la semplice applicazione di uno schema al grafo che si vuole rappresentare basti a rendere tale visualizzazione esplicativa. Lo schema dei legami stabilito a priori, infatti, deve essere funzionale agli argomenti di ricerca che vogliamo estrarre dal grafo risultante. Si pone quindi la problematica di cosa e come rappresentare nel nostro grafo. Questo perché le applicazioni di nostri interesse, Facebook, Twitter, ma anche le email,

tendono ad immagazzinare una grande quantità di informazioni e dati. Ecco che, allora, diventa importante come questi dati vengono analizzati. In [1] si evidenziano due possibili scenari di ricerca:

- **analisi basata sui collegamenti:** come suggerito anche dal nome, la ricerca verterà sull'analisi tra i vari attori all'interno della rete. Questo tipo di analisi è utile nell'identificazione all'interno delle comunità presenti nella rete o come questa sia evoluta nel tempo.
- **analisi basata sui contenuti:** in quanto, come già anticipato, i social network offrono possibilità di *mining* senza precedenti a causa dell'elevato contenuto ed eterogeneità di dati che immagazzinano in relazione alle attività di ciascuno di noi.

Nella realizzazione del nostro applicativo abbiamo sfruttato entrambe le modalità di ricerca perché, come è facile intuire, una ricerca incrociata, mescolando i due scenari appena descritti, fornisce risultati più efficaci.

Nella costruzione di un reticolo sociale, infine, è importante differenziare se si sta svolgendo un'analisi statica o dinamica della rete sociale perché cambia sensibilmente il modo in cui verrà rappresentato il grafo. Nel caso di un'analisi statica si suppone che il grafo vari lentamente nel tempo e si analizza un'istantanea del grafo ponendo, in generale, l'attenzione su particolari specifici. Al contrario, nel caso di analisi dinamica, caso naturale dei social network, le interazioni tra i vari attori mutano continuamente e ad una velocità molto elevata. A causa della grandezza di queste reti e della quantità di contenuti condivisi, un'analisi dinamica di una singola rete diventa, quindi, molto dispendiosa. Nella realizzazione del nostro lavoro, abbiamo pensato di creare una via di mezzo, effettuando un'analisi di una rete "cristallizzata", dando comunque la possibilità, in un secondo momento, di andare a modificare il grafo analizzato, qualora, nello stesso, siano avvenuti cambiamenti.

2.4.2 Visualizzazione di una rete sociale

In letteratura esistono diversi lavori che si pongono come valida alternative alla visualizzazione classica di una rete sociale. Un approccio originale lo troviamo in [16] dove si fa uso di una combinazione del diagramma di Eulero in combinazione con una mappatura ad albero. Questo tipo di approccio permette di rappresentare facilmente relazioni ed interessi comuni tra i vari attori ed organizzarli con un ordine gerarchico ed è molto valido per rappresentare in forma visiva tipi di dato testuale.

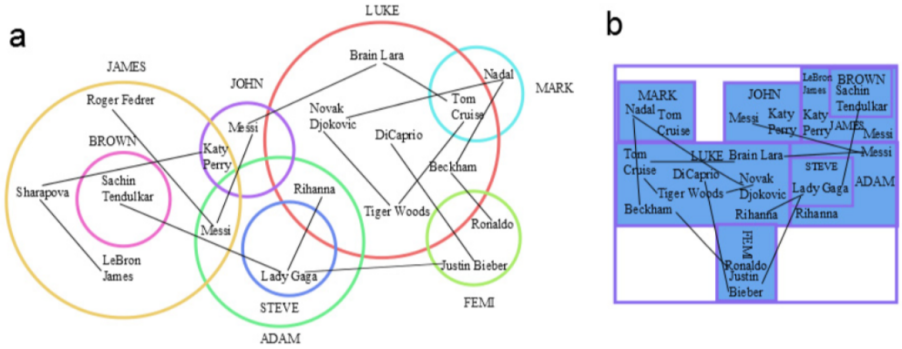
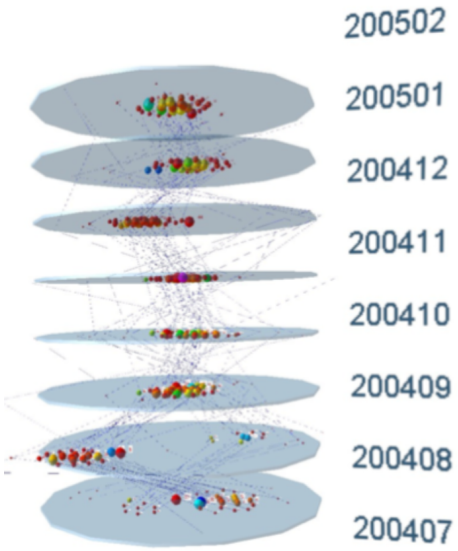


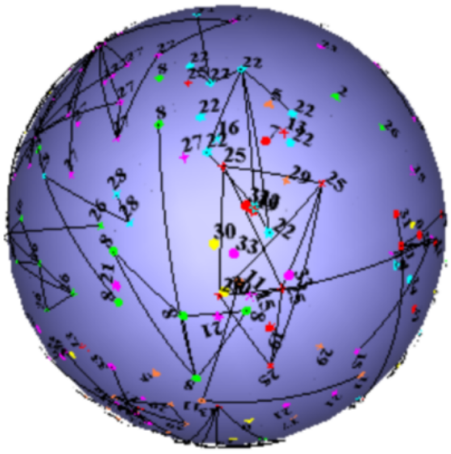
Figura 2.2: Diagramma di Eulero (a) e struttura ad albero (b)

Come si può evincere dalla *fig.2.2* i vari attori sono rappresentati mediante circonferenze e rettangoli e ciascuna figura racchiude gli interessi di ciascun attore del grafo. Gli interessi comuni sono posti nelle intersezioni delle circonferenze o vengono duplicati per ciascun attore nella visualizzazione ad albero. Per ultimo gli stessi dati di interesse correlati a ciascun attore vengono posti in relazione. Questo tipo di rappresentazione permette di fare deduzioni del tipo "Luke é amico di Jhon che ha come interesse Messi che a sua volta è in relazione con Federer".

Altro lavoro interessante è [17] dove la visualizzazione del grafo e rappresentata mediante un modello a piastre denominati 2.5D ed un modello a sfera. Il modello 2.5D viene utilizzato per rappresentare la variazione di una rete sociale nel tempo ed ogni livello rappresenta un'istantanea della rete al suo variare. Sebbene ogni livello sia indipendente dagli altri i nodi presenti su ciascun livello possono essere in relazione tra di loro e alcune volte vengono introdotti livelli intermedi per facilitare la lettura del grafo.



(a) Visualizzazione 2.5D



(b) Visualizzazione sferica

Figura 2.3: Esempi di visualizzazione di grafi

Alla visualizzazione 2.5D si affianca la visualizzazione sferica che permette una distribuzione omogenea degli attori coinvolti sulla superficie della sfera. Questa visualizzazione risulta essere molto utile per evitare che, in presenza di reti molto affollate, si abbia una concentrazione di nodi in un unico punto rendendo di fatto incomprensibile la lettura del grafo.

Possiamo citare, infine, il lavoro di [3] dove l'approccio proposto consistere nel rappresentare la rete sociale attraverso il sistema classico di nodi ed archi, ma utilizzando un duplice approccio ovvero del micro e macro layout. I nodi appartenenti al microlivello viene posto nell'area rappresentata dal nodo del macro livello, allo stesso modo gli archi del microlivello vengono raggruppati per formare un singolo arco all'interno del macrolivello. Con questo sistema il micro layout permette una focalizzazione a livello dei singoli attori e come ciascun di essi abbia instaurato le proprie relazioni, mentre il macro layout, ricavato dalla prima tipologia di visualizzazione, permette un raggruppamento dei vari attori da cui scaturisce un analisis a livello di clustering.

Ovviamente la letteratura è piena dei più svariati tentativi di rappresentazione delle reti sociali, alcuni che si rifanno alla visualizzazione classica di grafo mentre

(a) Micro grafo (b) Grafo multicircolare (c) Macro grafo

Figura 2.4: Rete organizzata tramite visualizzazione multi livello

altre che si discostando fortemente da quest'ultima, esplorando nuove possibilità. A causa dell'eterogeneità dei dati presenti all'interno degli applicativi di social networking, abbiamo deciso di adottare differenti tipologie di visualizzazione ognuna delle quali permette una differente forma di analisi del grafo rappresentato. A queste differenti possibilità di visualizzazione abbiamo affiancato anche alcune possibilità di filtrare il grafo con cui, oltre ad interagire con il grafo stesso, è possibile gestire in maniera dinamica i dati visualizzati. Quest'esigenza è stata mossa dal fatto che potersi concentrare in maniera settoriale su determinate porzioni di grafo e di rendere la visualizzazione a l'analisi stessa il più semplice possibile.

2.4.3 Argomenti di ricerca all'interno di un grafo sociale

Stabilite le modalità con le quali verrà rappresentata la nostra rete sociale, il risultato ci permetterà di analizzare in maniera accurata il grafo risultatante. In [1] sono riportati diversi scenari di analisi. Di seguito daremo una visione di quelli di nostro interesse e che hanno trovato spazio all'interno della nostra applicazione, ma che approfondiremo in maniera pratica nei prossimi capitoli.

- **Analisi statistica della rete sociale:** in questo scenario si esamina il grado di connessione dei nodi. Si cerca di individuare i nodi con poche connessioni, isolati, se vi sono nodi che fungono da "hub" o se il grado di connessione risulta essere più uniforme.

- **Rilevamento delle comunità nella rete sociale:** il problema del *clustering* riguarda l'individuazione delle regioni della rete che sono più densamente connesse in termini di interazione tra i vari nodi.

- **Classificazione dei nodi:** a causa dell'eterogeneità dei dati, risulta utile attribuire delle etichette ai vari nodi. In questo scenario è possibile individuare e analizzare la distribuzione di queste etichette.

- **Visualizzazione:** poiché i social network diventano sempre più grandi e complessi, ragionare sulle dinamiche sociali attraverso semplici statistiche è complicato e non molto intuitivo. La visualizzazione fornisce un modo naturale per riassumere le informazioni in modo da renderle molto più comprensibili.

- **Mining dei contenuti:** l'estrazione di dati è centrale nell'ambito dell'analisi dei social network, soprattutto in campo forense. A causa della massiccia quantità di dati in essi presente è possibile estrarre qualsiasi informazione che spazia dai contenuti testuali a quelli multimediali.

- **Tagging sociale e posizionale:** anche l'analisi del tagging diventa importante all'interno della nostra analisi poichè Gran parte dell'interazione tra utenti e social network avviene sotto forma di questa meccanica. Gli utenti non solo allegano brevi descrizioni ai vari contenuti condivisi all'interno del social network, ma possono anche allegare dati riguardante la localizzazione del contenuto condiviso

Capitolo 3

Social Network Forensic

Nel precedente capitolo abbiamo affrontato cosa sia e cosa riguarda un'analisi sociale, illustrando il *modus operandi* per arrivare a costruire un grafo che possa essere comprensibile da parte dell'utente finale. Fin'ora, abbiamo solo accennato al fatto che il nostro lavoro dovesse avere un riscontro anche in ambito legale per quanto concerne le analisi forensi. Il nostro obiettivo ultimo è, infatti, quello di costruire un applicativo che, oltre ad avere le funzionalità precedentemente descritte, possa anche essere di supporto per attività legali e forensi. L'analisi dei contenuti digitali è essenziale nelle indagini giudiziarie e nei procedimenti penali per la ricerca di informazioni probatorie, questo perché, sempre più spesso, i mezzi telematici costituiscono una fonte senza precedenti per quanto riguarda la grande quantità di informazioni che posso essere recuperate per rilevare intenti criminali. In questo capitolo andremo ad illustrare le meccaniche dell'analisi sociale in ambito legale evidenziando quali siano le metodologie ed i contenuti di maggiore spessore utili ai fini di un'azione giudiziaria.

3.1 Introduzione all'analisi forense

Le forme di comunicazione telematica, prima con le e-mail, ma ancora di più con l'avvento delle applicazioni sociali, sono diventate i principali vettori di contenuti dannosi all'interno della rete. Con questo, non ci riferiamo solamente a spam o

virus, ma anche, ed in maniera particolare, a tutta un a serie di contenuti testuali che vengono scambiati tra malintenzionati con intenti criminali. Per questo motivo è diventato pressoché indispensabile avere degli applicativi che abbiano la capacità di analizzare tali contenuti in modo da poter identificare possibili attività fraudolente. Questo fenomeno prende il nome di **Digital Forensics** e si riferisce al processo di acquisizione, elaborazione e conservazione di dati utili ai fini giudiziari o altre questioni civili [6]. A causa della complessità e della delicatezza di queste attività di indagine, ecco quindi che la scelta di strumenti adeguati per dare supporto ad un'indagine diventa quindi una problematica centrale.

All'interno dell'attività investigativa, si possono identificare due aspetti chiave: il primo consiste nell'acquisizione delle informazioni, mentre il secondo si focalizza sull'analisi dei dati estratti [5]. Accade che questa attività di analisi venga eseguita attraverso un complicato e costoso processo manuale e, a causa dell'elevato volume dei dati, possa essere soggetta ad errori. Emerge, quindi, l'esigenza di avere applicativi funzionali che possano svolgere in maniere automatica questo tipo di analisi.

3.1.1 Acquisizione dei dati ai fini legali

L'acquisizione di dati è il primo problema nel momento in cui si deve affrontare un'analisi forense, poiché la quantità e la qualità dei dati recuperati va ad influenzare l'analisi.

I dati appartenenti ai servizi di social netowrking e clouding, infatti, vengono raccolti con maggiore difficoltà rispetto a quanto non avvenga per i dati memorizzati sui calcolatori. Mentre la *digital forensics* tradizionale si basa, ad esempio, sull'acquisizione fisica dell'hardware e all'utilizzo di hash per garantire l'integrità delle prove, a causa della mancanza di API standardizzate nel campo della digital forensic, è possibile fare affidamento solo su soluzione isolate, specifiche per l'insieme di dati che si sta andando ad analizzare in quel determinato momento. Un altro fattore che influisce sull'acquisizione dei dati risiede nella collaborazione dell'operatore dell'applicazione. Quest'ultimo è tenuto a fornire dati esclusivamente nel momento in cui

ci siano procedimenti penali a carico di un individuo. È possibile prendere a riferimento, in questo senso, le linee guida per le forze dell'ordine per quanto riguarda due dei social network di maggiore utilizzo nella vita quotidiana ovvero Facebook e Twitter [9, 10].

Social network	**Ingiunzione legale**	**Dati acquisiti**
Facebook	Inchiesta penale	Informazioni di base: nome, indirizzi email, indirizzi IP, coordinate bancarie.
	Ordinanza del tribunale	Dati essenziali sugli utenti, intestazioni di messaggi e indirizzi IP. Esclusi i contenuti delle comunicazioni.
	Mandato di perquisizione	Messaggi, foto, video, post sul diario e informazioni sulla posizione.
Twitter	Ordine di comparizione Mandato di perquisizione	Informazioni non di pubblico dominio. Contenuti delle comunicazioni: tweet, messaggi privati, foto.

Tabella 3.1: Facebook and Twitter Law Enforcement Guidelines

Appare ovvio, quindi, come l'investigatore possa essere rallentato dall'ente proprietario dell'applicazione nella raccolta delle prove. A questo, infatti, bisogna fornire motivazioni legali valide affinché possa rilasciare informazioni riguardanti possibili indiziati. Nel nostro lavoro, ci siamo preoccupati di fornire una soluzione che non prevedesse alcuna interazione con la società fruitrice del servizio in modo tale da poter agire in più completa autonomia in sede di analisi forensi. Questo, però, non vuol comunque dire che qualsiasi insieme di dati possa essere usato ai fini di un'indagine. La raccolta delle informazioni deve essere comunque mossa da motivazioni penali valide affinché possa avere valore in sede legale.

3.1.2 Fonti di dati e loro interpretazioni

Sebbene le reti sociali siano differenti nella loro struttura e nelle loro caratteristiche, in base alla fonti di dati generici sotto esame, chiamati anche **social netowork data pools**, è possibile ricavare specifici tipi di informazioni [13] indipendentemente dall'applicazione da cui i dati provengono.

- **Social footprint**: indica la possibilità di dedurre cosa rappresenti il grafo dell'utente, ad esempio quali siano i suoi amici.
- **Schema di comunicazione**: possibilità di dedurre come venga utilizzata la rete per comunicare e quali siano, ad esempio, le persone con cui l'utente comunica di più.
- **Tempi di attività**: possibilità di stabilire quali siano i tempi in cui l'utente è più connesso al social network o quando sia stata svolta un'attività specifica.
- **Apps**: stabilire quali siano le applicazioni con cui l'utente interagisce e quale scopo.
- **Multimedia**: quali sono i contenuti multimediali caricati dall'utente o con quali altre persone è stato taggato.

Ciascuno di questi insieme di informazioni può essere molto significativo ai fini di un'indagine legale, ma può essere dedotto solo se le informazioni recuperate sono complete. La nostra applicazione cercherà di assolvere in questo senso fornendo quattro schemi di rappresentazione utili ai fini deduttivi.

3.2 Visualizzazione di un grafo in ambito della Social Network Forensic

L'analisi di un social netowrk in ambito forense passa inevitabilmente per una rappresentazione grafica dei dati analizzati. Una corretta visualizzazione dei dati in esame può essere determinante nell'individuazione di prove determinanti ai fini legali,

mentre l'utilizzo incrociato di più tecniche di visualizzazione permette un'analisi più ramificata. Il lavoro di [13] offre un importante spunto su quelle che possono essere le rappresentazioni più esplicative utili ai fini di analisi forensi e di cui presentiamo di seguito una panoramica.

3.2.1 Grafo di interconnessione sociale

Le informazioni per rappresentare un grafo di interconnessione sociale sono facilmente recuperabili perchè, nella maggior parte dei social network, questo tipo di informazioni sono di pubblico dominio. Sebbene potrebbe essere molto facile recuperare la lista di amicizie da Facebook, quella dei follower per Twitter o la lista dei destinatari per un'indirizzo di posta elettronica, non è banale mettere in pratica questa visualizazzione affinché possa mostrare informazioni effettivamente utili come ad esempio quali amici della lista sia entrati in relazione tra loro o come questi siano raggruppati. Questa visualizzazione produce un grafo $G = (V, E)$ dove $V = \{v_i; i = 1, ..., n\}$ è la lista degli amici, mentre $E = \{(v_i, v_j), ...\}$ rappresenta una relazione che collega due attori del nostro dataset.

3.2.2 Grafo di interazione sociale

Ai fini di un'indagine è importante capire come un utente comunichi con gli altri attori della rete. Queste comunicazioni possono includere sia messaggi trasmessi in maniera pubblica, ma anche quelli trasmessi in maniera privata. Il grafo risultante si dovrà adattare per fornire in maniera chiara informazioni attinenti a questo contesto. L'esempio che troviamo in [13] parla di un grafo dove l'insieme V rappresenta sempre gli attori del dataset, diversamente l'insieme E rappresenta un arco pesato dove il valore viene incrementato ogni volta che un messaggio viene inviato da v_i a v_j .

3.2.3 Visualizzazione tramite georeferenziazione dei contenuti

I social netowork, previo nostro consenso, acquisiscono in continuazione informazioni sulla nostra posizione. Alcune volte le informazioni di geolocalizzazione vengono condivise in maniera pubblica, altre volte rimangono immagazzinati all'interno del sistema. Un buon sistema di analisi forense deve essere in grado di acquisire, elaborare e visualizzare questi dati in modo tale da poter permettere agli investigatori di individuare quali siano i luoghi di interesse dell'utente in analisi.

Bisogna considerare come, durante questa visualizzazione, possano sorgere delle problematiche dovute all'associare un termine ad un contesto geografico. Queste parole, infatti, spesso sono ambigue e, se decontestualizzate, potrebbero avere un significato fuorviante. Il termine *nice*, ad esempio, nella lingua inglese, può essere usato sia come aggettivo sia come nome proprio della città *Nizza*. Ancora, quando utilizziamo il termine *London* siamo soliti riferirci alla capitale del Regno Unito senza considerare che vi sono città omonime in Canada e negli Stati Uniti [23]. Per questo motivo, l'elaborazione dei dati riguardanti la posizione deve essere particolarmente meticolosa, altrimenti un investigatore potrebbe basarsi su dati completamente sbagliati.

3.2.4 Timeline

Un'altra tipologia di visualizzazione menzionata in [13] è la così detta timeline. Grazie soprattutto agli smartphone, abbiamo modo di rimanere sempre connessi con la nostra rete sociale. All'interno di questa rappresentazione si potrebbero visualizzare, quindi, i tempi di attività dell'utente, dando modo ad un investigatore di capire quali con quale frequenza l'utente utilizzi il social network. In questo contesto potrebbe risultare utile una funzione che effettua zoom in o zoom out per concentrarsi su un range di tempo selezionabile a causa del fatto che gli utenti effettuano diverse decine di attività sociali ogni giorno.

3.2.5 Altre visualizzazione per le analisi forensi

Oltre ai quattro tipi di visualizzazioni definiti basilari, è possibile ricavare altri tipi di visualizzazioni che esplorano il grafo sociale in maniera più avanzata.

- **Monitoraggio degli eventi**: indagini sulla diffusione di contenuti dannosi all'interno di una rete richiedono l'identificazione di chi o cosa abbia innescato l'evento. Per questo motivo, avere una raccolta di impronte degli utenti di un social può fornire interessanti informazioni riguardanti la diffusioni di questi contenuti dannosi.

- **Timeline matching**: acquisire dati da un sistema fortemente centralizzato come quello di un social network permette agli investigatori di potersi basare sui *timestamp* forniti dal social stesso. Gli operatori dei social network, infatti, tendono a mantenere gli orologi coordinati su migliaia di server e questo permette una misurazione temporale univoca. Ne consegue che questi timestamp possono essere utilizzati per confrontare le timeline di due utenti differenti o addirittura un gruppo più ambio o creare una macro timeline a partire dalle singole di ciascun utenti.

- **Istantanee differenziali**: poiché il grafo sociale di un utente varia fortemente nel tempo, la sua fotografia potrebbe essere cambiata sensibilmente dal momento in cui viene catturata a quando viene analizzata. Per questo motivo un investigatore potrebbe trovare utile confrontare diversi grafi dello stesso utente per capirne l'evoluzione.

3.3 Estrazione delle informazioni testuali

La parte più significativa di un'analisi nell'ambito della social network forensics riguarda, in larga parte, l'estrazione e l'analisi dei contenuti testuali. Questo perché la loro struttura non è definita da regole precisa, ma varia a seconda di diversi fattori come, ad esempio, le capacità linguistiche della persona che scrive o di coloro

a cui il testo è rivolto. Il metodo analitico che si occupa di cercare argomenti o il significato di parole e frasi all'interno di questi documenti prende il nome di **linguistica forense** [18]. Il risultato delle analisi prodotte dalla linguistica forense sarà utile, in una fase successiva, per assolvere ad altre problematiche quali:

- l'identificazione dei contenuti o frammenti testuali aventi rilevanza penale
- la scoperta di relazioni tra i vari utenti
- il rilevamento di semantiche nascoste

All'interno del nostro lavoro abbiamo deciso di articolare il processo estrattivo in tre fasi:

1. **Identificazione delle parole non rilevanti**: bisogna escludere dalla fase di elaborazione tutte quei termini che vengono definite **stop words**. Usiamo questa definizione per indicare tutte quelle parole che, a causa della loro frequenza elevata all'interno delle lingua, sono ritenute poco significative ai fini della ricerca. Si possono considerare stop words, ad esempio, tutte le congiunzioni, preposizioni o i pronomi personali.

2. **Pre-elaborazione dei contenuti testuali all'interno del dataset**: in questa fase ci siamo basati su quanto descritto in [5], ovvero un sistema che elabora i testi digitali sotto forma vettoriale. L'insieme di documenti di interesse $D = \{D_1, ...D_n\}$, chiamato anche **corpus**, passa attraverso un processo di **tokenizzazione**. I documenti vengono, infatti, ridotti ad una sequenza di **token**, ovvero dei termini limitati da spazi, e rappresentati come un vettore che si trova all'interno dello spazio di un vocabolario $T = \{t_j; j = 1, ..., n_T\}$. Questo dizionario è, quindi, un insieme che può essere assemblato raccogliendo tutti i termini che occorrono almeno una volta all'interno della raccolta di tutti i documenti D_i e prende il nome di **bag of words**. È importante sottolineare come non tutte le parole siano significative ai fini del processo di analisi per cui da una bag of word risulta utile eliminare le stop words.

3. **Individuazione delle parole più significative all'interno del dataset di riferimento**: filtrato il corpus dalle stop words, si passa all'individuazione dei concetti veramente significativi. Questa fase consiste nell'associare un peso ad ogni parola w con $w \in D_i$ al fine di misurare l'importanza di un termine in un documento. Per fare questo abbiamo utilizzato uno strumento di largo utilizzo all'interno delle analisi del linguaggio naturale ovvero la funzione di peso chiamata **term frequency–inverse document frequency** o più brevemente **TF-IDF**. L'obiettivo di questa misurazione è quello di dare una maggiore importanza ai termini che compaiono poche volte all'interno del *corpus*. Tramite questa funzione si calcola quindi, la frequenza relativa delle parole in un documento specifico rispetto alla proporzione inversa di quella parola rispetto all'intero *corpus*. Questo metodo determinerà la rilevanza delle parole all'interno dei documenti, quindi le parole che tendono ad apparire in un piccolo insieme di documenti avranno un valore TFIDF più alto [15]. In maniera più formale, possiamo definire la funzione di peso **TF-IDF** come:

$$TFIDF_{(i,j)} = tf_{(i,j)} \cdot idf_{(i,j)} \tag{3.1}$$

dove $tf_{(i,j)}$ indica la frequenza numerica della parola w_j nel documento D_i, mentre con $idf_{(i,j)} = \log \frac{N}{df_i}$ indichiamo la frequenza inversa, dove con N si indica il numero totale di documenti, mentre il denominatore indica il numero di documenti dove è presente w [7]. Definito questo scenario, si può facilmente intuire perché è utile escludere le stop words nella fase di preelaborazione. Queste, infatti, non otterrebbero un punteggio elevato proprio a causa della loro frequenza ricorrente all'interno del *corpus* e quindi non sarebbero di nessuna utilità ai fini dell'analisi.

Capitolo 4

Caratteristiche degli applicativi attualmente disponibili

Nella stesura del nostro lavoro, ci siamo soffermarti sull'analizzare quelle che sono gli applicativi attualmente presenti sul mercato. Per fare questo, abbiamo preso in considerazione diversi fattori, evidenziandone pregi e difetti di ciascuno. Il nostro obiettivo è, infatti, quello di creare un applicativo che oltre ad essere di supporto per le analisi forensi, abbia anche l'ambizione di essere di utilizzo generico e non circoscritto ad uno specifico dataset e quindi ad uno specifico social network. In questo capitolo andremo ad illustrare quali sono i criteri di paragone e quali differenze possiamo trovare in questi applicativi

4.1 Criteri di confronto

L'analisi sociale e l'analisi forense offrono da sempre una grande attrattiva ed per questo motivo che nel corso del tempo sono stati diversi gli applicativi il cui bacino di utilizzo varia a seconda di diverse caratteristiche come descritto in [6]. Ecco quindi che la scelta dell'applicativo è determinante per poter svolgere al meglio l'attività di analisi. Di seguito andiamo a presentare i alcuni elementi che permettono una prima categorizzazione di questi programmi:

- **necessità del file sul HDD**: implica la necessità da parte del tool che i dati da analizzare siano memorizzati fisicamente sul disco rigido
- **opzione di ricerca**: opzioni di filtraggio che l'utente può applicare come ricerca di una parola chiave o differenziare l'importanza dei dati da visualizzare
- **informazioni estratte e capacità di recupero**: quali informazioni siamo in grado di recuperare in seguito all'analisi come un grafo sociale o dati testuali
- **formato di file supportato**: il formato dei file accettati in input dall'applicazione
- **supporto per la visualizzazione**: modalità di visualizzazione del grafo e dei dati in seguito all'elaborazione dei dati.
- **sistema operativo supportato**: non tutti gli applicativi possono essere lanciati sugli stessi sistemi operativi a causa della loro estensione
- **licenza d'uso**: se l'applicativo è un software proprietario o rilasciato con licenza open-source
- **formato di esportazione supportato**: possibilità di esportare i dati analizzati e formato dell'esportazione

Altro fattore determinante che incide sulla scelta di utilizzo di un applicativo all'interno dell'attività di analisi è la visualizzazione dei dati che questo è in grado di fornire. La visualizzazione fornita da un certo applicativo potrebbe essere più adatta rispetto ad altri per una specifica attività di analisi. Quest'ultima viene, infatti, influenzata in maniera inevitabile dalla visualizzazione utilizzata dall'applicazione poiché la profondità di analisi dipenderà, in buona parte, dalla correttezza dei dati visualizzati e dalla congruenza rispetto all'obiettivo di ricerca. La letteratura propone diverse tecniche che permettono la rappresentazione di una rete sociale le principali possibilità di visualizzazione di un grafo sociale:

- **rappresentazione classica**: tipica visualizzazione con cui si è soliti rappresentare un grafo sociale, ovvero gli attori mediante nodi e le relazioni tra gli stessi medianti archi

- **grafico a barre**: usati di solito per raggruppare dati e rappresentarne statistiche

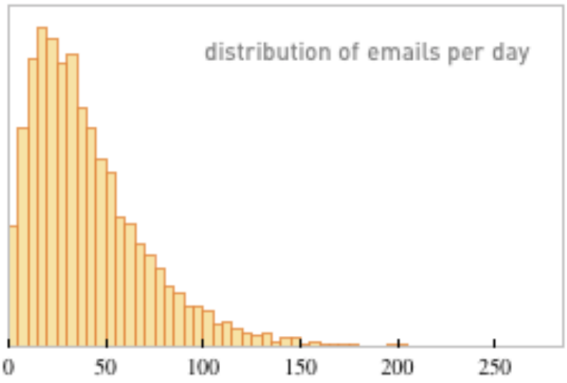


Figura 4.1: Diagramma a barre nella SNA [20]

- **mappa geografica**: utile qualora si volessero rappresentare contenuti su di una mappa geografica

- **rappresentazioni di cluster**: gruppi o cluster vengono solitamente rappresentati con un ordine gerarchico di solito tramite un ordinamento a torta su più livelli

- **rappresentazioni in tempo reale**: permette di visualizzare in tempo reale eventuali modifiche della rete stessa

- **rappresentazioni su timeline**: utile per visualizzare o posizionare contenuti su di una linea temporale

Figura 4.2: Rappresentazione su di una timeline [20]

4.2 Comparazione di software per la visualizzazione di SN

Una buona comparazione dei diversi applicativi attualmente disponibili la possiamo trovare all'interno di [2]. Qui possiamo trovare diversi strumenti e librerie open-source con una licenza che permette l'utilizzo a fini commerciali che permettono la rappresentazione delle rti sociali. Di seguito proponiamo una panoramica:

Nome	**Ambiente**	**Caratteristiche**
Cuttlefish	· Linux	· Visualizzazione dettagliata della rete · Manipolazione interattiva del layout ed editing dei grafo · Formati di dati molteplici per l'input e l'output
Gephi	· Linux · Mac OSX · Windows	· Capacità di gestire grafi fino ad un milione di nodi · Supporto a grafi diretti, indiretti e misti · Supporto a differenti tipi di misurazione · Filtraggio dinamico in base alle caratteristiche del grafo

Nome	**Ambiente**	**Caratteristiche**
Graph-tool	· Python	· Analisi statistica del grafo · Utilizzo di differenti algoritmi che permettono un livello di performance elevate · Algoritmi di layout basati su GTK+ · Possibilità di integrazione con GraphViz
GraphViz	· Linux · Mac OSX · Windows	· Possibilità di elaborare i dati di output in immagini SVG, PDF o come visualizzazione classica del grafo · Modificazione del layout con colori e font
MeerKat	· Linux · Mac OSX · Windows	· Visualizzazione di layout multipli · Editing interattivo · Calcolo della centralità della rete · Rilevamento automatico dei clustering
NetworkX	· Python	· I nodi del grafo possono essere qualsiasi cosa (testi, immagini, record XML) · I bordi dei nodi possono rappresentare altri dati correlati al nodo stesso
Vis.js	· Framework javascript	· Visualizzazione · Possibilità di visualizzazione in diversi modi (Graph2D, Graph3D, Timeline, Network) · possibilità di gestire grandi quantità di dati

Nome	**Ambiente**	**Caratteristiche**
SocNetV	· Linux · Mac OSX · Windows	· possibilità di applicare layout in base a proprietà sociali o matematiche · Calcolo delle proprietà di base del grafo (densità, diametro, connettività, etc.) · Differenti algoritmi per il layout
NodeXL	· Linux · Mac OSX · Windows	· importazione ed esportazione dei grafi nei formati di GraphML, Pajek, UCINet e di matrice · Zoom e scaling del grafo · Utilizzo di differenti tipologie di layout
SubDue	· Linux	· Rappresentazione dei dati utilizzando un grafico diretto con label · Apprendimento basato su grafici dai dati di input · Ultima release risale al 2011

Tabella 4.1: Tabella di applicazioni open-source per la visualizzazione di SN

Ai software appena descritti, possiamo affiancare strumenti specializzati per quanto riguarda l'analisi forense. Questi, infatti, oltre ad accettare tipi di dato eterogenei come input, permettono ricerche più selettive accompagnate talvolta da strumenti per l'estrazione testuale e quindi trovano maggiore impiego in settori investigativi:

- **Intella**: fortmente utilizzato a causa della sua potente capacità di analisi ed estrazione testuale, permette opzioni di filtraggio su diversi elementi quali e-mail, documenti e altri file di testo. Permette di combinare diversi risultati in una visualizzazione che raggruppa i risultati relativi alle ricerche per parole chiave e cliccando sui diversi cluster si ha la possibilità di visualizzare le risorse

correlate recuperando i documenti originali. L'estrazione testuale può essere fatta mediante l'utilizzo di espressioni regolari, caratteristica che fa di Intella uno degli strumenti più interessanti per quanto riguardo l'estrazione testuale [21].

- **Xplico**: permette la ricostruzione dei dati applicativi acquisiti mediante *packet sniffing* con strumenti come Wireshark. Nello specifico, permette di ricomporre dati applicativi indipendemente dal protocollo utilizzato. Ad esempio Xplico è in grado di ricostruire le mail scambiate indipendemente dal fatto che mittente e destinatario utilizzino protocolli POP, SMTP o IMAP. L'interfaccia grafica è un'interfaccia web e i suoi database backend possono essere indifferentemente SQLite, MySQL o PostgreSQL [22].

- **Paraben**: è un aggregato di software differenti per la digital foresics e analisi dei rischi in ambito di sicurezza digitale. Permette l'acquisizione di dati proveniente da computer, smartphone, email e servizi di cloud e rappresentarli attraverso un'interfaccia comune.

- **Email Tracker Pro**: offre la possibilità di rintracciare un'e-mail utilizzando il suo header, ma è anche dotato di un filtro antispam che analizza ogni e-mail in ingresso e avvisa l'utente se si tratta di sospetto spam [19].

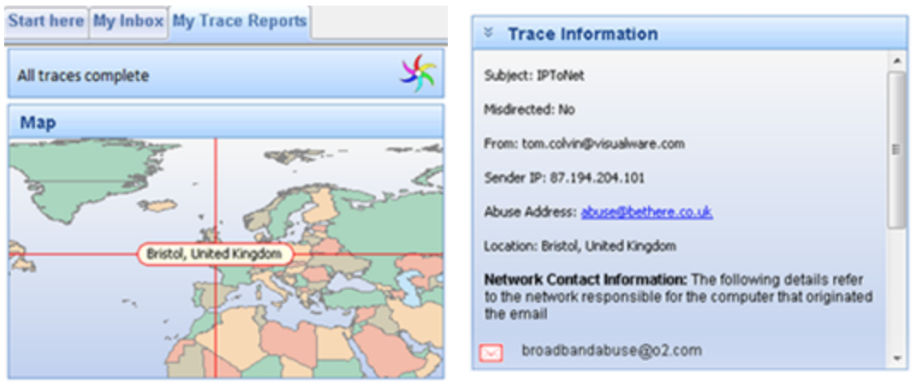
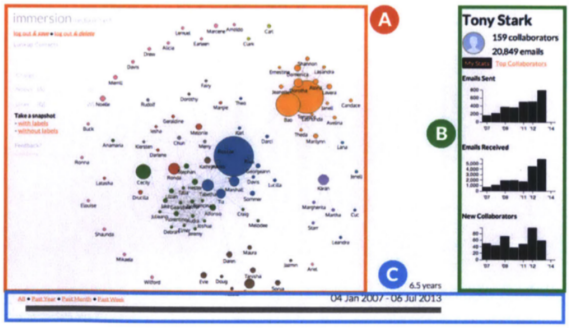
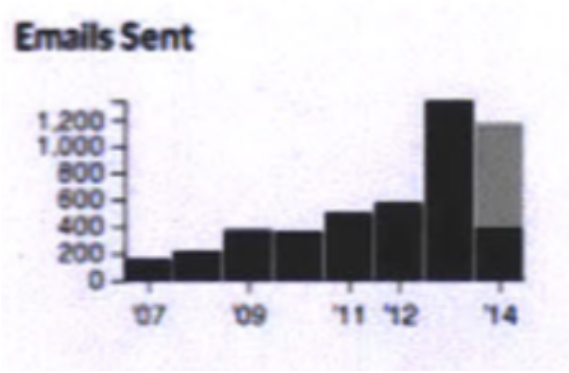


Figura 4.3: Esempi di visualizzazione per Email Traker Pro [19]

- **Immersion**: l'ultimo tool che siamo andati ad analizzare è quello che presenta la visualizzazione più intuitiva tra tutti, ovvero la classica basata su nodi ed archi a cui si accompagna una visualizzazione a barre per l'analisi statistica dei dati. Permette anche di filtrare l'intervallo temporale. In Immersion, ciascun attore è rappresentato come nodo circolare, e la dimensione del nodo corrisponde al numero di email che l'utente ha scambiato con altri attori ad esso collegati. Allo stesso modo la larghezza del link è modulata anche in base alla forza del rapporto tra mittente e destinatario. La vista a barre si differenzia in tre istogrammi dove il primo mostra come il numero di email inviate dall'utente si è evoluto nel corso degli anni, il secondo mostra una simile statistica, ma per il numero di email ricevute dall'utente nel corso degli anni, mentre il terzo mostra il numero di nuove persone a cui l'utente ha inviato email ogni anno.



(a) Vista classica



(b) Vista a barre

Figura 4.4: Esempi di visualizzazione per Immersion

Capitolo 5

Analisi dei requisti e progettazione del sistema

Nella realizzazione del nostro applicativo abbiamo tenuto conto dei diversi aspetti fin qui illustrati andando ad implementare una serie di funzionalità che permettessero di svolgere l'analisi nel modo più semplice possibile. L'obiettivo non è solo quello di ottenere un software che avesse una grande immediatezza visiva, ma che permettesse anche di rappresentare, con differenti metodologie, i diversi insieme di dati estraibili dai dataset di partenza. Come abbiamo avuto modo di constatare nel corso del nostro lavoro, infatti, ad oggi poche applicazioni permettono un'analisi trasversale dei social network e sono piuttosto rigidi riguardo le reti sociali che sono in grado di analizzare. Se a questo, aggiungiamo il fatto che ormai una singola persona tende ad avere più applicazioni di social networking con cui creare differenti tipologie di reti sociali ne consegue che è sicuramente utile avere un applicativo che permetta una centralizzazione dell'analisi in modo tale da permette un confronto tra reti differenti appartenenti allo stesso utente. La nostra applicazione permette, infatti, l'elaborazione e l'analisi dei dati proveniente da Facebook, Twitter e mail box.

In questo capitolo andremo ad illustrare tutte le fasi per giungere all'obbiettivo finale, partendo dalla fase di acquisizione dei dati fino a giungere a quella di rappresentazione, passando per la fase elaborativa ed estrattiva. Per ultimo ne illustreremo alcuni casi d'uso mediante l'utilizzo di alcuni dataset.

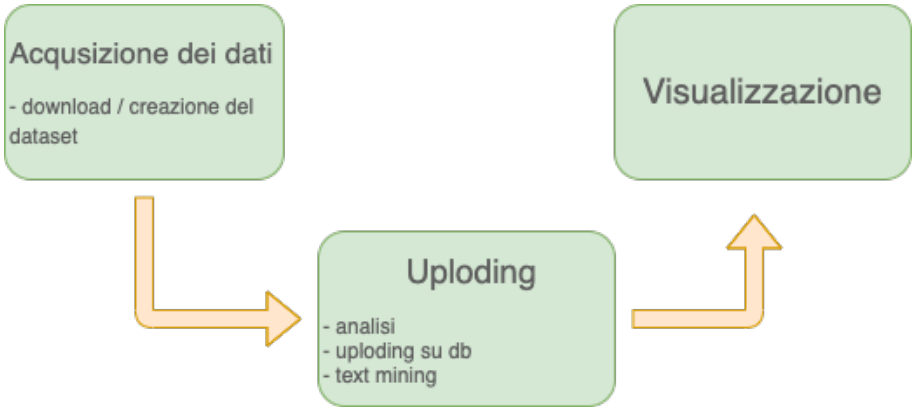


Figura 5.1: Pipeline del nostro applicativo per la SNA

5.1 Acquisizione dei dati di input

All'interno del nostro lavoro abbiamo già discusso riguardo le difficoltà che un investigatore potrebbe incontrare nell'acquisizione dei dati. Per ovviare a questa problematica e snellire il processo di acquisizione abbiamo pensato ad una soluzione che non prevedesse alcun tipo di interazione con l'ente fruitore del servizio. Da ormai qualche tempo, tutte le applicazioni di social networking permettono l'esportazione ed il download di tutti i dati caricati dall'utente. Questi dati comprendono tutte le informazioni riguardanti l'utente e la relativa attività sociale svolta dal momento dell'iscrizione all'applicazione sino al giorno di esportazione dei dati stessi. Grazie a questa quantità di dati, è possibile, quindi, tracciare in maniera molto precisa la rete sociale dell'utente in analisi.

Per quanto riguarda le email, i vari client di posta elettronica permettono l'esportazione della propria casella email in formati differenti tra loro. Questo costituisce un problema in quanto vorremmo avere un formato comune per i dati di input accettati dalla nostra applicazione. Per questo motivo, basandoci sul lavoro di [8], abbiamo adottato il formato **MBOX** come formato comune per la formattazione dei dataset per quanto riguarda le email. All'interno dei file con estensione *.mbox le varie email vengono concatenate insieme e ciascuna ha come delimitatore superiore una stringa che inzia con la parola "From" e come delimitatore inferiore uno o più spazi.

```
From MAILER-DAEMON Fri Jul  8 12:08:34 2011
From: Author <author@example.com>
To: Recipient <recipient@example.com>
Subject: Sample message 1

This is the body.
>From (should be escaped).
There are 3 lines.

From MAILER-DAEMON Fri Jul  8 12:08:34 2011
From: Author <author@example.com>
To: Recipient <recipient@example.com>
Subject: Sample message 2

This is the second body.
```

Figura 5.2: Esempio di un file MBOX

Differentemente per Facebook e Twitter, i dati estratti vengono esportati in maniera strutturata, ovvero in **JSON**, un formato molto comodo per lo scambio dei dati molto semplice da poter gestire in quanto è costituito da una sequenza non ordinate di coppie nome-valore ciascuna separata dal carattere ":" [11].

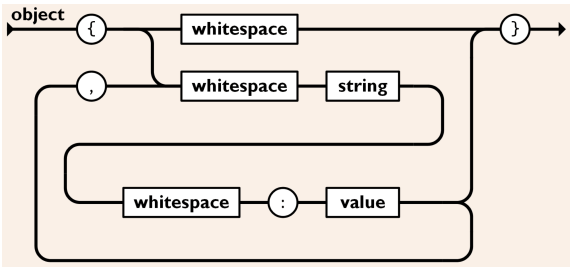


Figura 5.3: Struttura generale di un oggetto JSON [11]

Questo formato grezzo si presta molto bene per essere oggetto di elaborazione in quanto per ottenere un determinato valore, basterà accedervi tramite la sua chiave.

5.1.1 Acquisizione dei dati per Facebook e Twitter

I clienti di e-mail utilizzano formati differenti per l'esportazione dei loro dati per cui la generazione del file MBOX varia da client a client, al contrario Facebook e Twitter oltre ad esportare i vari dataset con un formato comune, hanno delle metodologie

molto simili di esportazione che consiste nella creazione e download del dump dei propri contenuti presenti all'interno del social.

Per scaricare il dump di un profilo Facebook di un utente bisogna seguire il percorso "Account → Impostazioni e privacy → Impostazioni" e cliccare, dal menù appena raggiunta, su "Le tue informazioni su Facebook".

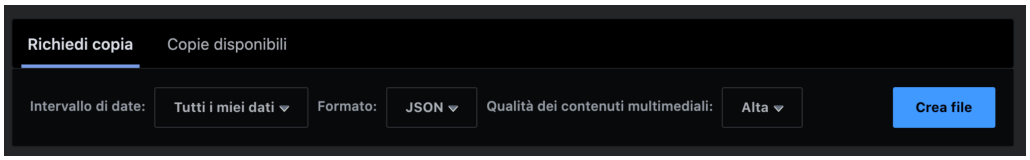


Figura 5.4: Schermata di download per Facebook

Da questa schermata si ha la possibilità di modificare alcune opzioni per la creazione del file o, qualora si fossero già creati precedentemente dei dump, di riscaricarli.

Intervallo di date	Permette di selezionare un intervallo temporale. Il dump risultante conterrà solo i dati all'interno del range selezionato. Questo campo di default non è impostato per cui l'intervallo predefinito va dalla data di iscrizione dell'utente alla data corrente.
Formato	Possibilità di esportare i file del dump in formato *.html o *.json
Qualità dei contenuti multimediali	Permette di modificare la qualità di foto e video che l'utente ha caricato su Facebook nel corso del tempo. Banalmente ad una maggiore qualità dei contenuti corrisponde una maggiore dimensione del file di dump.

Tabella 5.1: Opzioni di creazione per il dump di Facebook

Twitter offre un sistema di download del dump molto simile. Per ottenere il file bisogna selezionare dal menù laterale la voce "Altro → Impostazioni e privacy →

Il tuo account → Scarica l'archivio dei tuoi dati", fino ad arrivare alla schermata sottostante. Twitter, tuttavia, non permette di personalizzare il dump e quindi il file risultante andrà dalla registrazione dell'utente alla data in cui si è richiesto il file.

Figura 5.5: Schermata di download per Facebook

In ambedue i casi i file che si ottengono da queste fasi è un pacchetto compresso *.zip contente una serie di cartelle dove ciascuna racchiude uno o più file contenenti oggetti JSON. I file compressi, contenenti i dump dei due social network, insieme al file con estensione *.mbox, saranno i file caricabili all'interno del nostro applicativo e che in una fase successiva, si occuperà di decomprimerli, analizzarli e caricare sul database i vari dati processati.

5.2 Struttura del server

Come abbiamo avuto modo di vedere nella sezione precedente, il nostro lavoro vuole dare la possibilità all'utente, di poter caricare all'interno dell'applicazione dataset sempre differenti. Questa, però, non è l'unica difficoltà che abbiamo dovuto risolvere in quanto la maggiore problematica derivava dal voler creare un applicativo che potesse gestire in maniera rapida ed efficiente un'elevata quantità di dati come può

essere quella relativa ad un grafo sociale. Per questo motivo abbiamo pensato di creare un sistema che si basasse sulla comunicazione tra client e server. In questo modello tutto lo sforzo di analisi, elaborazione e mining viene gestito dal server al momento dell'uploading del dump dei dati, mentre l'unico compito del client è quello di rappresentate graficamente i dati richiesti. In particolare, una volta che i dati sono stati caricati sul server, questo si preoccuperà di scompattarli, analizzarli e caricarli su **Neo4j**, un database a grafo, vero cuore del nostro applicativo.

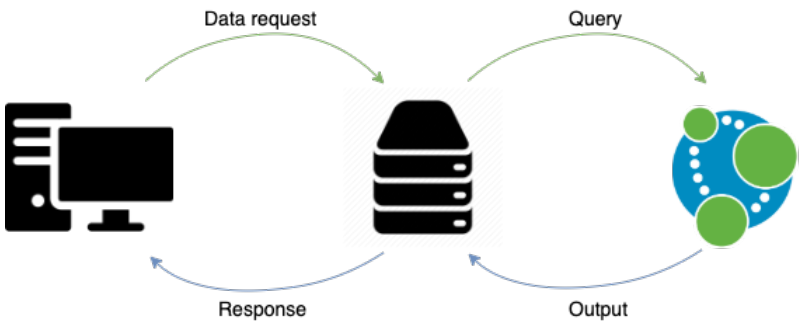


Figura 5.6: Flusso di gestione dei dati

Prima di passare ad illustrare cosa sia e come funzioni Neo4j, è utile evidenziare come un dataset di email possa essere sovrastrutturato. Questo, infatti, può contenere dei dati non rilevanti ai fini dell'analisi. Per questo motivo, nel caso particolare dei dataset delle email, in fase di analisi viene effettuata una procedura di pulizia che rimuove queste parti come, ad esempio, le parti di testo che iniziano con "RE:" che indicano una email alla quale si è inviata una risposta. Rimandiamo al lavoro di [8] per una spiegazione maggiormente dettagliata di come viene svolta questa fase di pulizia poiché il nostro lavoro riprende la stessa metodologia in quanto si vuole porre come la sua naturale prosecuzione.

5.2.1 Neo4j e il linguaggio Cypher

Abbiamo posto diverse volte l'attenzione su come la visualizzazione classica di una rete sociale sia quella costituita da circonferenze unite tra loro da archi. Di conseguenza, ci è sembrato naturale una memorizzazione che si basasse sulla medesima forma, utilizzando quindi un database a grafo. Un database a grafo, infatti, come

suggerisce anche il nome, salva i dati sotto forma di nodi e archi collegati tra loro e, per questo motivo, si pone come la più valida alternativa alla memorizzazione dei dump sociali.

Figura 5.7: Logo di Neo4j

Sebbene in commercio esistano diversi tipi di database a grafo, la scelta è ricaduta su **Neo4j** per diversi fattori. In primo luogo Neo4j è disponibile sia con licenzia commerciale per sistemi aziendali che necessitano di alti livelli di prestazioni e sicurezza, ma anche con licenza open source. Altro fattore che ha influito sulla scelta di questo database è la semplicità del linguaggio di interrogazione utilizzato. Nonostante Neo4j utilizzi un linguaggio di interrogazione nativo denominato **Chyper**, questo risulta essere molto immediato in quanto è molto simile ai linguaggi di interrogazione più conosciuti come SQL dal quale trae ispirazione. La sintassi di Cypher fornisce un modo visivo e logico per abbinare modelli di nodi e relazioni all'interno grafo con cui gli utenti possono dichiarare cosa si vuole selezionare, inserire, aggiornare o cancellare.

```
MATCH (n:Friend)
WHERE n.age> 25
return n
```

Figura 5.8: Esempio di una query in Chyper

Cypher, infine, è anch'esso open source grazie al progetto openCypher che fornisce una specifiche aperte del linguaggio e un'implementazione di riferimento del parser. Gli ultimi fattori determinanti per la scelta di Neo4j come database di riferimento

per la nostra applicazione risiedono nel fatto che è possibile estendere le funzionalità del database mediante una serie di estensioni chiamate procedure APOC e perché il database è accompagnato da uno strumento di sviluppo web *user friendly* chiamato **Neo4j Browser**, una GUI con cui è possibile interrogare il database per ricevere i dati in remoto.

Neo4j è utilizzato da aziende come Microsoft, Ebay, IBM, Volvo e moltissime altre ed si mostra estremamente comodo per rappresentare con estrema naturalezza strutture ad albero o grafi. Con un database di questo tipo le operazioni di interrogazione risultano essere molto più veloci rispetto alle controparti relazionali poiché la ricerca di un nodo in relazione con un altro è un operazione primitiva che non richiede calcoli complessi come può essere un *join* tra due o più tabelle. Sebbene Neo4j offre molteplici vantaggi per l'analisi di un grafo, risulta piuttosto inefficiente per quanto rigurda ricerche complesse che coinvolgono confronti matematici tra tuple. Risulta totalmente inadeguato, infine, per la memorizzazione di file binari poichè non ha alcun tipo di funzione per la memorizzazione di immagini o video obbligando di fatto ad adottare un sistema differente di memorizzazione per questi tipi di informazioni. Future implementazioni di questo applicativo potrebbero tenere in considerazione questo problema e potrebbero cercare ad una soluzione adatta a questa problematica.

5.2.2 Uploading dei dati

Il punto di forza del nostro applicativo proviene dal fatto che il gran parte del lavoro viene svolto in maniera asincrona dal server. Qualora l'utente stia caricando un dataset all'interno dell'applicazione, può continuare ad utilizzarla senza dover aspettare che l'elaborazione il caricamento su di Neo4j sia effettivamente completato. Poiché avevamo la necessità di doverci interfacciare con il database a grafo, abbiamo scelto di scrivere il nostro server in linguaggio **python 2.7** poiché per quest'ultimo è disponibile un toolkit denominato **py2neo** che permette di interagire con il database.

Py2neo è una libreria che permette di accedere a Neo4j da applicazioni python e da linea di comando, molto semplice ed intuitiva da utilizzare. Il codice [*Vede-*

re Appendice A: Creazione della connessione verso Neo4j] permette di aprire una connessione verso il database.

Una volta aperta la connessione, py2neo mette a disposizione tutta una serie di API per svolgere tutte le opzioni di creazione, interrogazione ed eliminazione verso il database [*Vedere Appendice A: Creazione del Facebook User Node tramite py2neo*].

È interessante evidenziare come, a differenza di quanto succede con altri linguaggi che si interfacciano con altri database, con Neo4j e py2neo non ci siamo bisogno di chiudere la connessione una volta conclusa l'operazione.

Per l'uploading dei dati su Neo4j avviene mediante script python denominati in maniera differente in base al social netowork che rappresentano: **fbDumpUploader.py**, **twitterDumpUploader.py** e **mboxDumpUploader.py**[*Vedere Appendice A: Upload della lista amici di Facebook su Neo4j*]. Ciascuno di questi, scompatta il file caricato, analizza i file in ingresso (*.json o *.mbox) e carica su Neo4j tutti dati utili ai fini di un'analisi investigativa.

	Nodi	**Archi**
Facebook	· Utente root · Amici · Amici rimossi · Post · Post degli amici · Dati sulla posizione · Contatti telefonici	· Relazioni di amici e utente root · Relazioni tra amici taggati insieme · Relazioni tra amici e contenuti testuali (post, commenti . . .) · Relazioni tra contenuti testuali e dati di geolocalizzazione
Twitter	· Utente root · Follower · Following · Tweet · Dati sulla posizione	· Relazioni tra follower e utente root · Relazioni tra following e utente root · Relazioni tra tweet ed utente root · Relazioni tra utenti taggati negli stessi tweet · Relazioni tra tweet e dati di geolocalizzazione
Mbox	· Utente root · Mittenti · Destinatario · Email	· Relazioni mittenti ed email · Relazioni destinatari ed email

Tabella 5.2: Dati caricati per ciascun social network

5.2.3 Organizzazione dei dati caricati

A tutti i nodi caricati all'interno di Neo4j, indifferentemente dal social netowork da cui sono stati estratti, vengono associati alcuni metadati come il timestamp o un *node degree* ovvero un grado del nodo.

In primo luogo, abbiamo dovuto pensare ad un sistema per permettere in maniera esclusiva l'accesso ai dati caricati da uno specifico utente per evitare che una persona possa accedere ai dati caricati da un'altra. A questo proposito abbiamo pensato di associare a ciascun nodo tre etichette che ne identificano univocamente il nodo all'interno del database

1. Una stringa tra "Facebook", "Twitter" e "Mail box" che identifica specificatamente il social netowork
2. L'indirizzo email dell'utente a cui il dataset scaricato è associato
3. L'indirizzo email dell'utente che sta utilizzando la nostra applicazione

In questo modo riusciamo ad evitare che gli utenti utilizzatori dell'applicazione accedano a dati che non siano stati caricati da loro. Questo perché, sia nella nostra applicazione, ma anche nei servizi di social networking o email non possono esistere due utenti aventi lo stesso indirizzo di posta elettronica. Questo sistema, inoltre, ci assicura che, in fase di elaborazione i dati un dump non vengano confusi con i dati di un altro, già caricato in precedenza.

Oltre ai metadati che identificano la proprietà di un nodo rispetto ad un account, ricollegandoci ai concetti già citati nella sezione 2.3.1, a qualsiasi dato che rappresenta una persona fisica del grafo sociale, abbiamo associato due valori indicati un *in degree* ed un *out degree*. Lo *in degree* indica quante volte una persona è stata citata dall'utente proprietario del profilo. L'*out degree*, invece, indica quante volte l'utente proprietario è stato citato complessivamente da una determinata persona. La somma di questi due valori, infine, produce il *node degree* ovvero il grado complessivo del nodo.

Abbiamo usato uno schema simile anche per quanto riguarda i dati testuali con-

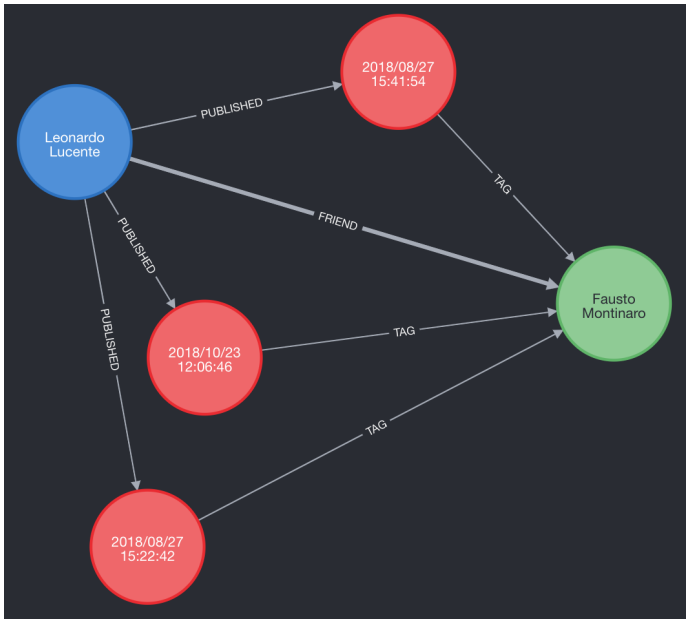


Figura 5.9: Porzione di grafo memorizzata su Neo4j

divisi all'interno del social network. In questo caso, però, il *node degree* è un valore unico che indica il numero di persone taggate all'interno di quel contenuto, sia esso uno post, un tweet o una email. Anche agli archi viene associato un *edge degree* che rappresenta il numero complessivo di volte per cui due attori siano presenti nello stesso contenuto testuale. Queste etichette calcolate in fase di uploading, si riveleranno utili in fase di visualizzazione per rendere maggiormente esplicativo il nostro grafo finale.

5.2.4 Estrazione testuale

Nel nostro lavoro abbiamo ampiamente discusso di come l'analisi testuale sia fondamentale nello svolgimento di un'analisi forense. Per questo motivo, la nostra applicazione oltre a visualizzare, sotto forma di grafo, i dati analizzati è anche in grado di analizzare tutti i contenuti testuali del dataset in esame, per poi estrarne tutti i termini significativi e rappresentarli, in una fase successiva, in una forma più comprensibile dall'utente finale. Questa operazione, tuttavia, risulta essere molto dispendiosa dal punto di vista computazionale, soprattutto se si lavora su insiemi di

dati molto grandi, arrivando ad impiegare anche diversi minuti. Proprio per questo motivo abbiamo lasciato all'utente la possibilità di scegliere se caricare i dati estratti su Neo4j al momento dell'uploading di tutti dati o di effettuare questo caricamento in una fase successiva.

Per effettuare l'estrazione testuale, viene calcolata una matrice *TFIDF* come descritto precedentemente nel paragrafo 3.3. In questo contesto, abbiamo una matrice $n \cdot m$ dove n è il numero totale dei documenti del *corpus*, mentre m indica il numero totale dei termini estratti da tutti i documenti. La posizione $n_i \cdot m_j$, quindi, indica il numero di occorrenze del termine m_j all'interno del documento n_j. Per effettuare questo calcolo, abbiamo utilizzato librerie specializzate di python quali :

- **nltk**: è una libreria che permette di creare script python per lavorare con i linguaggi naturali fornendo una serie di metodi per la classificazione, tokenizzazione, parsing, etc. Nello specifico, abbiamo usato la libreria nltk per il recupero delle stopwords, in quanto fornisce strumenti per la creazione automatica di insiemi contenenti questi termini in diverse lingue e poterli anche combinare insieme.

- **pandas**: è una libreria per la manipolazione e l'analisi dei dati. In particolare, offre strutture dati e operazioni per manipolare tabelle numeriche.

- **sklearn**: è un moduo python utile per il *machine learning* che contiene diversi metodi per la classificazione, il clustering ed altre utili funzioni. In particolare abbiamo usato *TfidfVectorizer* che permette la conversione di una collezione di documenti in una matrice TFIDF.

Nello specifico, il calcolo della matrice TFIDF avviene tramite una chiamata al metodo ***calculateTFIDF()*** [*Vedere Appendice A: Creazione della matrice TFIDF*] dove vengono prima identificate le stopwords, successivamente avviene una **tokenizzazione** dei documenti ed, infine, il calcolo della matrice TF-IDF. Grazie all'utilizzo combinato di diversi strumenti e a poche righe di codice, riusciamo ad effettuare l'intero calcolo in tempi piuttosto brevi. Con questo approccio, a ciascun termine viene

associato un peso che determina l'importanza della parola all'interno di quel documento, ma, poiché vogliamo ottenere il peso della parola sull'intero dataset, di ogni termine ne viene calcolata la media che ne definisce il peso complessivo. In una fase successiva all'estrazione e all'analisi, l'intera **bag of words** viene caricata su Neo4j secondo uno schema che associa un termine ad un contenuto testuale solo se il primo è contenuto nel secondo.

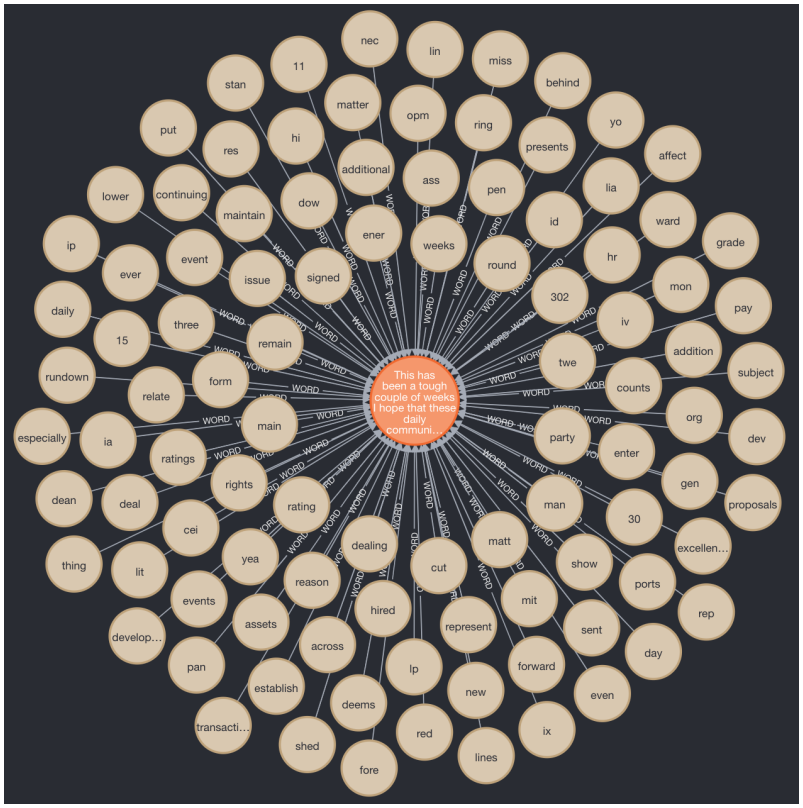


Figura 5.10: Esempio di associazione contenuti-parole in Neo4j

5.3 Interfaccia del client e visualizzazioni disponibili

L'interfaccia della nostra applicazione si divide in tre aree:

1. La prima parte è dedicata alle opzioni di filtraggio per poter interagire con il server. È possibile selezionare il range temporale, il valore minimo dei nodi e

degli archi ed effettuare ricerche per nome di persona o parola chiave

2. La seconda area corrisponde anche alla parte principale dove vengono rappresentati i dati ricevuti dal server. È possibile selezionare quattro differenti tipi di visualizzazione e delle quali discuteremo nelle prossime sezioni.

3. Nella terza area è possibile visualizzare informazioni supplementari relative al grafo rappresentato relative al nodo o arco selezionato (**selected**), sulla base del range selezionato *filtered* o su tutto il grafo in esame (**all**).

4. Permette l'apertura del drawer da cui è possibile accedere all'area personale dell'utente per poter cambiare l'email o la password con cui ci si è registrati e di eliminare eventuali dump già presenti nel sistema o caricarne di nuovi.

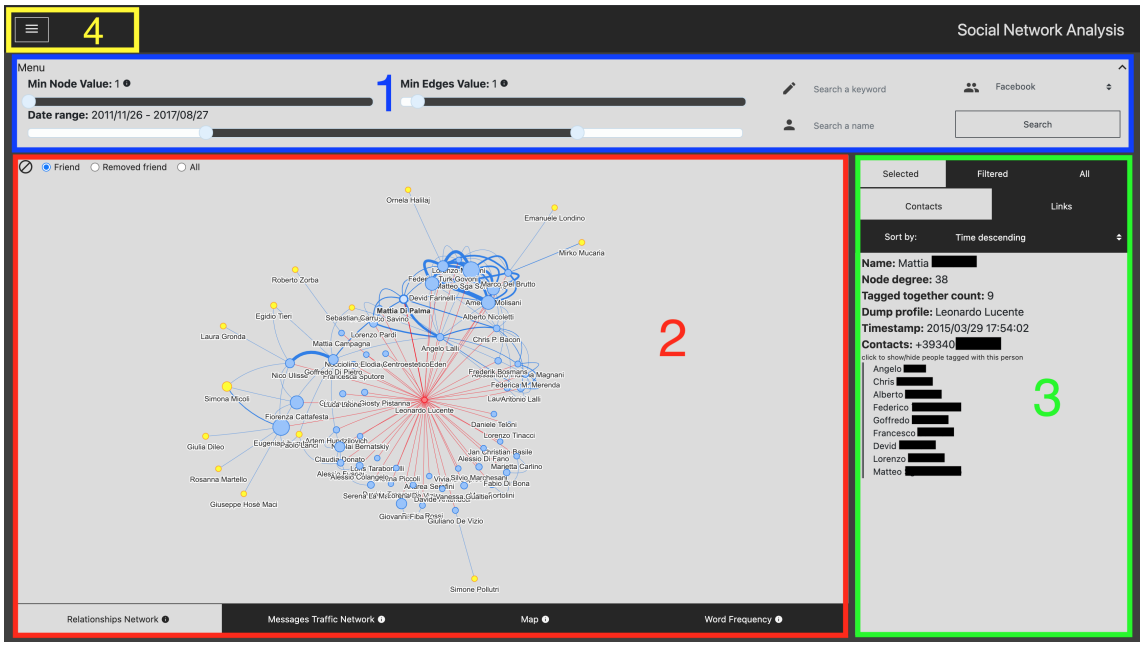


Figura 5.11: Interfaccia dell'applicazione

La comunicazione tra client e server avviene tramite chiamate asincrone, dove il primo richiede al secondo una certe porzione dei dati impostati tramite i filtri dell'area 1 e si vede restituire un oggetto JSON che viene rappresentato secondo la tipologia selezionata. In questo scenario, quindi, il client effettua chiamate ajax al server che risponde restituendo un oggetto JSON facilmente interpretabile dal client.

I dati restituiti, infine, vengono visualizzati mediante l'utilizzo di alcuni framework javascript che rappresentano la parte centrale del client. Nella nostra applicazione abbiamo utilizzati diversi schemi di assegnazione di ruoli che danno luogo a quattro visualizzazioni di tipologie differenti.

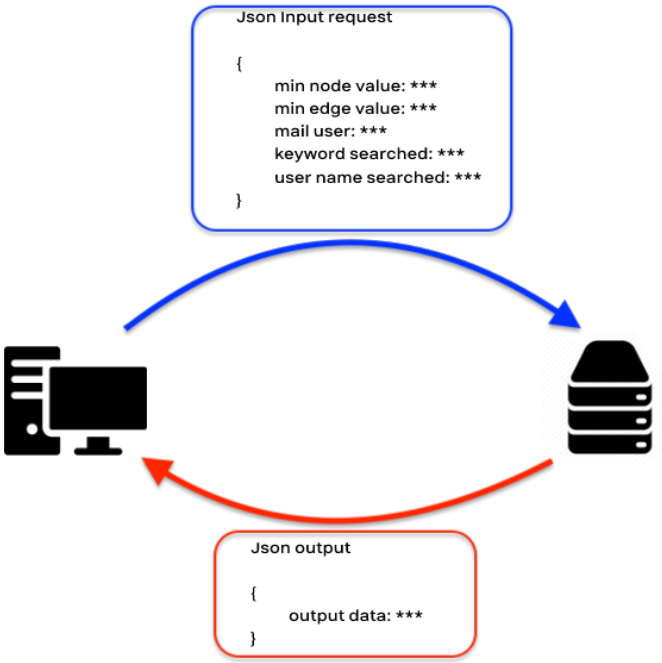


Figura 5.12: Richiesta e risposta dei dati

5.3.1 Relationship network

La prima visualizzazione disponibile nella nostra applicazione è un grafo relazionale. Questo scenario rappresenta la forma più basilare di un grafo sociale, ovvero l'interazione tra i vari attori presenti all'interno del dataset. In particolare il grafo delle relazioni permette di visualizzare come l'utente proprietario del dump abbia interagito con gli altri utenti. L'interazione rappresentata è specifica del social network analizzato. Per Facebook, infatti, rappresenta quando l'utente root ha aggiunto o rimosso un'altra persona, per Twitter indica la prima interazione con un'altro attore del grafo mentre per quanto riguarda le mailbox l'interazione è data dalla ricezione o dall'invio di un email da o verso un'altra persona.

In questo contesto è utile vedere come i vari attori abbiano interagito tra loro, per questo motivo due attori sono collegati da un arco qualora i due siano presenti nello stesso contenuto. Questo arco risulta essere più spesso, in base al numero di volte in cui le due persone sono presenti insieme nel range di tempo selezionato. Risulterà molto facile, quindi, identificare quali sono le coppie di persone con cui l'utente root abbia interagito contemporaneamente.

Anche la grandezza dei nodi è determinante nella nostra visualizzazione ed un nodo apparirà più grande, tanto è più alto è il suo *node degree* ovvero il numero complessivo di interazioni tra una persone e l'utente.

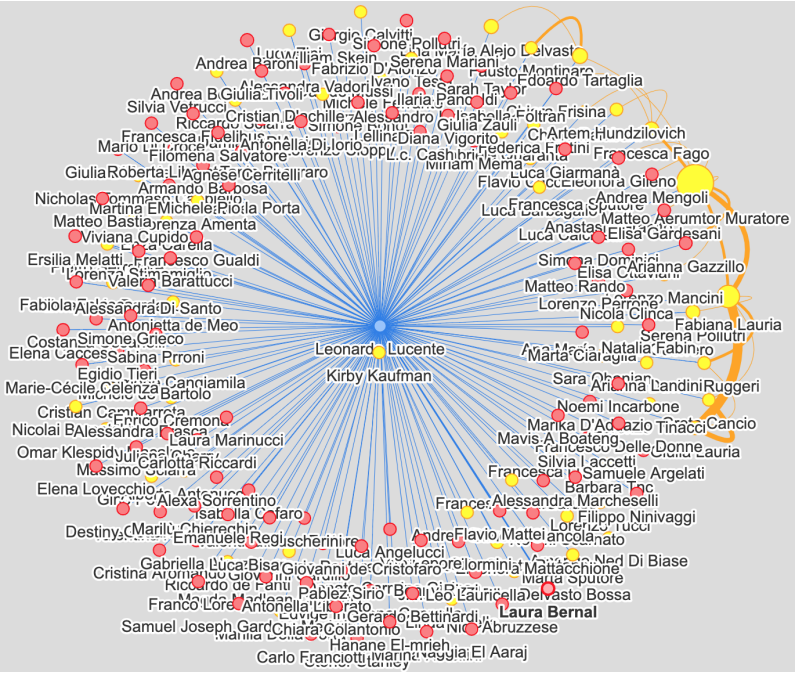


Figura 5.13: Esempio di visualizzazione del grafo delle relazioni

Per la visualizzazione del grafo abbiamo utilizzato un framework javascript denominato **neovis.js** che permette di rappresentare in maniera automatica i dati restituiti dal server utilizzando lo stesso modello con cui i dati vengono salvati sul database. Ogni elemento del grafo visualizzato è selezionabile, permettendo di mostrare una serie di informazioni aggiuntive come timestamp, il suo *node degree*, etc. Un limite di questo framework è che non permette di impostare un colore per differenti tipi di nodi. Questa potrebbe essere una limitazione nel caso in cui un

investigatore, ad esempio, voglia visualizzare, nell'ambiente Facebook, solamente gli amici rimossi. Per ovviare a questa problematica abbiamo reso selezionabile i tipi di nodi da visualizzare. Qualora, quindi, si volesse visualizzare solamente un sotto insieme dell'intera lista di attori, basterà cliccare sull'apposito radio button.

La nostra applicazione, infine, supporta la visualizzazione di più dataset dello stesso social, appartenenti a persone diverse e caricati dallo stesso utente. In questo caso, qualora, gli utenti proprietari dei dump siano presenti reciprocamente nella lista dell'altro utente, verrà visualizzato un arco che indica questo rapporto reciproco.

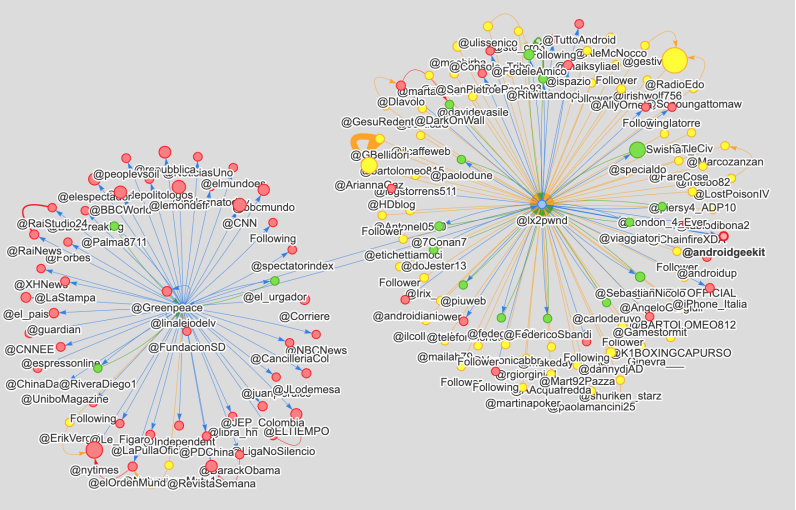


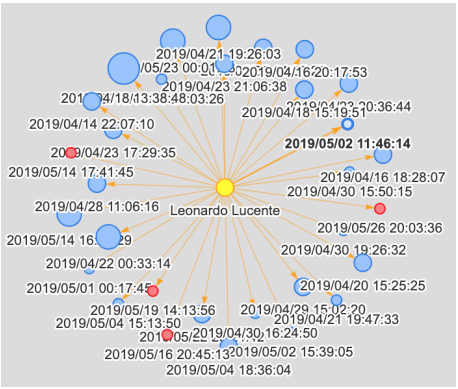
Figura 5.14: Visualizzazione simultanea di due grafi

5.3.2 Message traffic network

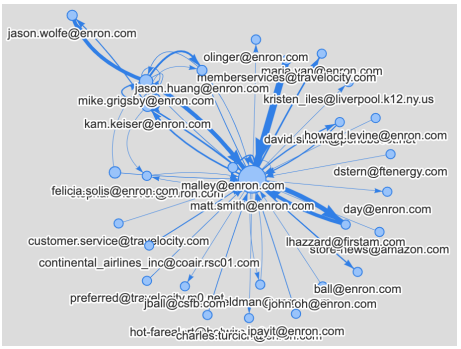
Il grafo sul traffico dei messaggi è molto simile a quello delle relazioni, in questo caso però vi è una differenziazione per quanto riguarda ciò che viene rappresentato dai grafici di Facebook e Twitter e quanto viene rappresentato dai dati estratti dal dump di una mail box.

Per quanto riguarda Facebook e Twitter, il grafo risultante mostra il nodo dell'utente proprietario del profilo collegato ai vari contenuti condivisi. A differenza del grafo delle relazioni, però, in questo caso gli archi sono dotati del verso che indica se

il contenuto è stato pubblicato dall'utente o se lo ha ricevuto. Anche in questo caso e possibile differenziare la visualizzazione del grafo in base alla tipologia del nodo. La grandezza dei vari nodi, inoltre, dipende da quante persone siano taggate.



(a) Visualizzazione per Facebook e Twitter



(b) Visualizzazione per le mailbox

Figura 5.15: Esempio di visualizzazione per il traffico del messaggi

La visualizzazione ottenuta è differente nel caso in cui si stia analizzando una mail box. Questo perché grazie ai campi che indicano il mittente e il destinatario e possibile ricostruire un grafo che mostra come sia avvenuto lo scambio tra tutti gli attori appartenenti al grafo. Come sopra, anche in questo caso la grandezza del nodo dipende dal suo *node degree* quindi da quanti messaggi quel determinato attore abbiamo inviato o ricevuto, ma questa volta è possibile ricavare informazioni significative anche dallo spessore dell'arco che collega due attori che appare più spesso in base al numero totale dei messaggi condivisi da quella determinata coppia.

In ambedue i casi ogni è possibile interagire con il grafo selezionato andando a mostrare alcune informazioni aggiuntive come il contenuto testuale di quel determinato nodo, il suo timestamp, il *node degree* ed eventuali persone taggate in quel contenuto o, per quanto riguarda gli archi, l'effettivo numero di email scambiate tra due attori.

5.3.3 Map

Attraverso i social network è possibile condividere con le altre persone la propria posizione o permettere agli stessi di acquisire i dati di geo localizzazione. Permettere, quindi, di potere acquisire e visualizzare i movimenti di un utente potrebbe rivelarsi fondamentale ai fini investigativi. Qualora l'utente abiliti il social network ad accedere ai dati sulla posizione, i contenuti condivisi con la rete verranno pubblicati con metadati riguardanti le coordinate geografiche che, in fase di uploading, vengono memorizzati come attributi sul nodo. Quando si sceglie questo tipo di visualizzazione, l'applicazione richiede i contenuti che presentano latitudine e longitudine per poi poterli rappresentare tramite **OpenLayers**.

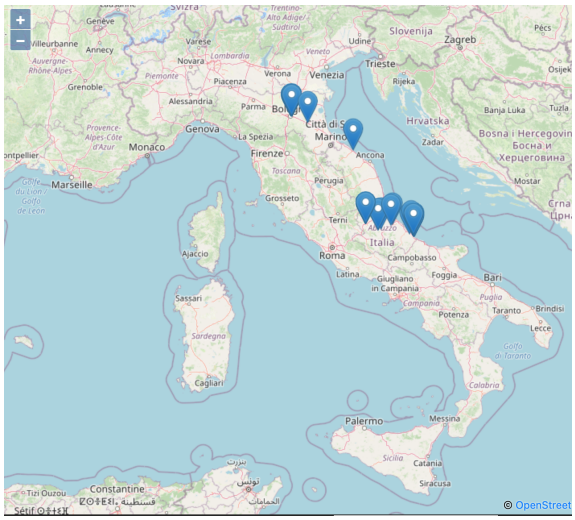


Figura 5.16: Visualizzazione della mappa

OpenLayers è una libreria javascript che permette la creazione e visualizzazione di mappe interattive all'interno del browser. Tramite OpenLayers è possibile ad accedere sia ad informazioni cartografiche poste sotto licenza come Google Maps o Bing, ma anche a mappe a contenuto libero come **OpenStreetMap**. Il JSON ricevuto viene interpretato da OpenLayers e, successivamente, i dati vengono posizionati sulla mappa. Anche gli indicatori posizionati sulla mappa sono interamente selezionabili e questo permettere di visualizzare l'effettivo contenuto condiviso in una

determinata posizione, il timestamp, le coordinate geografiche ed eventuali persone taggate.

Questa visualizzazione, al momento, è disponibile esclusivamente per Facebook e Twitter. La possibilità di accedere ai dati relativi alla posizione è una prerogativa dei social network come Facebook e Twitter che sfruttano i GPS dei dispositivi mobili per memorizzare e condividere questo genere di informazioni. I servizi di email, al contrario, non accedono a queste informazioni e non condividono nei loro *header* questo genere di informazioni per cui ci è stato impossibile ricostruire una mappa, ma lasciamo la possibilità di aggiungere questa funzione in implementazioni future.

5.3.4 Word frequency

La quarta ed ultima visualizzazione è quella legata alla ricerca dei termini significativi estratti dal dump secondo quanto descritto nella sezione 5.2.4. Essendo un'operazione estremamente dispendiosa dal punto di vista computazionale, abbiamo lasciato all'utente la possibilità di effettuare l'estrazione testuale o al momento dell'uploading insieme a tutti gli altri dati oppure in una fase successiva, qualora si chiedesse esplicitamente questo tipo di visualizzazione.

Anche in questo caso, il server restituisce al client un oggetto JSON, i cui dati vengono pre formattati in modo tale che il client li abbia già pronti per la visualizzazione. Per la visualizzazione abbiamo usato un framework javascript chiamato **Frappe**, una libreria molto semplice e leggera che permette la visualizzazione di diagrammi di Gantt all'interno del browser, ma che abbiamo dovuto modificare leggermente per permettere la visualizzazione degli elementi sulla stessa riga.

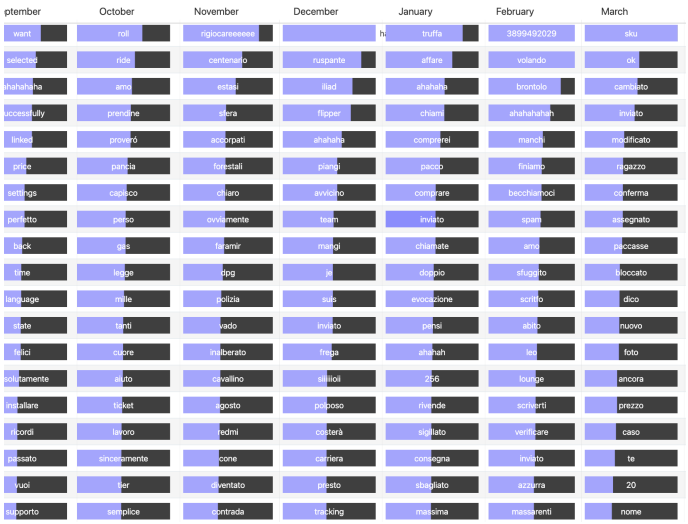


Figura 5.17: Timeline della frequenza delle parole

I vari elementi vengono raggruppati per mese e anno in base al timestamp del contenuto in cui compaiono. Ad ogni termine, inoltre, come abbiamo detto precedentemente, è associato un valore realativo al *term frequency – inverse document frequency* che indica la rilevanza che quel termine ricopre all'interno del corpus. Per ogni colonna, i vari termini vengono ordinati per valori decrescenti, posizionando in alto i termini che hanno ricevuto un punteggio *tf-idf* più elevato, mentre in basso troviamo i termini che, a causa del loro frequente utilizzo, hanno totalizzati dei punteggi più bassi. In questo modo è possibile individuare facilmente le parole che all'interno del corpus potrebbero avere un maggiore rilievo poiché utilizzati poco frequentemente. Oltre al posizionamento, **Frappe** colora di celeste tutti gli elementi del grafico in base al valore associato all'elemento stesso, quindi più un valore è alto, tanto più la barra di quel elemento risulterà essere colorata. Al contrario se un valore risulta essere basso, allo stesso modo la barra azzurra sarà inferiore rispetto alla barra grigia sottostante. Come per le visualizzazioni precedenti, infine, è possibile selezionare uno specifico termine per poter visualizzare quali siano i cui appare quello specifico termine.

Capitolo 6

Caso di studio

In questo capitolo andremo ad illustrare nello specifico come possa essere utilizzata la nostra applicazione ai fini di un'indagine investigativa. Per svolgere questa operazione avremo bisogno, ovviamente, di dataset su cui operare. Per quanto riguarda l'analisi relativa all'email abbiamo utilizzato il dataset 'Enron', un dataset facilmente reperibile in rete e davvero molto popolare in quanto usato altre volte con fini sperimentali. Per quanto riguarda, invece, Facebook e Twitter, useremo i miei dump personali provenienti da questi due social network.

Di seguito daremo una panoramica sui dataset utilizzati, cercheremo di illustrare come la nostra applicazione possa essere utile per estrapolare informazioni rilevanti andando a testarne le caratteristiche e come interpretare i risultati visualizzati. I dati su cui concentreremo la nostra analisi saranno presi in maniera arbitraria dai dataset sopra citati.

6.1 Panoramica dei dati utilizzati

I tre dump utilizzati nella nostra applicazione sono molto differenti tra di loro per formato, struttura interna e dati che è possibile ricavarne. È utile, quindi, offrirne una panoramica che permetterà di comprendere meglio le analisi successive.

La scelta dei dump utilizzati è stata mossa, in primo luogo da una motivazione

politica che riguarda il diritto alla privacy. I dati che condividiamo quotidianamente tramite le applicazioni di social network sono strettamente personali e risulta difficile che le persone vogliano che le proprie informazioni private siano accessibili a tutti. Durante lo sviluppo, infatti, sebbene alcune persone si siano mostrate interessate al nostro applicativo, molte altre sono apparse molto restie a fornirci i loro dati anche solo per testare il corretto funzionamento dell'applicazione. La scelta del dataset "Enron" è, quindi, dovuta al fatto che, in seguito allo scandalo che ha colpito questa società, le email riguardanti le persone coinvolte sono state rese pubbliche dalla Federal Energy Regulatory Commission durante l'indagine. Inoltre, grazie alla grande quantità di attori coinvolti, si presta molto bene al conseguimento del nostro obiettivo. Per Facebook e Twitter la scelta di utilizzare i miei dati personali è anche mossa da motivazioni personali. Sono un utente di queste applicazioni da ormai diversi anni, precisamente dal 2009 e, sebbene entrambe offrano un metodo per acquisire facilmente i propri dati passati, la loro visualizzazione ed interpretazione non è altrettanto elementare. Sebbene l'obiettivo ultimo della nostra applicazione sia l'utilizzo ai fini forensi, ciò non esclude che possa essere utilizzata anche per ripercorrere il cambiamento della propria rete sociale nel corso del tempo.

6.1.1 Facebook dataset

Il dataset del mio profilo Facebook è davvero molto ricco in quanto la mia iscrizione al social network risale al febbraio 2009. Da quel momento, ho fatto largo uso dell'applicazione, accumulando una grandissima quantità di dati che ben si presta al nostro tipo di analisi. Il mio dataset, infatti, consente l'estrazione di praticamente tutte le informazioni acquisibili da Facebook per un totale di 10220 nodi tra lista amici, post, commenti, etc. A loro volta, questi dati, caricati all'interno dell'applicazione hanno dato luogo a 14.194 relazioni.

Il dump è organizzato in un ordine gerarchico di cartelle dove il nome di ciascuna si riferisce a macro informazioni ricavabili da ciascuno dei file *.JSON che queste contengono al loro interno. Il file scaricato da Facebook è un file compresso in formato zip che può essere già utilizzato come input del nostro programma. Sarà

Nodi estratti	Quantità
Proprietario del dump	1
Amici	135
Amici rimossi	703
Post	2038
Post degli amici	726
Commenti	4944
Messaggi privati	169
Rubrica telefonica	1302
Dati di posizionamento	191
Partecipanti a DM	11

Archi calcolati	Quantità
Amicizie	838
Localizzazioni	191
Pubblicazioni	7731
Risposte a DM	166
Tagging	5044
Amici taggati insieme	224

Tabella 6.1: Dati ricavati dal dump di Facebook

poi il server a decomprimerlo, a reperire i dati dagli opportuni file JSON e a caricarli su Neo4j.

6.1.2 Twitter dataset

Il dump del mio profilo, contrariamente a quello Facebook, è di dimensioni molto più ridotte sebbene sia iscritto al social network più o meno dallo stesso tempo, ovvero da novembre 2010. La mia attività su questo social è stata inferiore rispetto al precedente, ciò nonostante è possibile ricavarne tutti i dati più significativi per un totale di 5003 nodi e 6748 archi.

La struttura interna dei file del dataset di Twitter è simile a quella di Facebook ovvero degli oggetti JSON. Questi, però, hanno estensione *.js e non sono organizzati una una struttura ad albero, ma sono tutti posizionati in una directory comune accesibile dal percorso **twitter** → **data**. Come Facebook, il dump scaricabile da Twitter è un file compresso in zip che può essere caricato direttamente nella nostra applicazione senza che necessiti di alcun tipo di pre elaborazione.

Nodi estratti	Quantità
Proprietario del dump	1
Solo follower	286
Solo following	111
Follower/following	36
Tweet	3506
Retweet	801
Tweet piaciuti	262

Archi calcolati	Quantità
Account che ci seguono	322
Account seguiti	147
Tagging	1665
Utenti taggati insieme	45

Tabella 6.2: Dati ricavati dal dump di Twitter

6.1.3 Mbox dataset (Enron)

La Enron Corporation, fondata nel 1985, è stata una delle più grandi multinazionali statunitensi operante nel settore dell'energia negli ultimi anni XX secolo per poi fallire nel 2001. La società, infatti, venne coinvolta in forte scandalo che la condannò alla bancarotta, in quanto, in seguito ad un'inchiesta, si venne a scoprire che l'azienda aveva mentito sui suoi profitti ed era stata accusata di una serie di affari illeciti, incluso l'occultamento di debiti in modo che non apparissero nei rendiconti dell'azienda. Tutto questo portò ad accusare gran parte del consiglio di amministrazione per frode e riciclaggio di denaro [14].

Il seguito allo scandalo, l'intero dataset venne reso pubblico e comprende più di 2000 email e 150 utenti, per lo più dirigenti aziendali. In particolare, il dataset che stiamo utilizzando è scaricabile grazie alla Carnegie Mellon University School of Computer Science [4] che ha predisposto un dataset privo di problemi di integrità, non presenta allegati e gli indirizzi email non validi sono stati convertiti in qualcosa simile a *user@enron.com* o in *no_address@enron.com* quando nessun destinatario viene specificato.

Il file da noi utilizzato è un unico file avente estensione *.mbox contente diverse email aventi come utente principale Adam Smith. Il file si presenta come una concatenazione di email differenti delimitate dal campo *From:* ed è lo stesso preparato

Nodi estratti	Quantità
Nodi diretti	1449
Nodi indiretti	296
Mail	1642

Archi calcolati	Quantità
Archi diretti	2483
Archi indiretti	387
Mail inviate	3019
Mail ricevute	9936

Tabella 6.3: Dati ricavati dal dump di Enron

ed utilizzato per il lavoro di [8].

6.2 Analisi della rete sociale

In questa parte del nostro lavoro andremo a fornire un esempio di analisi sociale che è possibile effettuare mediante la nostra applicazione e che sia di supporto ad azioni investigative. Cercheremo di mostrare come sia possibile fornire un clusterizzazione degli attori presenti all'interno della rete sociale, definire quali siano i contenuti condivisi tra lo stesso gruppo di attori, stabilire in quali luoghi si sia mosso l'utente analizzato e quali siano le parole più rilevanti ai fini di un'indagine.

Come abbiamo già detto precedentemente, il nostro obiettivo è quello di continuare e ampliare il lavoro di [8] che abbiamo usato come punto di partenza per lo sviluppo della nostra applicazione. Poiché il dataset di input utilizzo per l'analisi delle mailbox risulta essere essere lo stesso, così come il metodo di analisi ed estrazione, sorvoleremo sul fornire una spiegazione di quando si può evincere con queste informazioni poiché già dettagliatamente discusse all'interno di questo progetto. Ci concentreremo, quindi, sui due nuovi ambienti che hanno trovato spazio all'interno del nostro lavoro ovvero Facebook e Twitter.

6.2.1 Analisi sociale con dataset di Facebook

Precedentemente abbiamo affermato che avremmo preso i nostri dati in maniera del tutto casuale, ma bisogna evidenziare come bisogni sempre ponderare la scelta dei

dati da porre sotto la lente d'ingrandimento. Rappresentare un'enorme quantità di dati non sarebbe di nessun aiuto ai fini di un analisi perché, a causa della grandezza del grafo stesso, potremmo non ricavarne nessuna informazione utile. Per questo, per poter esplicare meglio le funzionalità della nostra applicazione, prendiamo come riferimento un sotto insieme di dati compresi come lasso di tempo tra il 27/11/2011 e il 09/08/2017, il cui valore minimo dei nodi sia 1, mentre il valore minimo degli archi sia 0. Quest'ultimi due valori indicano che vogliamo che il grafo restituito rappresenti tutte le persone con cui l'utente proprietario del profilo, in questo caso io, Leonardo Lucente, abbia interagito almeno una volta e non ponendo alcun limite al numero di volte che due utenti possano essere stati taggati insieme.

Grafo delle relazione per Facebook

La visualizzazione del grafo delle relazioni può essere impostata su tre categorie di nodi:

- Amici
- Amici rimossi
- Sia amici che amici rimossi

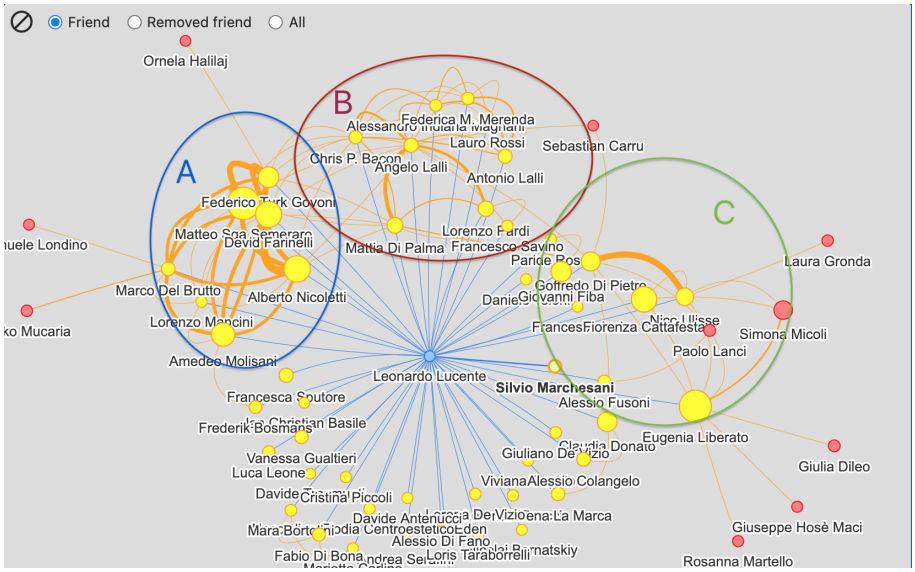


Figura 6.1: Esempio di grafo sociale per Facebook

Il grafo mostra in giallo quelli che, nell'intervallo di tempo selezionato, sono utenti presenti nella lista amici del soggetto in analisi, mentre in rosso quelli che, nello stesso intervallo di tempo, sono stati eliminati da questa lista. È facile distinguere anche tre gruppi di persone fortemente connessi tra di loro rappresentati in figura dalle lettere A,B e C. Questi tre gruppi, inoltre, sono anche connessi tra loro ed in particolar modo molti utenti presenti nel gruppo A sono in relazione con utenti del gruppo B e, seppur in misura minore, utenti di B sono connessi con altri di C. Le relazioni tra questi nodi si possono tradurre in legami del mondo reale dove ognuno degli attori faccia parte di un reale gruppo all'interno della sfera di amicizie del soggetto, ma che abbia anche un qualche tipo di rapporto meno importante con persone facenti parte di gruppi differenti. Allo stesso tempo, a causa della mancanza di archi tra utenti di A ed utenti di C, si potrebbe supporre come questi utenti non si conoscano di persona e non ci sia alcun tipo di relazione.

Lo spessore degli archi, ci può aiutare a dedurre dal grafo come il nome etichettato come "Matteo Semeraro" sia stato taggato molteplici volte con altri utenti come "Federico Govoni", "David Farinelli" e "Alberto Nicoletti" rispetto ad altri. Allo stesso modo l'utente "Goffredo di Pietro", essendo stato taggato molte volte con l'utente "Nico Ulisse" potrebbe avere una relazione più forte rispetto ad altri nodi con cui è stato taggato un numero inferiore di volte. Il voler taggare ripetutamente insieme due utenti e quindi il voler condividere con entrambi la stessa informazione potrebbe identificare come oltre ad essere in relazione con l'utente root, questi siano anche in forte connessione tra loro nella vita reale.

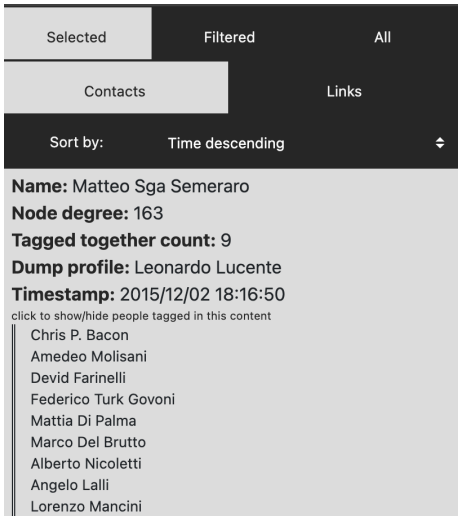


Figura 6.2: Vista sullo specifico nodo

In questa vista è anche determinante la grandezza dei nodi ed infatti risaltano all'occhio alcuni nodi come "Eugenia Liberato", "Fiorenza Cattafesta" ed alcuni altri già citati precedentemente come "Matteo Semeraro" o "Devid Farinelli". Il fatto che questi nodi appaiano in dimensioni maggiore rispetto agli altri vuol dire che l'utente root ha interagito più con questi che con tutti gli altri. Qualora si voglia visualizzare il numero esatto di iterazioni avute o tutte le persone taggate con un determinato nodo basterà cliccare sul nodo specifico per ottenere tutte le informazioni del caso come mostrato in figura 6.2.

Traffico dei messaggi per Facebook

Continuiamo mostrando la vista sul traffico dei messaggi. Con questa visualizzazione è possibile mostrare lo storico dei messaggi, differenziandoli per tipo e visualizzando quelli che sono i contenuti all'interno dei quali l'utente ha taggato più persone. Nello specifico questa la visulizzazione di questo grafo può essere differenziata per quattro diversi tipi di nodi a cui si aggiunge una vista complessiva d'insieme:

- Post
- Post Degli amici
- Commenti

- Messaggi privati
- Vista complessiva

Usando gli stessi filtri utilizzati per la precedente visualizzazione possiamo concentrarci su di uno dei nodi più grandi emersi dalla precedente visualizzazione "Devid Farinelli". La vista complessiva di questa visualizzazione si presenta come in figura *6.3*

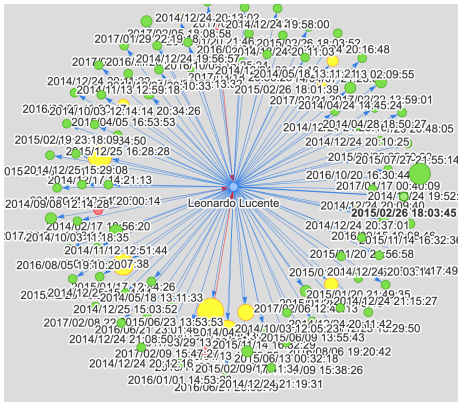
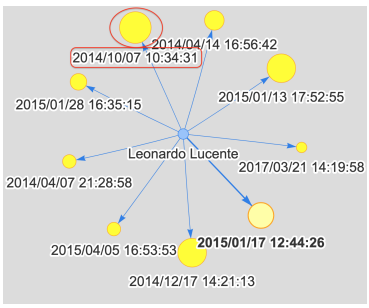
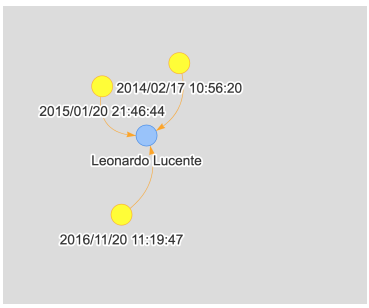


Figura 6.3: Vista sullo specifico nodo

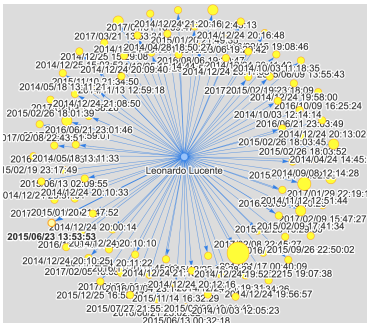
Sebbene il grafo possa sembrare molto confuso, è già possibile notare come l'utente abbia ricevuto alcuni messaggi dall'attore "Devid Farinelli" e che alcuni contenuti sono più grandi di altri, sinonimo di una maggiore presenza di persone taggate in quel determinato contenuto. Differenziando la visualizzazione per tipologia specifica di nodo si ottengono grafi come in figura *6.4*



(a) Grafo dei post



(b) Grafo dei post degli amici



(c) Grafo dei commenti

Figura 6.4: Visualizzazione del traffico dei messaggi

Il grafo *6.4 (b)* mostra nodi aventi tutti la stessa dimensione il che vuol dire che non ci sono altre persone taggate i questi contenuti. Gli archi, inoltre, sono entranti rispetto all'utente e questo vuol significare che si tratta di contenuti condivisi dall'attore sul profilo dell'utente. Situazione differente invece per i grafi *6.4 (a)* e *6.4 (c)* dove spiccano subito nodi di dimensioni maggiori rispetto agli altri di dimensioni più o meno inferiori. In particolare in *fig. 6.4 (a)*, selezionando uno dei nodi di dimensioni maggiori è possibile visualizzare tutte le informazioni ad esso collegate tra cui il contenuto testuale.

Figura 6.5: Vista sullo specifico nodo

Visualizzazione geografica per Facebook

La visualizzazione basata sui dati di geolocalizzazione si discosta per rappresentazione dalle due viste in precedenza, ma risulta essere altrettanto utile per poter ricostruire i movimenti compiuti dall'utente. Nell'intervallo 27/11/2011 - 09/08/2017, i dati mostrano una maggiore condivisione da luoghi situati in Italia e alcuni altri in Francia, precisamente a Parigi.

Questo quadro, porta a pensare che la localizzazione principale dell'utente sia in Italia, mentre il contenuto condiviso a Parigi potrebbe indicare che si sia trattato di un evento occasionale. Andando ad ingrandire la mappa dove vediamo come

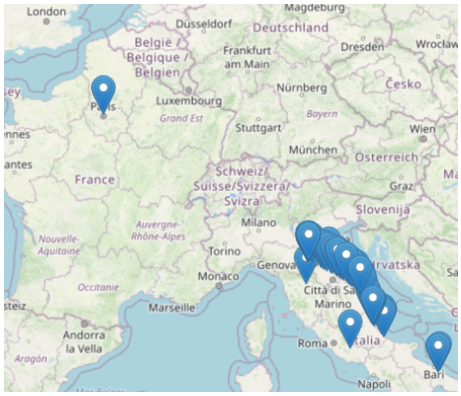
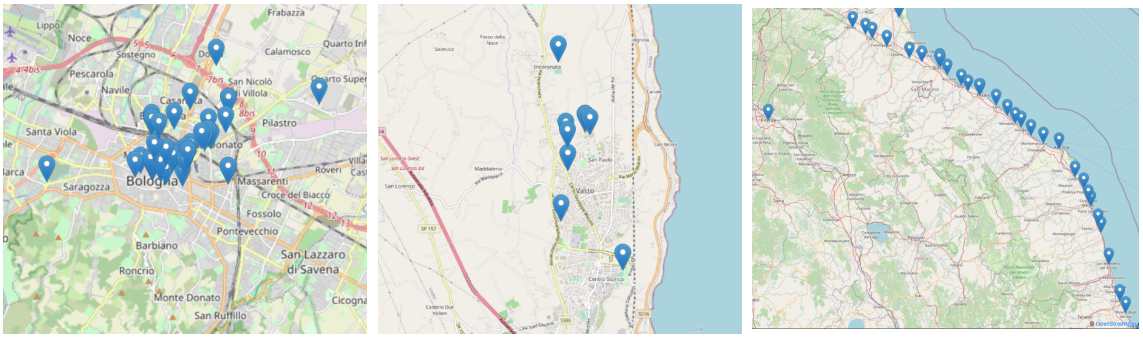


Figura 6.6: Visualizzazione cartografica

una maggiore condivisione di contenuti provenga da due città in particolare, Bologna (BO) e Vasto (CH), ma che molti contenuti siano stati condivisi tra numerose posizioni situate tra la prima e la seconda.



(a) Mappa della condivisione a Bologna

(b) Mappa della condivisione a Vasto

(c) Mappa della condivisone sull'adriatica

Figura 6.7: Visualizzazione cartografica

Considerando l'arco temporale analizzato, appare improbabile che l'utente abbia cambiato ripetutamente la sua posizione, ma che, piuttosto, la continua condivisione proveniente da due città particolati collegate da questa tratta, dipenda dal fatto che questo si muova principalmente in queste due zone e che utilizzi la tratta evidenziata come itinerario ricorrente per muoversi tra le due città.

Andando a selezionare i punti di questa tratta come in figura *figura 6.8*, infatti, si può notare come questi dati siano stati acquisiti da parte di Facebook tutti lo stesso giorno a poche ore di distanza l'uno dagli altri. Questo porta a classificare

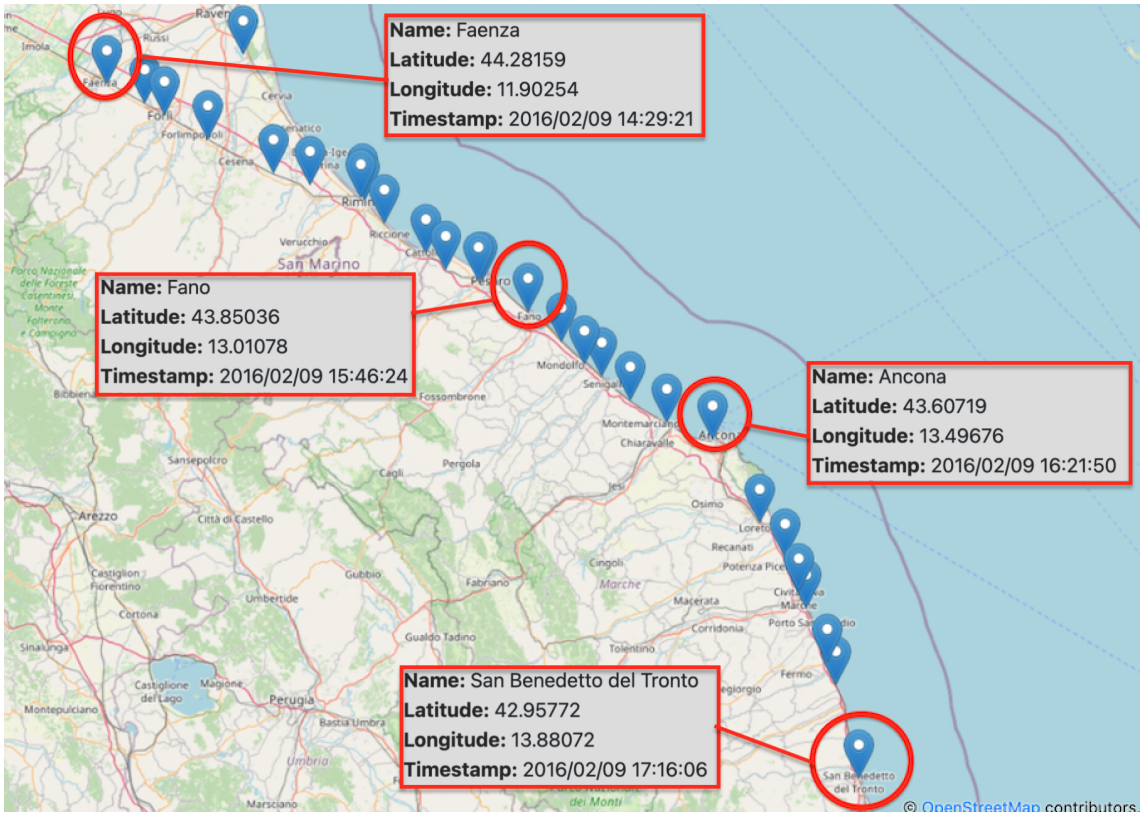


Figura 6.8: Dettagli di alcune posizioni

la nostra intuizione come corretta, ovvero che effettivamente l'utente si sia mosso lungo questo percorso nel febbraio 2016.

Visualizzazione per la frequenza delle parole

La visualizzazione riguardante la frequenza delle parole, come abbiamo già detto, misura il peso che ciascun termine ricopre all'interno del corpus dei documenti. Con questa è possibile visualizzare le parole che sono state utilizzate meno frequentemente dall'utente e che, per questo motivo, potrebbero essere centrali nella comprensione dei contenuti. Nell'intervallo selezionato otteniamo lo schema come in *figura 6.9*

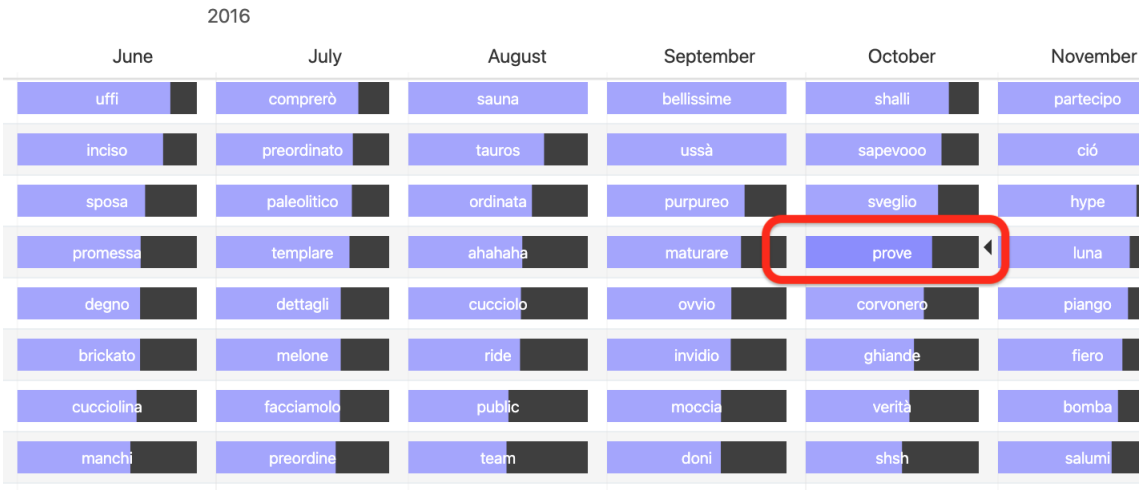
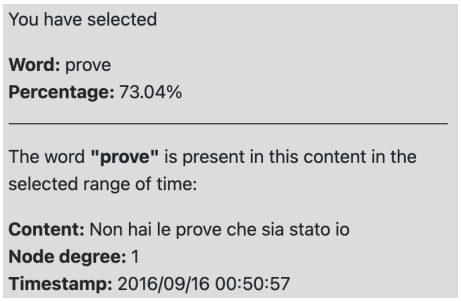
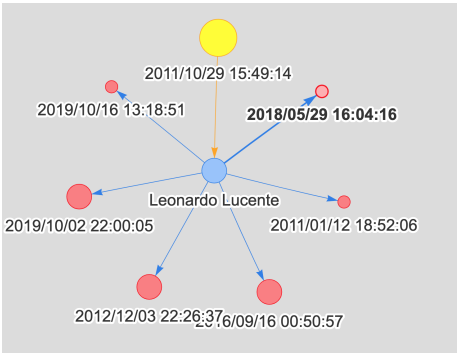


Figura 6.9: Timeline della frequenza delle parole

Le parole posizionate più in alto sono quelle che hanno ricavato un punteggio *tf-idf* più elevato. Selezionando una parola è possibile visualizzare i contenuti in cui compare. In questo modo è possibile contestualizzare il termine per poter ricavare eventuali significati o informazioni nascoste. Selezionando la parola precisa, ad esempio "prove", è possibile visualizzare il suo punteggio che in questo caso risulta essere 73.04%. Questo è un punteggio si elevato, ma che comunque non è il massimo raggiungibile e che lascia intendere che la parola è stata usata altre volte anche al di fuori dell'intervallo temporale selezionato.



(a) Vista specifica sulla parola



(b) Messaggi ricavati selezionando la parola interessata

Figura 6.10: Utilizzo combinando per la ricerca delle parole

6.2.2 Analisi sociale con dataset di Twitter

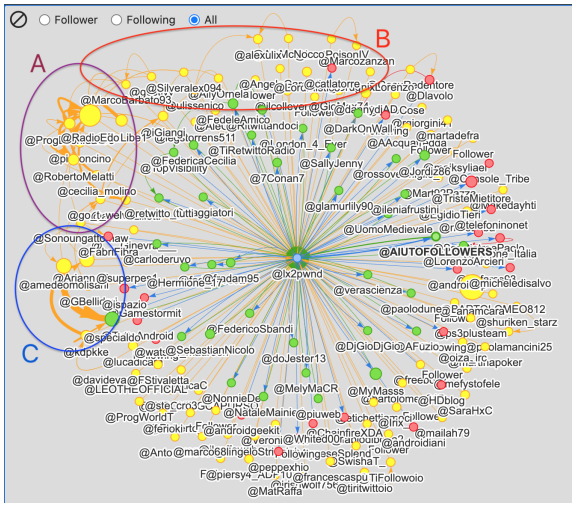
Le informazioni ricavabili analizzando un dataset Twitter sono del tutto simili a quelle che si possono dedurre dall'analisi di un dump di Facebook. È interessante però notare come le relazioni di "utenti seguiti" ed "utenti che ci seguono" portano a strutturare il grafo delle relazioni in maniera differente rispetto a quello di cui abbiamo precedentemente discusso. Ci ficalizzeremo, pertanto, esclusivamente su queta parte poichè sarebbe superfluo discutere nuovamente di argomenti già trattati e che risulterebbero essere del tutto simili, eccezione fatta per lievissime sfumature.

Grafo delle relazione per Twitter

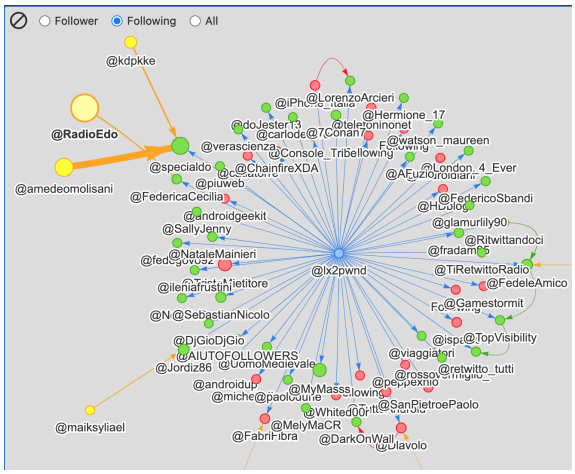
All'interno di Twitter, le relazioni tra gli utenti si differenziano da quelle che si possono instaurare su Facebook. In quest'ultimo, infatti, la relazione è bilaterale, ovvero nel momento in cui un utente A aggiunge un utente B, anche B aggiunge A. Questo, però non vale per Twitter dove se un utente A decide di seguire un utente B, quest'ultimo può non ricambiare. Questo ci porta ad identificare tre tipologie di attori all'interno del grafo di Twitter:

1. utenti che si seguono
2. utenti da cui si viene seguiti
3. utenti che si segue, ma da cui, allo stesso tempo, si viene seguiti

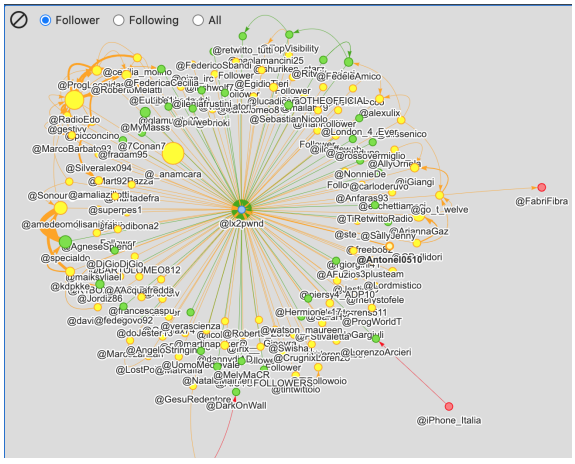
È interessante a questo proposito vedere come si configuri il grafo delle relazioni per il dump di Twitter. In *6.11* possiamo vedere il grafico calcolato sul mio dump nel periodo compreso tra il 18/11/2010 e il 25/04/2020. Anche se ad una prima occhiata possa sembrare piuttosto confusionario, in realtà è possibile notare come ci siano dei nodi, quindi degli attori, che hanno sia archi entranti, ma anche uscenti che appartengono alla categoria 3 nell'elenco sopra citato. Gli altri, invece, possiedono solo uno dei due archi e fa si che questi attori si classificano o come "follower" o "following". Si può anche notare come ci siano delle porzioni di grafo più connesse (cluster A) rispetto ad altre meno connesse come i cluster B o C.



(a) Grafo completo



(b) Vista parziale per gli utenti seguiti



(c) Vista parziale per gli utenti da cui veniamo seguiti

Figura 6.11: Grafo delle relazioni per Twitter

Differenziano la vista per tipologia di utenti, otteniamo i due grafi riportati in *fig 6.11 (b)* e *fig 6.11 (c)*. Da qui è più semplice delineare quali siano i singoli attori con cui l'utente ha avuto più interazione (che vengono etichettati con il proprio username utilizzato all'interno dell'applicazione) che corrispondono essere "@radioedo" e "@anamcara". Selezionando ciascuno di questi due nodi otteniamo i rispettivi dettagli tra cui il link che ci riporta direttamente sul profilo Twitter selezionatoo il numero di interazioni totali tra l'utente root e quello selezionato.

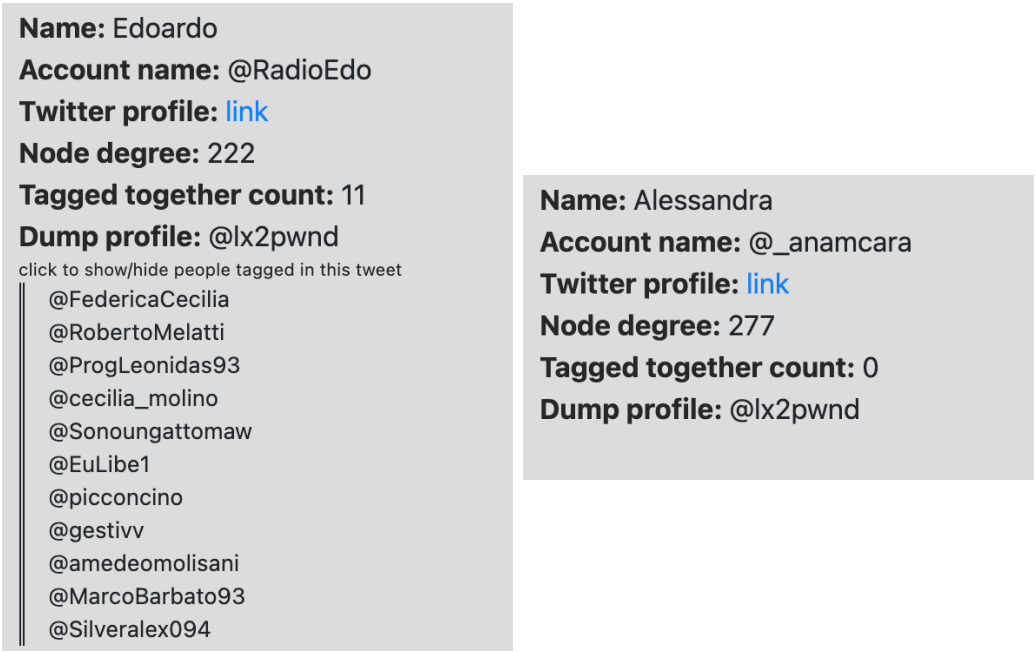


Figura 6.12: Vista specifica per gli utenti

Capitolo 7

Conclusioni

Nel nostro lavoro abbiamo cercato di delineare un quadro riguardante l'analisi sociale in campo forense, partendo dal concetto di grafo e scendendo sempre più nel dettaglio, fino a presentare la nostra applicazione. La nostra ambizione era quella di continuare il lavoro di [8] per creare un'applicazione general purpose che permettesse di svolgere analisi non su un dataset pre impostato, ma che fosse liberamente selezionabile dall'utente. Se la precedente versione dell'applicazione operava esclusivamente sul dataset Enron, la nostra applicazione permette anche di rappresentare i dati proveniente da altri due social netowork ovvero Facebook e Twitter.

Volevamo, inoltre, che lo studio fosse trasversale rispetto all'insieme di dati di input sfruttando al meglio la grande eterogeneità dei dati stessi ricavabili dai social network. Per questo siamo andati ad analizzare non solo gli aspetti sociali individuabili tra gli attori, ma anche la diffusione dei contenuti sia nel tempo che nello spazio e, per ultimo, abbiamo utilizzato tecniche di *mining* testuale. Per la prima parte delle nostre analisi siamo andati a studiare strumenti già esistenti e come questi rappresentassero i dati. Questa preparazione, si è tradotta nell'utilizzo di schemi in 2D per la rappresentazione della rete sociale, ma anche per la diffusione dei contenuti. Per quanto riguarda la fase di estrazione testuale ci siamo basati sull'utilizzo del sistema di classificazione testuale denominato **term frequency - inverse document frequency** e posizionato i dati in maniera gerarchica su di una timeline. Nella fase di sperimentazione abbiamo utilizzato sia il dataset Enron, ma

anche i dati provenienti dai miei profili personali a causa della difficoltà di reperire dati adatti che permettessero di svolgere delle analisi adeguate.

Non possiamo che essere soddisfatti del risultato raggiunto, ma sappiamo che l'attuale progetto possiede ampi margini di miglioramento. Uno dei maggiori limiti di cui soffre la nostra applicazione è l'impossibilità di elaborare e analizzare contenuti multimediali da cui sarebbe possibile ricavare informazioni preziose aggiuntive a quelle già ricavabili tramite la nostra applicazione. Come conseguenza, la possibilità di analizzare anche contenuti multimediali allargherebbe sicuramente l'orizzonte di applicazioni analizzabili con la nostra applicazione. L'eventuale aggiunta di analisi di foto e video potrebbe permentte anche l'analisi di social network come Instagram o Tik Tok, ormai molto utilizzabili sopratutto tra i più giovani. La visualizzazione cartografica, inoltre, allo stato attuale, non funziona per le mail in quanto gli header non contengono informazioni posizionali, per questo potremmo suggerire in integrare nella nostra applicazione alcune librerie di language analysis che potrebbero estrarre dai contenuti delle email informazioni posizionali per poi rappresentare i dati su di una mappa. Un altra funzionalità interessante che si potrebbe integrare è quella di riconoscere vocaboli uguali. Ad esempio, il termine *social network* e il termine *SOCIAL NETWORK* sono valutati come termini differenti. Introdurre una procedura che confronti le parole e ne calcoli un valore di similarità per stabilire se due termini sono in realtà riconducibili ad un unico termine comune potrebbe offrire nuovi importanti sviluppi.

Appendici

Appendice A

Codice sorgente per l'applicazione

Il codice utilizzato per l'applicazione è completamente disponibile [12]. Tutto il server è scritto in python 2.7, mentre il client in javascript e fa uso di jQuery 3.4.1.
All'interno del percorso **/server/** è possibile trovare tutti gli script python utilizzati. In particolare in **/server/dumper** sono posizionati tutti gli script che gestiscono l'upload su Neo4j, mentre in **/server/dataSearcher/nlp** è possibilite trovare gli script che si occupano dell'analisi e dell'estrazione testuale.
Il percorso **/static/** raggruppa tutti i file javascript che gestiscono il client: in **/static/plugin/** sono posizionati il codice relativo ai vari framework utilizzati all'interno dell'applicazione, mentre **/static/visualization/** ospita i vari file dedicati ai quattro tipi di visualizzazione.

Creazione della connessione verso Neo4j

```
from py2neo import Graph
graph = Graph(
            host="address:port",
            auth=("user name", "password")
        )
```

Creazione del Facebook User Node tramite py2neo

```
from py2neo import Graph, Node
node = Node(
            'fbUser',
            name=config['profile']['name']['full_name'].encod
            ↪ e('latin1'),
            email=config['profile']['emails']['emails'],
            birthday=str(config['profile']['birthday']['year']) + '/'
            ↪ + str(config['profile']['birthday']['month']) + '/' +
            ↪ str(config['profile']['birthday']['day']),
            profile_uri=config['profile']['profile_uri'],
            graph_information=[usr, 'facebook'],

            ↪ userProfileProperty=config['profile']['emails']['emails'][0]
        )
        graph.create(node)
```

Creazione della matrice TFIDF

```
import nltk
from nltk.corpus import stopwords
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer

def removeStopWords():
    stop_words = set(stopwords.words('italian'))
    english_stop_words = set(stopwords.words('english'))
    # we have define an array of other stopwords used in the corpus
    othersWords = [...]
    stop_words.update(list(set(othersWords)))
```

```
    stop_words.update(list(set(english_stop_words)))
    return stop_words

def calculateTFIDF(corpus):
    stop_words = removeStopWords()
    tfidf = TfidfVectorizer(stop_words = stop_words)
    x = tfidf.fit_transform(corpus)
    df_tfidf = pd.DataFrame(x.toarray(), columns=tfidf.get_feature_names())
    return  {c: s[s > 0] for c, s in zip(df_tfidf, df_tfidf.T.values)}
```

Upload della lista amici di Facebook su Neo4j

```
def createFriendsNode(path, usr, userProfileProperty):
    file = path + '/' + 'friends/friends.json'

    if os.path.exists(file):
        friends = json.loads(open(file).read())
        for friend in friends['friends']:
            name = friend['name'].encode('latin1')
            timestamp = time.strftime('%Y/%m/%d %H:%M:%S',
            ↪  time.localtime(friend['timestamp']))

            node = Node(
                'Friend',
                name=name,
                timestamp=timestamp,
                graph_information=[usr, 'facebook'],
                userProfileProperty = userProfileProperty
            )
            graph.create(node)
```

```
graph.run(
    'MATCH (u:fbUser {graph_information: [\'' + usr +
    ↪  '\',\'facebook\']}) '
    'MATCH (f:Friend {graph_information: [\'' + usr +
    ↪  '\',\'facebook\']}) '
    'WHERE '
    ' u.userProfileProperty=\'' + userProfileProperty + '\'
    ↪  AND '
    ' f.userProfileProperty=\'' + userProfileProperty + '\' '
    'MERGE (u)-[r:FRIEND
    ↪  {relationship_type:[\'friend\']}]->(f) '
)
```

Elenco delle figure

Elenco delle tabelle

Bibliografia

[1] Charu C. Aggarwal. *An Introduction to Social Network Data Analytics*. 2011.

[2] Butler Analytics. Free and open source social network analysis software. `http://www.butleranalytics.com/20-free-and-open-source-social-network-analysis-software/`, 2015.

[3] Michael Baur, Ulrik Brandes, Jürgen Lerner, and Dorothea Wagner. *Group-Level Analysis and Visualization of Social Networks*. 01 2009.

[4] CMU William W. Cohen. Mld. enron email dataset. `https://www.cs.cmu.edu/~./enron/`.

[5] Sergio Decherchi, Simone Tacconi, Judith Redi, Alessio Leoncini, Fabio Sangiacomo, and Rodolfo Zunino. *Text Clustering for Digital Forensics Analysis*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[6] Vamshee Krishna Devendran, Hossain Shahriar, and Victor Clincy. *A Comparative Study of Email Forensic Tools*, volume 6. Journal of Information Security, 2015.

[7] Rachid Hadjidj, Mourad Debbabi, Hakim Lounis, Farkhund Iqbal, Adam Szporer, and Djamel Benredjem. Towards an integrated e-mail forensic analysis framework. *Digital Investigation*, 5(3):124 – 137, 2009.

[8] Ivan Heibi. A visual framework for graph and text analytics in email investigation. 2017.

[9] Facebook Inc. Facebook law enforcement guidelines. `https://www.facebook.com/safety/groups/law/guidelines/?_rdr`, 2021.

[10] Twitter Inc. Twitter law enforcement guidelines. https://help.twitter.com/it/rules-and-policies/twitter-law-enforcement-support#7, 2021.

[11] Json. Introducing json. https://www.json.org/json-en.html.

[12] Leonardo Lucente. Codice sorgente dell'applicativo per SNF. https://drive.google.com/drive/folders/1Dsn7SYXdB1byHN_lghHUmvznYBFtAxWU.

[13] Martin Mulazzani, Markus Huber, and Edgar Weippl. *Data Visualization for Social Network Forensics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[14] BBC News. Enron scandal at-a-glance. http://news.bbc.co.uk/2/hi/business/1780075.stm, 2002.

[15] Juan Ramos. *Using TF-IDF to determine word relevance in document queries*. 01 2003.

[16] Mithileysh Sathiyanarayanan and Nikolay Burlutski. *Visualizing Social Networks Using a Treemap Overlaid with a Graph*, volume 5. 2015. Second International Symposium on Computer Vision and the Internet (VisionNet'15).

[17] X. Shen, N. Nikolov, K. Xu, Y. Wu, X. Fu, and S. Hong. Visualization and analysis of email networks. pages 1–8, feb 2007.

[18] Michael Spranger and Dirk Labudde. *Semantic Tools for Forensics: Approaches in Forensic Text Analysis*. 11 2013.

[19] Visualware. Emailtrackerpro email tracer and spam filter. http://www.emailtrackerpro.com/.

[20] Fernanda B. Viégas, Scott Golder, and Judith Donath. Visualizing email content: Portraying relationships from conversational histories. pages 979–988, 2006.

[21] vound software. Vound product suite. https://www.vound-software.com/.

[22] Xplico. Network forensics analysis tool. https://www.xplico.org/.

[23] Yisleidy Linares Zaila and Danilo Montesi. Geographic information extraction, disambiguation and ranking techniques. 2015.