



UNIVERSITÀ DEGLI STUDI ROMA TRE

Dipartimento di Ingegneria

Corso di Laurea Magistrale in Ingegneria Informatica

Tesi di Laurea Magistrale

Social Network Analysis basata su Tecniche di Machine Learning

Laureando

Alessandro Di Leone

Matricola 472999

Relatore

Prof. Alessandro Micarelli

Correlatore

Prof. Giuseppe Sansonetti

Anno Accademico 2018/2019

*A mio padre per avermi spiegato cosa significa essere buoni e gentili, senza parlare.
Con i piccoli gesti di tutti i giorni.*

A me stesso. Per la caparbietà. Per le decisioni sbagliate. Per quelle che si riveleranno giuste. Per quello che ho imparato e per come lo sfrutterò.

*"È facile smarriti in una matassa di sentimenti
ma quanto costa rimanere fermi?
Mi è sempre piaciuto perdermi per poi scoprire nuove strade.*

*Scontrarsi non vuol dire morire
ma deviare verso altre direzioni."*

(Sassa)

Ringraziamenti

In queste poche righe intendo ringraziare le persone che mi hanno accompagnato in questo lavoro e nel corso di tutta la carriera universitaria. Per primo vorrei ringraziare sentitamente il Prof. Giuseppe Sansonetti che mi ha seguito, consigliato e supportato durante tutto il periodo di tesi, permettendomi di lavorare in un clima di assoluta serenità e allegria costante. Un grande ringraziamento va inoltre al Prof. Micarelli che mi ha dato l'opportunità di sviluppare questo progetto e sperimentarlo nel laboratorio di Intelligenza Artificiale.

Un grazie speciale ai miei genitori, vi devo tutto.

Un saluto particolare va ai miei amici Davide e Filippo per il loro sostegno durante le fatiche e lo sconforto che caratterizzano il nostro percorso ma soprattutto per i momenti di gioia e soddisfazione al raggiungimento del traguardo. Mi sento inoltre di ringraziare i miei amici Mattia ed Emma per essere semplicemente loro. Grazie a mio fratello, per avermi sopportato in tutti questi anni. Non ti libererai mai di me. Per ultima, ma non meno importante, voglio ringraziare sentitamente Leila, per aver riempito di lealtà e semplicità attimi in cui il mondo mi appariva caotico. Ad Maiora.

Introduzione

L'uomo è per sua natura un essere sociale, affermava Aristotele. La capacità di comunicare nasce dall'esigenza di condividere con i propri simili non solo informazioni utili per la sopravvivenza, ma anche emozioni, piaceri e interessi. Questa necessità di scambiare informazioni e scoperte in modo rapido ha dato il via a una rivoluzione che ebbe inizio con quello che ora tutti chiamano Web 1.0. Era l'Internet dei contenuti, i siti web erano semplici testi statici simili alle pagine di un libro o a fogli di word. Contenevano anche immagini o video, ma lo scopo di queste pagine era la mera consultazione, l'informazione, senza interazione fra utente e contenuto. Ma il Web non si è fermato e con lo sviluppo e l'evoluzione delle community online spinte dall'avanzata inesorabile dei social network si è giunti ad un alto grado di interattività che ha condotto a una concezione diversa del Web inteso sì come un luogo di aggregazione e condivisione, ma anche come un immenso database all'interno del quale tecniche di Intelligenza Artificiale e in particolare algoritmi di Machine Learning esistenti da tempo, scoprono la loro massima efficacia nel cercare di spiegare e a volte scoprire dal nulla fenomeni che si diffondono sottotraccia. In questo contesto hanno acquisito grande importanza problemi di Community Detection, Topic Modeling e Sentiment Analysis basati sui contenuti generati dagli utenti del Web. L'analisi delle reti sociali (Social Network Analysis, SNA) e l'elaborazione, e successiva modellazione, di complesse strutture di dati (BigData) hanno come obiettivo quello di analizzare le connessioni tra i diversi soggetti che compongono il network. L'aspetto rilevante è rappresentato, quindi, dalle relazioni che intercorrono tra i vari soggetti. La Community Detection rappresenta una specializzazione della SNA volta ad analizzare un gruppo di utenti che sono interessati a un determinato argomento, o a un approccio comune alla vita, a un brand, a un personaggio e così via. Una

volta identificate, però, queste comunità in base a diversi criteri è necessario identificare di cosa si occupano questi gruppi di utenti e soprattutto cosa pensano degli argomenti in questione. E' proprio a questo punto che entrano in gioco problematiche di Topic Modeling e Sentiment Analysis. In base a come interagiscono e sfruttando le informazioni che si scambiano gli utenti di una community è possibile applicare modelli matematici in grado di estrarre i *topic*, ovvero le tematiche specifiche trattate al fine di guidare processi di decision making e dare un forte contributo a processi di business di varia natura, dalla politica ai mercati azionari, dal marketing alla comunicazione, dall'ambito sportivo a quello delle scienze mediche e naturali. Inoltre, con la diffusione massiva dei social media gli utenti hanno cominciato a divulgare le proprie opinioni in rete, andando così a costituire una mole di dati considerata fondamentale per istituzioni, personaggi politici e brand a causa dell'incidenza che essa può avere sulla reputazione e sul grado di soddisfazione. L'estrazione e l'analisi dalle piattaforme dell'espressione dei giudizi degli utenti viene definita Sentiment Analysis: per una definizione più precisa, partendo dalla SNA, l'analisi computazionale di sentimenti ed opinioni espressi all'interno di testi generati in rete su un prodotto, un servizio, un individuo, un'organizzazione, un evento, e così via. L'opinione espressa ha un orientamento che indica se essa sia positiva, negativa o neutra. Oggi uno dei più popolari e rappresentativi servizi del Social Web è senza dubbio la piattaforma di microblogging Twitter¹. Gli utenti di questo servizio possono pubblicare dei brevi messaggi (tweet) nella piattaforma con i quali condividono informazioni, pensieri o attività.

Questa tesi ha l'obiettivo di illustrare algoritmi di Community Detection, Topic Modeling e Sentiment Analysis applicati proprio al social network Twitter. Tra le problematiche che si sono affrontate nel corso del lavoro di tesi vi sono innanzitutto questioni relative all'analisi e la comprensione delle modalità con cui combinare le diverse tipologie di interazioni messe nel social network Twitter in modo tale da rivelare la reale struttura sociale esistente. Infine, particolare attenzione è stata rivolta all'analisi comparativa e quindi nella valutazione di tecniche di individuazione di comunità online su social network e Sentiment Analysis dall'algoritmo proposto rispetto ad altri algoritmi che sono stati implementati e che sono noti in letteratura.

¹<https://twitter.com/>

Il presente elaborato di tesi è strutturato come segue. Nel *Capitolo 1* si analizza il contesto del Social Web e in particolare si approfondiscono le caratteristiche del social network Twitter, al fine di comprendere ed evidenziare l'ambito in cui si colloca il presente lavoro di tesi, ovvero la Social Network Analysis.

Il *Capitolo 2* ha l'obiettivo di introdurre e presentare il task relativo alla Community Detection tramite una panoramica relative alle tecniche maggiormente utilizzate in generale e testate nell'ambito di questo lavoro di tesi.

Il *Capitolo 3*, così come il precedente, si propone di illustrare il task relativo alla modellazione dei topic da estrarre dal dataset a disposizione.

Nel *Capitolo 4* si descrivono gli obiettivi relativi allo step di Opinion Mining, se ne descrivono gli approcci fondamentali e si introducono le metodologie menzionate nel corso della trattazione.

Nel *Capitolo 5* è illustrata l'architettura generale del processo e sono discusse le scelte prese per ogni step.

Il *Capitolo 6* è, infine, dedicato alla descrizione delle valutazioni sperimentali effettuate. Sono illustrati i risultati ottenuti dai test eseguiti al fine di valutare le varie tecniche e scegliere quelle più efficaci ed efficienti per gli scopi della ricerca.

Indice

Introduzione	iv
Indice	vii
Elenco delle figure	x
1 Social Network Analysis	1
1.1 L'evoluzione del Social Web: da 2.0 a 3.0	1
1.2 Twitter	3
1.2.1 Microblogging	3
1.2.2 Caratteristiche fondamentali	5
1.3 Modellazione di una Rete Sociale	6
1.3.1 Omofilia e Triadic Closure	7
1.3.2 Densitá e Coefficiente di Clustering	9
1.3.3 Centralitá	10
1.3.4 Ponti e Ponti Locali	10
1.3.5 Cascading Behavior e Diffusione delle Informazioni	12
2 Community Detection	13
2.1 Introduzione al problema	13
2.2 Clustering: Tecniche e Algoritmi	14
2.3 Tecniche di clustering su vettori	16
2.3.1 Word Embedding	16
2.3.2 K-means	18
2.3.3 DBSCAN	19

2.3.4	Clustering gerarchico	21
2.4	Tecniche di clustering su grafo	24
2.4.1	Affinity Propagation	25
2.4.2	Spectral Clustering	26
2.4.3	Louvain Method	27
2.4.4	Hyperlink Induced Topic Search (HITS)	29
3	Topic Modeling	31
3.1	Introduzione al problema	31
3.2	Algoritmi per Topic Modeling	32
3.2.1	Latent Semantic Analysis	32
3.2.2	Latent Dirichlet allocation	33
4	Opinion Mining	36
4.1	Introduzione al Problema	36
4.2	Approcci: Apprendimento automatico e dizionari	36
4.2.1	Tipologie	38
4.2.2	Sfide	39
4.3	Strumenti	40
4.3.1	SentiStrength	40
4.3.2	Reti neurali ricorrenti: Long Short-Term Memory	42
4.3.3	Support Vector Machine	45
5	Architettura del Sistema	49
5.1	Introduzione al caso di studio	49
5.2	Architettura della soluzione	50
5.2.1	Schema processo	51
5.2.2	Data Acquisition: Twitter API	54
5.2.3	Persistenza del dataset	55
5.2.4	Modellazione del Grafo Sociale	55
5.3	Analisi	57
5.3.1	Louvain Method per Community Detection	57

5.3.2 Authority Analysis	59
5.3.3 Topic Coherence	60
6 Risultati	63
6.1 Modellazione centrata su utente	63
6.1.1 Confronto fra algoritmi di clustering	65
6.2 Descrizione Dataset	68
6.2.1 Considerazioni quantitative	68
6.2.2 Considerazioni qualitative	69
6.3 Analisi opinion Trump	69
6.4 Polarizzazione utenti: Democratici Vs Repubblicani	71
6.5 Confronto algoritmi Opinion Mining	72
6.5.1 LSTM	73
6.5.2 SVM lineare	74
6.5.3 SVM non lineare	75
Conclusioni e sviluppi futuri	79
Conclusioni	79
Sviluppi futuri	80
Appendice	81
Strumenti software	81
Librerie	81
Ambiente	81
Tool	82
Bibliografia	83

Elenco delle figure

1.1	Grafo diretto e indiretto.	7
1.2	Formazione di un legame.	8
1.3	Bridge	10
1.4	Local bridge	11
2.1	Doc2Vec model	17
2.2	Dendrogramma	22
2.3	Agglomerative Hierarchical Clustering Technique	23
2.4	Affinity Propagation	26
2.5	Algoritmo Louvain: esempio di passo.	28
3.2	Esempio di applicazione di Latent Dirichlet Analysis.	34
3.3	Modello grafico LDA	34
4.1	Modello di rete neurale ricorrente.	43
4.2	Modello LSTM	44
4.3	Esempio di separazione lineare mediante le SVM.	46
5.1	Moduli del sistema.	50
5.2	Modellazione Mention.	56
5.3	Modellazione Retweet.	56
5.4	Modellazione Hashtag.	56
5.5	Comunità rilevate.	58
5.6	Valori di Authority e Hub.	59
5.7	Visualizzazione Authority.	60

5.8	Coherence Score.	61
5.9	Topic cluster 9.	61
5.10	Topic cluster 11.	62
6.1	Rete sociale centrata su un singolo utente (il cui nodo relativo è stato rimosso per una migliore visualizzazione).	64
6.2	Distribuzione tweet K-means con Word2Vec embedding.	65
6.3	Distribuzione tweet K-means con Doc2Vec embedding.	66
6.4	Risultati dell'Elbow Method.	67
6.5	Distribuzione tweet DBSCAN con Doc2Vec embedding.	67
6.6	Termini principali dei cluster identificati tramite DBSCAN.	68
6.7	Dimensioni del dataset.	68
6.8	Linguaggio dei tweet.	68
6.9	Distribuzione utenti rispetto a UGC.	69
6.10	I dieci hashtag più utilizzati nel periodo di osservazione.	70
6.11	Filtraggio tweet inerenti Trump.	70
6.12	Diagramma opinion inerente Trump.	71
6.13	Polarizzazione utenti Trump vs. Clinton.	72
6.14	Estratto del training set.	73
6.15	Topologia della Rete Neurale Ricorrente.	74
6.16	Risultati della sentiment analyis tramite LSTM.	75
6.17	Esempio di misclassificazione.	75
6.18	Risultati della sentiment analyis tramite linear SVM.	76
6.19	Classificazione corretta con entità multiple.	76
6.20	Risultato della k-fold cross validation.	77
6.21	Heatmap CVGrid.	77
6.22	Risultati della sentiment analyis tramite SVM con kernel gaussiano.	78

Capitolo 1

Social Network Analysis

1.1 L'evoluzione del Social Web: da 2.0 a 3.0

Tra il 2004 e il 2005, un grande editore americano, "O'Reilly Media"¹, organizzò una serie di conferenze negli U.S.A. per spiegare le nuove opportunità che la Rete Internet e il Web in particolare, metteva a disposizione degli utenti. In questi incontri venne coniato e ufficializzato il concetto di "Web 2.0".²

Il termine Web 2.0 rappresenta uno slogan che identifica un grande mutamento di paradigma nel Web: gli utenti, da fruitori passivi di informazioni digitali, divengono essi stessi produttori e consumatori di contenuti e conoscenza in rete (consumer+producer: "prosumer").

Vennero descritti e pubblicizzati i nuovi modi della comunicazione digitale e gli innovativi servizi di cui potevano usufruire anche gli utenti comuni, "non esperti di informatica". Durante questi convegni, si evidenziò la novità della partecipazione attiva dell'utente nella fase comunicativa e la gestione autonoma dei contenuti multimediali.

Il Web 2.0 quindi, evidenziando la sua duttilità e semplicità d'uso, offre all'utente comune, non esperto, l'opportunità della libera espressione dell'individuo, che può: pubblicizzare la sua persona, la sua professione o attività e/o la sua competenza per un argomento specifico, in maniera semplice, veloce e gratuita.

¹<https://storiadiinternet.wordpress.com/twitter> (Last access: 09/09/2019)

²<https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html> (Last access: 11/09/2019)

Ne scaturisce una forte interazione tra utenti e si sviluppano concezioni di comunicazione nuova : comunicazione uno-a-molti e molti-a-molti, rispettivamente i cosiddetti *blog* e *social media*.

Nascono dunque servizi online in grado di consentire la creazione, il caricamento, la condivisione e lo scambio di contenuti informativi di vario tipo (*User Generated Content, UGC*) all'interno di reti e comunità virtuali.

Due categorie principali di social media:

- Social Networking Site, piú comunemente definiti semplicemente Social Network, sono servizi online che permettono di comunicare e interagire con la propria "rete sociale", esempi ne sono Facebook e LinkedIn.
- Content Sharing Site, servizi online che consentono di pubblicare in rete e di condividere con altri utenti materiali di varia natura, esempi ne sono Flickr, Youtube, Instagram e Twitter.

A queste due milestones, Web 1.0 e 2.0 se ne aggiunge ormai una terza, che forse è logica conseguenza delle precedenti e che ha l'autorità per prendere l'etichetta di web 3.0. Siamo passati dall'abbondanza di informazioni, per arrivare all'abbondanza o forse sovrabbondanza delle interazioni: da qui il tema dei big data e quello degli analytics che necessariamente li accompagna.

Il termine Web 3.0 è ancora parzialmente non definito in maniera formale. La chiave di lettura per la comprensione del termine però è la seguente: sicuramente, vuol dire andare al di là dell'aspetto relazionale che contraddistingue il 2.0 e fare riferimento non piú alle persone che comunicano tra loro. E piú in là delle persone ci sono le macchine, ovvero l'intelligenza artificiale.

Si tratta di una trasformazione del World Wide Web in un ambiente dove i documenti pubblicati sono associati ad informazioni e metadati che ne specificano il contesto semantico in un formato adatto all'interrogazione e l'interpretazione (es. tramite motori di ricerca) e, piú in generale, all'elaborazione automatica. In pratica, un Web in cui le macchine non solo leggono, ma interpretano.

Tutto questo è reso possibile da una disponibilità di dati e potenza computazionale sempre crescente che sta aumentando in modo straordinario, a mano a mano che la

tecnologia penetra sempre piú nei comportamenti sociali. Tutte le interazioni generate nella giornata sono una fonte di dati enorme su un generico cliente e se si stima che ogni anno i dati prodotti dagli utenti siano superiori alla somma di tutti i dati prodotti in precedenza é facile capire che siamo di fronte ad una nuova era.

Il contributo fondamentale é dato dall'*Internet of Things*, con l'inserimento di sensori in ogni possibile device e con il conseguente aumento delle interazioni con i clienti che contribuisce alla generazione di miliardi di dati di uscita. In un mondo completamente digitalizzato, ogni consumatore lascia dietro di sé un *digital wake* che aumenta in modo esponenziale rispetto all'aumentare dei *digital touchpoints*. Siamo di fronte ad un punto di flesso e secondo McKinsey, questo flesso sarà in grado di aumentare in modo drammatico la produttività in tutti i settori, dal pubblico al privato. È proprio qui che entrano in gioco l'Intelligenza Artificiale e gli algoritmi di Machine Learning.

1.2 Twitter

Twitter é un servizio di rete sociale e microblogging che fornisce agli iscritti (twitterer), una pagina personale su cui poter pubblicare messaggi di testo di lunghezza massima prefissata (140 caratteri) chiamati tweet. Oggi Twitter con piú di 190 milioni di utenti e oltre 65 milioni di messaggi postati ogni giorno é sicuramente uno dei servizi piú popolari nel Social Web.

All'inizio non fu progettato per essere un social network. Il suo ideatore Jack Dorsey lo pensó per farlo funzionare come piattaforma di comunicazione per dispositivi mobile. Col passare del tempo, però, la sua sopravvivenza dipese sempre piú dagli adattamenti informatici dei suoi programmati, che per farlo diventare piú competitivo tra i vari sistemi di comunicazione, lo fecero diventare un vero e proprio social network.

La diffusione di Twitter fu vertiginosa: passó da 105 milioni di utenti dell'aprile 2011 ai 200 milioni alla fine dello stesso anno.

1.2.1 Microblogging

Con l'evoluzione di Internet durante gli anni '90, i blog sono diventati un mezzo popolare per condividere avvenimenti, novità e altre informazioni con il resto del mondo. Il blog,

nel suo formato originario e celebre, costituisce il precursore del microblogging, il quale propone contenuti molto piú compresi rispetto ai consueti blog.

Il termine microblogging (conosciuto anche come micro-blogging o micro blogging) definisce una combinazione di blog e messaggistica istantanea: questo ibrido permette agli utenti di realizzare comunicazioni o aggiornamenti degni di nota per poi condividerli online con un pubblico sotto forma di brevi informazioni.

Diversamente dai blog tradizionali, il microblogging é caratterizzato dalla condensazione e dalla diffusione di informazioni in microformato. Nello specifico, Twitter, la piattaforma di social media, si é evoluta a tal punto da diventare uno dei servizi piú popolari di questa nuova forma del blogging. Non sempre é facile riuscire a esprimere il riassunto di concetti complessi stando nel limite di 144 caratteri, come é il caso dei post di Twitter, ma i vantaggi sono evidenti: grazie al nuovo metodo di scambio di informazioni cosí breve e immediato, comunicare simultaneamente con una moltitudine di persone diventa molto piú semplice. Ma soprattutto per quanto riguarda i possessori di smartphone, é una vera comoditá potersi informare tramite questi brevi messaggi compatti invece di scorrere a fatica lunghi e complessi siti web da un piccolo touchscreen.

I messaggi brevi, come lo sono ad esempio i tanto amati tweet, possono essere pubblicati e diffusi in una molteplicitá di formati. Ció comprende sia gli attuali formati di testo e di immagine, come i video, i messaggi vocali e gli iperlink.

Di base il microblogging ricorda messaggi brevi come una volta erano diffusi soprattutto sotto forma di SMS, che ora vengono invece utilizzati per la comunicazione di massa. In questo modo é possibile comunicare in maniera semplice e veloce con i follower tramite il microblogging, informandoli online su notizie attuali.

Nel frattempo il microblogging é diventato un mezzo di comunicazione fondamentale anche per le scuole superiori e gli enti di formazione. Tramite rispettive app i ricercatori possono discutere a livello internazionale su temi di interesse comune oppure fornire informazioni su nuove scoperte. Cosí si puó catturare l'attenzione sui propri progetti e risvegliare l'interesse su scala mondiale di circoli di esperti.

1.2.2 Caratteristiche fondamentali

La particolarità unica di Twitter, rispetto agli altri social network, è che non ci sono cerchie o gruppi di amici, ma esistono due categorie di utenti a cui riferirsi, i *Follower* e i *Following*:

- I follower (seguaci) di un utente sono coloro i quali sono interessati a ricevere tutti i tweet e aggiornamenti dell'utente seguito.
- I following (letteralmente seguiti), invece, sono gli utenti che si è scelto di seguire in qualità di follower (colleghi, amici, personaggi famosi, Vip, aziende, gruppi musicali ecc.).

Con la funzione *Retweet*, si possono riproporre i tweet che si ricevono ai propri follower, che leggeranno il messaggio come se fosse personale, anche se viene specificato l'autore originale del "cinguettio".

Poi ci sono le *Liste* che permettono di selezionare e raggruppare, secondo categorie personali, i follower/following.

Altre funzionalità del network sono le sezioni *Scopri*, *Notifiche* e *Messaggi*: La funzione "Scopri" permette di scoprire le attività quotidiane dei following e aiuta a trovare nuovi utenti. La sezione "Notifiche" elenca tutte le "interazioni": dalle conversazioni con altri utenti, ai tweet in cui l'utente è stato menzionato, compresi i Retweet e i nuovi follower. I "Messaggi" o "Direct Messages (D.M.)" sono una specie di posta elettronica privata del social network.

Un hashtag³, rappresentato dal simbolo #, viene utilizzato per indicare parole chiave o argomenti. Una qualunque espressione alfanumerica preceduta dal simbolo # che forma un tutt'uno con esso. Non può contenere spazi, segni di punteggiatura, segni speciali, trattini ecc..

Questa funzione è nata su Twitter e permette agli utenti di seguire facilmente gli argomenti a cui sono interessati.

Gli utenti utilizzano il simbolo hashtag (#) nei Tweet prima di una parola chiave o frase pertinente per categorizzarli e permettere agli altri di vederli in tutta facilità nella

³<https://help.twitter.com> (Last access: 11/09/2019)

ricerca di Twitter. Cliccando o toccando una parola sotto forma di hashtag presente in qualsiasi messaggio verranno mostrati gli altri Tweet in cui è presente tale hashtag. Gli hashtag possono essere inseriti ovunque in un Tweet. Le parole sotto forma di hashtag che diventano molto popolari sono spesso *Argomenti di tendenza*.

Il vantaggio principale è l'immediatezza: con una sola parola si è in grado di sintetizzare l'argomento di cui si sta parlando, coinvolgendo in tempo reale più interlocutori nella conversazione.

Un hashtag da informazioni immediate sul tono di un post, dichiarando se l'intento dell'utente è quello di informare, denunciare o protestare. Funziona come "punto d'incontro in cui ritrovarsi" per chi twitta e per chi cerca i tweet con lo stesso hashtag.

Ricorrendo all'uso dell'hashtag nella "Barra di Ricerca", si possono ricercare tutti i tweet che riguardano un particolare argomento.

La *Ricerca avanzata* consente di effettuare una ricerca più mirata, permette di cercare i tweet con molta più precisione rispetto alla "ricerca generica", in cui bisogna cercare tra migliaia e migliaia di tweet che vengono inviati contemporaneamente. Per esempio si possono cercare i tweet che contengono una frase esatta; cercare tweet in una lingua particolare o i tweet di un determinato utente e così via. Si può anche geolocalizzare la ricerca, ovvero cercare i tweet inviati da una specifica località.

1.3 Modellazione di una Rete Sociale

All'interno di un processo di analisi di una rete sociale come ad esempio un Social Network, risulta immediata l'importanza della concettualizzazione di una rete sociale come un grafo.

La Teoria dei Grafi fornisce un linguaggio unificante per descrivere la struttura di una rete.

Un grafo consiste di un insieme di nodi (vertici) e un insieme di archi (collegamenti). Un arco collega due nodi. Se due nodi sono vicini(neighbors) essi sono collegati da un arco (edge). Gli archi possono avere un orientamento. Archi diretti: la relazione vale solo tra testa e coda dell'arco. Archi indiretti (edge): la relazione vale in tutte e due le direzioni.

Grafi diretti e indiretti codificano diversi tipi di reti.

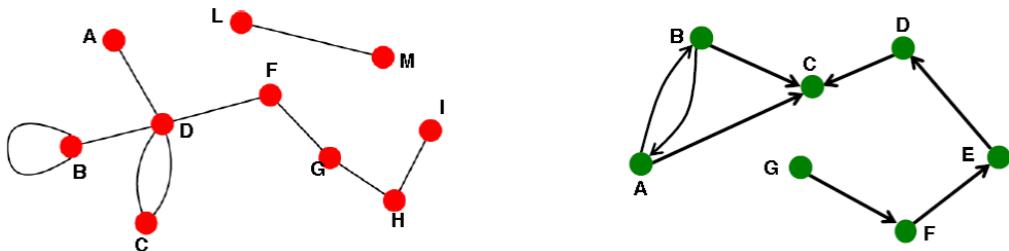


Figura 1.1: Grafo diretto e indiretto.

- Grafi indiretti, la relazione tra due nodi è frutto di un'azione concordata tra i due nodi. Ad esempio amicizia, alleanza, conoscenza, connessione, ecc...
- Grafi diretti, la relazione tra due nodi è frutto di un atto unilaterale. Ad esempio link a pagine web, followers o anche citazioni di un articolo.

I network reali sono grafi sparsi. Nello studio delle reti i nodi e gli archi rappresentano entità del mondo reale.

Alcune astrazioni di network sono comunemente utilizzate: ad esempio relazioni di amicizia su Facebook sono rappresentate con un grafo indiretto e non pesato, al contrario un tabulato di conversazioni telefoniche potrebbe essere rappresentato con un grafo pesato e diretto, in cui la direzione indica il chiamante e il chiamato mentre il peso la durata della chiamata o i dati scambiati.

1.3.1 Omofilia e Triadic Closure

In Twitter la relazione di following tra due twitterer non deve essere necessariamente reciproca (ad esempio account che presentano un elevato numero di follower e che quindi sono riconducibili a celebrità sono caratterizzati da un numero trascurabile di followed). Una relazione reciproca evidenzia, quindi, un legame maggiore, dettato magari da alcuni fattori specifici.

L'*Omofilia* è un fenomeno riscontrabile nelle reti sociali, in cui le persone tendono a instaurare relazioni con gli individui della rete con cui condividono delle caratteristiche socio-demografiche, comportamentali o interpersonali. La conseguenza fondamentale di tale fenomeno è che le relazioni tra persone simili sono più frequenti rispetto a quelle tra

persone dissimili. In Twitter l'omofilia implica che l'utente che instaura una relazione di following con un altro sia interessato agli argomenti pubblicati da quello specifico utente.

Partendo da questo presupposto, è possibile quindi cercare di definire una legge comune che governa la formazione dei legami all'interno di una rete sociale, virtuale o meno. La cosiddetta *Triadic Closure*.

Questa teoria si basa su un principio fondamentale: Se due persone in una rete sociale hanno un amico in comune, allora vi è una maggiore probabilità che essi diventeranno essi stessi amici ad un certo momento futuro.

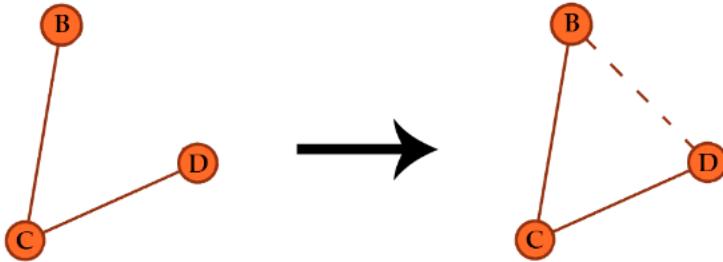


Figura 1.2: Formazione di un legame.

Se A e B hanno un amico C in comune, allora:

- è più probabile che A e B si incontrino (dal momento che entrambi trascorrono del tempo con C).
- A e B tendono ad avere fiducia l'un l'altro (dal momento che hanno un amico in comune).
- C ha un incentivo a far incontrare A e B (è più difficile per C mantenere due rapporti disgiunti).

Se si osserva l'evolversi di una rete sociale nel tempo si noterà la creazione di nuovi archi nel tempo e la tendenza alla chiusura dei triangoli.

1.3.2 Densitá e Coefficiente di Clustering

Analizzare le componenti del grafo fornisce informazioni globali sulla struttura della rete: Cosa lega ogni componente? Come si diffondono le informazioni? Che ruolo svolgono i nodi?

A tal proposito tre indici fondamentali nell'analisi e detection di comunità virtuali sono la *Densitá*, il *Coefficiente di clustering* e la *Centralitá*, oltre alla teoria dell'*Information Cascade*.

La densitá in una rete descrive il livello generale dei legami fra i punti in un grafo. Piú sono numerosi i nodi direttamente collegati fra loro piú un grafo é denso. La densitá di un grafo si calcola come rapporto tra il numero delle linee di un grafo e il numero possibile di linee tra i nodi.

L'inclusività misura la percentuale di soggetti coinvolti nei legami o gli scambi del gruppo ed é calcolata come numero totale di punti in un grafo meno il numero di punti isolati. Qui diventa fondamentale il concetto di *Coefficiente di Clustering*.

Il coefficiente di clustering indica quanto i vicini di un dato nodo sono connessi tra loro. Per un nodo i di grado k_i il coefficiente di clustering locale é definito come:

$$C_i = \frac{2L_i}{k_i * (k_i - 1)} \quad (1.1)$$

dove L_i é il numero di edge esistenti tra i vicini del nodo i . C_i rappresenta la probabilitá che due vicini di i scelti a caso siano connessi da edge. In altre parole, piú il vicinato di i é densamente interconnesso, maggiore é il suo coefficiente di clustering locale.

Il coefficiente di clustering di un'intera rete viene catturato dal coefficiente di clustering medio:

$$\langle C \rangle = \frac{1}{N} \sum_1^N C_i \quad (1.2)$$

che rappresenta la probabilitá che selezionando casualmente due vicini di un qualche nodo, questi abbiano un edge che li connette.

1.3.3 Centralità

Un attore é quindi tanto piú centrale nella rete quanto piú é nella posizione di interagire velocemente con gli altri attori.

Esistono due tipi di centralitá:

- Centralitá locale: se un punto ha un gran numero di connessioni con altri punti del suo ambiente circostante.
- Globalmente centrale: se ha una posizione d'importanza strategica nella struttura complessiva della rete.

La misura piú semplice della centralitá si ottiene dal calcolo dei gradi, ovvero un nodo é centrale se ha un grado elevato cioé é adiacente a tutti gli altri nodi. Un soggetto con il grado piú alto rappresenta metaforicamente il luogo nel gruppo dove "le cose accadono". In contrasto, i soggetti con un basso grado rappresentano le posizioni periferiche nella rete.

1.3.4 Ponti e Ponti Locali

I nodi che si collocano in una posizione di intermediari (cioé localizzati sui percorsi che collegano coppie di nodi non adiacenti) possono esercitare un potere di controllo sul flusso delle informazioni.

Si definisce un come un *ponte*(bridge) un arco che connette due nodi A e B in un grafo la cui assenza determina una separazione netta, con A e B facenti parte di due componenti completamente distinte. In altre parole, questo arco rappresenta l'unico percorso possibile tra i suoi endpoint, i nodi A e B.

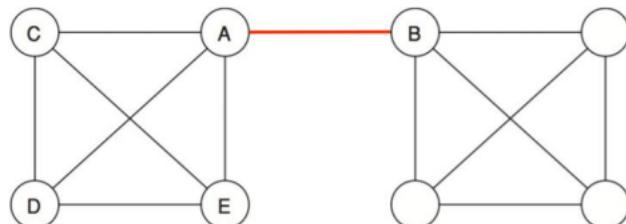


Figura 1.3: Bridge

Si definisce un come un *ponte locale* (local bridge) un arco che unisce due nodi, A e B, che non hanno amici in comune. In altre parole, cancellando l'arco in questione si allungherebbe la distanza tra A e B di un valore strettamente maggiore di due.

Importante e da sottolineare inoltre la connessione implicita con l'idea di Triadic Closure esposta in precedenza. I due concetti sono in completa opposizione: un arco é un local bridge precisamente quando non costituisce in alcun modo un lato di un qualsiasi triangolo presente nel grafo.

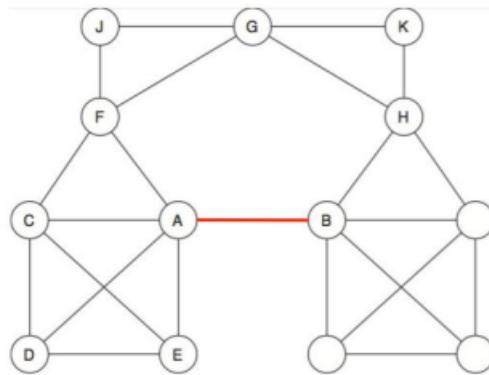


Figura 1.4: Local bridge

Questo a livello di contenuto informativo ha un valore molto elevato: paradossalmente infatti, conoscenza nuova e opportunità sociali hanno molta più probabilitá di fruire all'interno di un local bridge invece che in un arco normale. Questo é determinato dal fatto che una connessione singola con un nodo tramite ponte locale, provoca una relazione con un gruppo di entitá fondamentalmente diverse da quella in cui é immerso il nodo di partenza. Per effettuare un paragone infatti la cerchia di persone all'interno della quale un individuo si trova é caratterizzata da entitá amiche, aventi una gran voglia di aiutare, ad esempio in una ricerca di informazioni. Il problema sta nel fatto che per il concetto di omofilia, gli amici sono sostanzialmente delle persone simili all'individuo di partenza e quindi non in grado di apportare nuova informazione utile oltre a quella che già si possiede.

1.3.5 Cascading Behavior e Diffusione delle Informazioni

All'interno di una rete sociale, la formazione di nuovi legami è strettamente legata anche alle motivazioni secondo cui si sviluppano determinati rapporti.

Le dipendenze in una rete sociale sono generalmente basate su due aspetti fondamentali:

- La struttura della rete sottostante, studiata tramite la teoria dei grafi.
- Un'interdipendenza nel comportamento degli individui presenti all'interno dello stesso sistema, dove il comportamento di ciascun individuo viene fortemente influenzato dalla combinazione dei comportamenti degli altri componenti della cerchia (teoria dei giochi).

Esistono infatti vari scenari dove potrebbe essere razionale per un individuo imitare scelte effettuate da altri. Due ragioni distinte possono essere identificate chiaramente: Per effetto dell'informazione, dove le scelte effettuate da altri possono essere giustificate dall'apporto di conoscenza che non si possiede, ma che chi ha scelto si suppone abbia. E benefici diretti, dove si ottiene vantaggio nell'adottare soluzioni già utilizzate dai propri vicini.

Poiché quindi le decisioni di una particolare entità all'interno di una rete sociale possono essere basate su cosa hanno deciso di adottare i propri vicini, un pattern particolare emerge in merito alla formazione dei legami della rete: una decisione viene presa quando una certa frazione di neighbors ha scelto la stessa opzione.

Questo ha un'implicazione molto forte in merito alla centralità di un nodo, ma soprattutto in merito all'autorità di cui gode un particolare nodo nell'influenzare scelte altrui. Scegliendo infatti con cura elementi particolarmente autorevoli, i cosiddetti *influencer*, all'interno di una rete sociale, si possono generare reazioni a cascata molto importanti da sfruttare in molti processi di business, dal marketing fino alla politica.

Capitolo 2

Community Detection

Il rilevamento delle comunità é la chiave per comprendere la struttura di reti complesse e, in ultima analisi, estrarre informazioni utili da esse. Le applicazioni sono diverse: dalla sanitá alla geografia, dalle interazioni umane alla mobilitá fino anche all'economia.

Gli ambienti di tagging collaborativo come Twitter sono ricchi di interazioni sociali implicite. Analizzando i tag che gli utenti assegnano o ricercano, osservando i collegamenti tra entitá, ma anche analizzando i contenuti generati dagli stessi é possibile scoprire gruppi di utenti con interessi correlati.

2.1 Introduzione al problema

Il rilevamento delle comunità nelle reti, chiamato anche graph clustering, é un problema ad oggi non definito in maniera completa. Non esiste ancora una definizione universale dell'obiettivo da perseguire. Costituisce infatti un problema inherentemente difficile e non completamente risolto, nonostante il grande sforzo della comunità scientifica. Basti pensare che una definizione rigorosa di comunità sociale data da Fortunato [FH16], indica la comunità come un sottoinsieme della rete, i cui utenti sono tutti amici tra loro. In termini matematici e nella teoria dei grafi questo equivale ad una clique (cricca) e il problema associato é un problema NP-Completo.

Di conseguenza, non esistono linee guida chiare su come valutare le prestazioni di diversi algoritmi e su come confrontarle tra loro. Da un lato, tale ambiguitá lascia molto spazio alla possibilità di proporre diversi approcci al problema, che spesso dipendono

dalla specifica domanda di ricerca e (o) dal particolare caso di studio. D'altra parte, ha introdotto molto rumore nel campo, rallentandone i progressi.

Per comunità online si intende un gruppo di entità (i.e., utenti, organizzazioni, pagine web, ...) che interagiscono in un ambiente online e condividono obiettivi comuni, caratteristiche, o interessi. Esempi ne sono una comunità di fan di baseball, oppure una comunità di appassionati di fotografia digitale. L'appartenenza di un utente alla comunità è implicita e i modi per farne parte sono molteplici (e.g., postare su blog, newsgroup, forum; inviare instant message (e.g., tweet) o email ad altri membri della comunità; comprare e vendere online item relativi al topic). Inoltre un utente può avere più interessi e quindi essere membro di più di una comunità online.

Gli algoritmi dedicati alla Community Detection, hanno in comune l'intento di massimizzare una qualche funzione che misura la bontà della partizione del grafo identificata e classificata come comunità. I metodi per identificare le community online possono essere suddivisi in 4 categorie (non esclusive):

- Community centrata sui nodi: ogni nodo in un gruppo soddisfa determinate proprietà.
- Community incentrata sui gruppi: considera le connessioni all'interno di un gruppo nel suo complesso. Il gruppo deve soddisfare determinate proprietà senza che l'analisi debba necessariamente approfondirsi a livello del singolo nodo.
- Community incentrata sulla rete: partiziona l'intera rete in diversi set disgiunti.
- Community gerarchica: segmentata in base ad una definita struttura gerarchica che caratterizza il network.

2.2 Clustering: Tecniche e Algoritmi

Il processo di raggruppamento di un insieme di oggetti fisici o astratti in classi di oggetti simili è denominato *clustering*. un cluster è una collezione di oggetti che sono simili l'uno all'altro e sono dissimili dagli oggetti di altri cluster. un cluster di oggetti può essere trattato collettivamente come un gruppo in molte applicazioni.

Il clustering é un esempio di learning non supervisionato a differenza della classificazione (che é una forma di learning supervisionato), il clustering infatti non si basa su classi predefinite e su campioni di training etichettati.

Esiste un gran numero di algoritmi di clustering. La scelta dell'algoritmo da utilizzare in un dato contesto dipende dal tipo di dati disponibili, dal particolare scopo e dall'applicazione.

In generale,i principali metodi di clustering possono essere classificati come di seguito specificato:

- *Metodi di partizionamento*, dato un database di oggetti o tuple di dati, un metodo di partizionamento costruisce k partizioni dei dati, dove ciascuna partizione rappresenta un cluster, e $k \leq n$. In altre parole, l'algoritmo classifica i dati in k gruppi che, nel loro insieme, soddisfano i seguenti requisiti: (1) ciascun gruppo deve contenere almeno un oggetto, e (2) ciascun oggetto deve appartenere esattamente ad un gruppo.
- *Metodo gerarchico*, crea una decomposizione gerarchica di un dato insieme di oggetti. I metodi gerarchici possono essere classificati in agglomerativi o divisivi, basandosi su come viene effettuata la decomposizione gerarchica.
- *Metodi basati sulla densitá*, molti metodi di partizionamento clusterizzano gli oggetti basandosi sulla loro distanza. Tali metodi possono trovare solo cluster a forma sferica e incontrare difficoltà nell'individuare cluster di forma arbitraria. Sono stati sviluppati altri metodi di clustering basandosi sulla nozione di densitá. La loro idea generale é quella di far crescere un dato cluster fino a quando la densitá (numero di oggetti o punti di dati) in un vicinato non eccede una determinata soglia; in altre parole, é necessario garantire che, per ciascun punto all'interno di un dato cluster, il vicinato di un determinato raggio debba contenere almeno un numero minimo di punti. Tale metodo puó essere usato per filtrare rumore e scoprire cluster di forma arbitraria.
- *Metodi basati sulla griglia*, metodi basati sulla griglia quantizzano lo spazio degli oggetti in un numero finito di celle che formano una struttura a griglia. Tutte le operazioni di clustering vengono, quindi, eseguite sullo spazio quantizzato.

- Metodi basati sul modello, i metodi basati sul modello ipotizzano un modello per ciascuno dei cluster e trovano la migliore disposizione dei dati rispetto al determinato modello.

2.3 Tecniche di clustering su vettori

2.3.1 Word Embedding

L'importanza di metodi automatici per la classificazione ed estrazione di informazioni da testi è cresciuta significativamente negli ultimi anni, a causa della produzione sempre maggiore di questo tipo di dati, specialmente tramite piattaforme web. Questo ha portato allo sviluppo di nuovi algoritmi per analizzare testi non strutturati.

Le tecniche di "Embedding", che associano parole o parti di testo di lunghezza variabile a vettori di dimensione fissa mantenendo relazioni di similarità semantica, sono state un grande progresso per il campo del "Natural Language Processing". Sostanzialmente accade infatti che venga preso in input un corpus di documenti e venga poi costruito preliminarmente un vocabolario dal training set. Quindi poi viene appresa la rappresentazione vettoriale delle parole. I vettori risultanti a questo punto possono essere utilizzati come input per svariati algoritmi di natural language processing e machine learning. Inoltre, avanzamenti nelle tecniche di Deep Learning hanno migliorato significativamente la classificazione del testo, grazie agli affinamenti delle architetture delle reti neurali ricorrenti, in grado di processare sequenze di dimensioni variabili.

Questo approccio ha guadagnato una grande popolarità con l'introduzione della tecnica Word2Vec nel 2013. Doc2Vec può essere considerato un'estensione di Word2Vec il cui obiettivo è quello di creare una rappresentazione vettoriale di un documento, invece che di una singola parola.

Word2Vec e Doc2Vec sono implementati in varie librerie. Famose sono le librerie gensim per Python, la versione sviluppata da Google all'interno di Tensorflow e Spark all'interno della libreria per il machine learning MLlib.

Gli algoritmi più popolari per creare rappresentazioni Word2Vec sono *Skip-Gram model* e *Continuous Bag-of-Words model* (CBOW):

- Skip-Gram model, il task della rete neurale consiste in: Data una parola all'interno di una frase, la rete predirà quanto è probabile che ogni parola nel vocabolario sia la parola vicina della parola di input.

Il modello Bag-of-Words continuo (CBOW) è esattamente l'opposto di Skip-Gram.

- Continuous Bag of Words, il compito della semplice rete neurale è: Dato un contesto di parole (che circonda una parola) in una frase, la rete predirà quanto è probabile che ogni parola nel vocabolario sia effettivamente la parola appartenente a quella frase dato il contesto.

Rispetto a Word2Vec, in Doc2Vec viene semplicemente aggiunto un altro vettore all'input contenente l' "id del paragrafo".

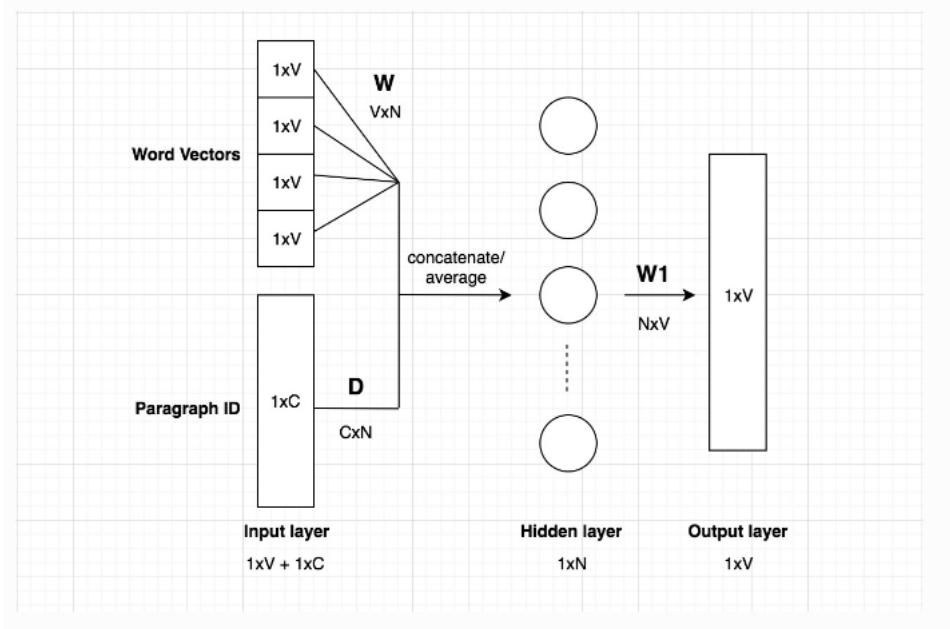


Figura 2.1: Doc2Vec model

In modo tale da tener conto anche del contesto all'interno quella determinata parola opera. un esempio banale di ciò potrebbe essere il caso in cui una stessa parola venga utilizzata in situazioni differenti con significati differenti.

2.3.2 K-means

L'algoritmo K-means é probabilmente la tecnica di clusterizzazione piú famosa (ed un caso particolare dell'algoritmo di Lloyd). Viene utilizzato in moltissimi campi, come la computer vision e la geostatistica, sia per la sua semplicitá di implementazione che per le elevate performance che lo caratterizzano.

É un algoritmo di tipo iterativo che raffina la suddivisione degli oggetti a ogni ciclo e che si basa sul concetto di centroide. un centroide é un punto nello spazio che rappresenta, sostanzialmente, un cluster e che corrisponde al punto medio dei punti del cluster stesso : Determina k centroidi che utilizza allo scopo di definire i cluster. una determinata osservazione é considerata appartenente ad un determinato gruppo se essa in base ad una determinata metrica di similaritá risulta essere piú vicina al centroide di un determinato cluster rispetto ai centroidi degli altri gruppi di osservazioni.¹

K-Means é costituito da un processo iterativo secondo cui si alternano due fasi principali:

- si assegnano i data point ai cluster in base ai centroidi stabiliti.
- si definiscono nuovamente i centroidi sulla base della corrente ripartizione dei data point ai cluster.

La scelta dei centroidi iniziali è un fattore che influenza molto il risultato finale, in quanto il K-means non garantisce il raggiungimento dell'ottimo globale (il miglior risultato possibile), ma puó attestarsi in un punto di ottimo locale, ossia la configurazione di cluster migliore che gli é possibile raggiungere date le condizioni iniziali.

Nell'ambito della ricerca atta a migliorare le performance dell'algoritmo K-means, nel 2007 David Arthur e Sergei Vassilvitskii idearono un metodo di inizializzazione che chiamarono K-means++ [DA07]. L'idea su cui si basa la loro tecnica consiste nel costruire un set di centroidi iniziali che siano il piú "sparpagliati" possibile.

¹<https://stanford.edu/~cziech/cs221/handouts/kmeans.html>

ALGORITMO K-MEANS++ (PSEUDOCODICE):

Input:

- un insieme I di N oggetti da clusterizzare.
- un intero C che rappresenta il numero di cluster desiderati.

Repeat:

- Step 1: Scegliere randomicamente un oggetto in I ed eleggerlo a centroide. Repeat $C - 1$ times:
- Step 2: Per ogni punto i dell'insieme I calcolare la distanza $D(i)$ dal centroide piú vicino tra quelli costruiti fino a questo momento.
- Step 3: Scegliere un punto j in I con una probabilitá che sia proporzionale al quadrato della distanza $D(j)$ ed eleggerlo a centroide.

Questa procedura di inizializzazione ha dato risultati sorprendenti, dimostrando, in alcuni casi, di saper aumentare notevolmente le performance dell'algoritmo K-means, anche di svariati ordini di grandezza, oltre a raggiungere il risultato finale molto piú velocemente.

Inoltre di fondamentale importanza é la scelta del numero k di cui inizializzare l'algoritmo. Uno dei metodi piú immediati é quello dell'elbow method, ovvero un metodo grafico che considera la variazione totale dello scarto quadratico d'errore interno (WSSE), come funzione del numero di cluster ottenibili. Esso si basa sull'idea che il numero k di cluster scelti dovrebbe essere tale che l'aggiunta di un ulteriore cluster non fornisca una maggiore modellazione dei dati e dunque non migliori il WSSE.

2.3.3 DBSCAN

Il DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) é un metodo di clustering proposto nel 1996 da Martin Ester *et al.*. È basato sulla densitá perché connette regioni di punti con densitá sufficientemente alta [ME96].

DBSCAN usa una definizione di cluster basata sulla nozione di *density-reachability*. Un punto q è direttamente raggiungibile da un punto p se la loro distanza è minore di un assegnato ε (cioé, è parte del suo ε -vicinato) e se p è circondato da un sufficiente numero di punti, allora p e q possono essere considerati parti di un cluster. Il punto q è density-reachable da p se c'è una sequenza p_1, \dots, p_n di punti con $p_1 = p$ e $p_n = q$ dove ogni p_{i+1} è density-reachable direttamente da p_i . Si osservi che la relazione density-reachable non è simmetrica dato che q potrebbe situarsi su una periferia del cluster, avendo un numero insufficiente di vicini per considerarlo un elemento genuino del cluster. Di conseguenza la nozione density-connected diventa: due punti p e q sono density-connected se c'è un punto o tale che sia o e p sia o e q sono density-reachable. [EKSX96]

Un cluster, che come già anticipato rappresenta un sotto-insieme dei punti del database, soddisfa due proprietà:

- Tutti i punti all'interno del cluster sono mutualmente density-connected.
- Se un punto è density-connected a un altro punto del cluster, anch'esso è parte del cluster.

DBSCAN necessita di due parametri: ε (eps) e del numero minimo di punti richiesti per formare un cluster (minPts). Si comincia con un punto casuale che non è stato ancora visitato. Viene calcolato il suo ε -vicinato e se contiene un numero sufficiente di punti viene creato un nuovo cluster. Se ciò non avviene il punto viene etichettato come rumore e successivamente potrebbe essere ritrovato in un ε -vicinato sufficientemente grande riconducibile ad un punto differente entrando a far parte di un cluster.

Se un punto è associato ad un cluster anche i punti del suo ε -vicinato sono parte del cluster. Conseguentemente tutti i punti trovati all'interno del suo ε -vicinato sono aggiunti al cluster, così come i loro ε -vicinati. Questo processo continua fino a quando il cluster viene completato. Il processo continua fino a quando non sono stati visitati tutti i punti. DBSCAN presenta i seguenti vantaggi:

- Non richiede di conoscere il numero di cluster a priori, al contrario dell'algoritmo K-means.

- Puó trovare cluster di forme arbitrarie. Può anche trovare un cluster completamente circondato da un cluster differente a cui non é connesso (dato il parametro MinPts, il cosiddetto effetto single-link, cluster differenti connessi da una sottile linea di punti, viene ridotto).
- Possiede la nozione di rumore.
- Richiede soltanto due parametri ed é per lo piú insensibile all'ordine dei punti nel database: solo i punti posti sull'arco fra due cluster differenti possono cambiare la loro appartenenza se l'ordine dei punti viene cambiato mentre l'assegnazione ai cluster é unico solo su isomorfismi.

Tuttavia la qualità del clustering generato da DBSCAN dipende dalla sua misura della distanza. La piú comune misura usata é la distanza euclidea. In particolare per il high-dimensional data, questa misura della distanza diventa quasi inutile tanto da esser denominata "Maledizione della dimensionalità"; nei fatti diventa difficile trovare un valore appropriato per epsilon. Tuttavia questo effetto é presente anche in altri algoritmi basati sulla distanza euclidea.

Inoltre DBSCAN non é in grado di classificare insiemi di dati con grandi differenze nelle densitá, dato che la combinazione minPts-epsilon non puó poi essere scelta in modo appropriato per tutti i cluster.

2.3.4 Clustering gerarchico

In statistica e apprendimento automatico, il clustering gerarchico é un approccio di clustering che mira a costruire una gerarchia di cluster. Le strategie per il clustering gerarchico sono tipicamente di due tipi:

- Agglomerativo: si tratta di un approccio "bottom up" (dal basso verso l'alto) in cui si parte dall'inserimento di ciascun elemento in un cluster differente e si procede quindi all'accorpamento graduale di cluster a due a due.
- Divisivo: si tratta di un approccio "top down" (dall'alto verso il basso) in cui tutti gli elementi si trovano inizialmente in un singolo cluster che viene via via suddiviso ricorsivamente in sotto-cluster.

Il risultato di un clustering gerarchico é rappresentato in un dendrogramma.

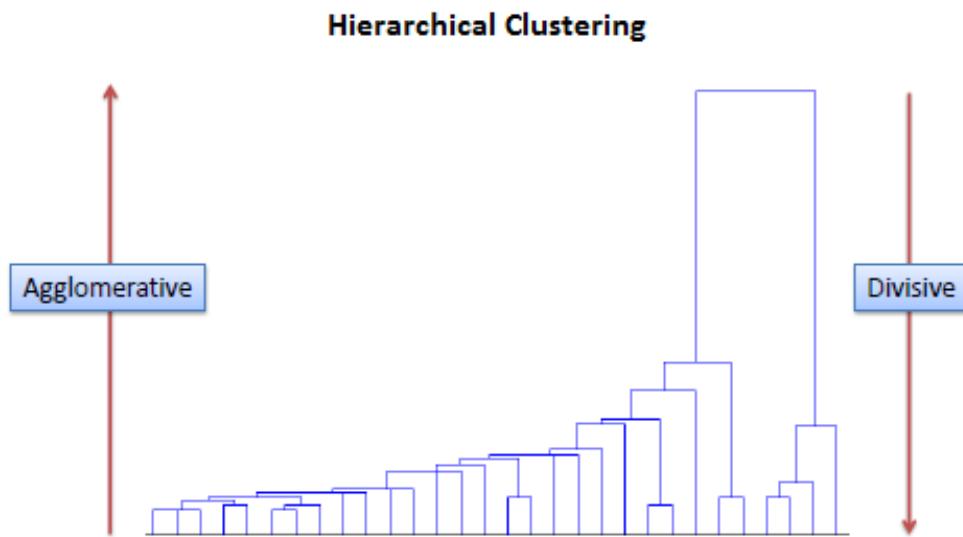


Figura 2.2: Dendrogramma

Secondo la tecnica del clustering agglomerativo, inizialmente ogni singolo data point é considerato come un cluster composto da se stesso. Ad ogni iterazione, cluster simili si fondono fino a convergenza dell'algoritmo. Sostanzialmente si procede come di seguito:

Viene calcolata la matrice di prossimitá all'interno della quale viene definita la distanza di ciascun punto da tutti gli altri. Ogni data point quindi, come anticipato viene classificato come cluster e iterativamente vengono uniti i cluster piú "vicini" e viene aggiornata la matrice di similaritá fino a quando la gerarchia non denota in cima un singolo cluster. La tecnica inversa invece, quella divisiva, rappresenta un metodo molto poco utilizzato in applicazioni reali. Consiste essenzialmente nell'opposto della tecnica espressa in precedenza: Inizialmente infatti si considerano tutti i punti come appartenenti ad un singolo cluster, e iterativamente si separano i punti che presentano una distanza particolarmente elevata. L'algoritmo converge quando viene definita una gerarchia che comprende ogni singolo data point classificato come un singolo cluster.

Risulta pertanto fondamentale la definizione di *distanza* utilizzata particolarmente in questa trattazione. Per decidere infatti quali cluster devono essere combinati (approccio agglomerativo) o quale cluster deve essere suddiviso (approccio divisivo) é necessario

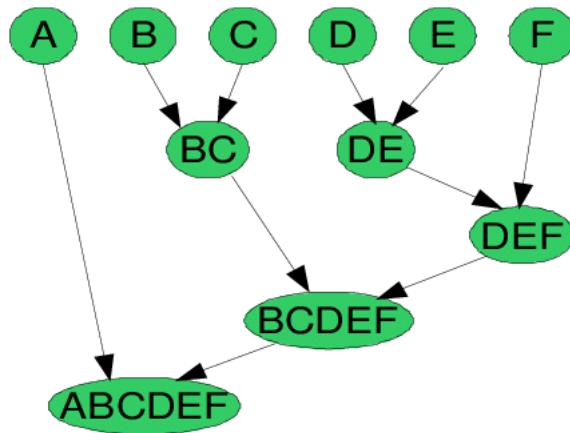


Figura 2.3: Agglomerative Hierarchical Clustering Technique

definire una misura di dissimilaritá tra cluster. Nella maggior parte dei metodi di clustering gerarchico si fa uso di metriche specifiche che quantificano la distanza tra coppie di elementi e di un criterio di collegamento che specifica la dissimilaritá di due insiemi di elementi (cluster) come funzione della distanza a coppie tra elementi nei due insiemi.

La scelta di una metrica appropriata influenza la forma dei cluster, poiché alcuni elementi possono essere piú "vicini" utilizzando una distanza e piú "lontani" utilizzandone un'altra.

Metriche comuni sono le seguenti:

- La distanza euclidea.
- La distanza di Manhattan.
- La norma uniforme.
- La distanza di Mahalanobis, che corregge i dati per scale differenti e le correlazioni nelle variabili.
- La distanza di Hamming, che misura il minimo numero di sostituzioni richieste per cambiare un membro nell'altro.

Per capire invece quando degli insiemi vanno uniti si usano i *Criteri di Collegamento* (Linkage Criteria). I criteri di collegamento vengono effettuati basandosi sulle distanze

fra gli elementi dei cluster e fanno capire se occorre unire i cluster o meno. I criteri sono:

- Collegamento Singolo (Single Linkage): la distanza di collegamento fra due cluster viene definita come la distanza minima fra i due elementi del cluster.
- Collegamento completo (Complete Linkage): la distanza di collegamento fra due cluster viene definita come la distanza massima fra i due elementi del cluster.
- Collegamento Medio (Average Linkage): la distanza di collegamento fra due cluster viene definita come la distanza media fra tutti i punti del cluster con tutti gli altri punti del cluster.

2.4 Tecniche di clustering su grafo

Come è noto il problema *SET-PARTITION*, in cui, dato un insieme S di naturali, si richiede di costruire una bipartizione S_1, S_2 di S tale che

$$S_1 \cup S_2 = S, S_1 \cap S_2 = \emptyset \quad (2.1)$$

$$\sum_{s \in S_1}^s = \sum_{s \in S_2}^s = k \quad (2.2)$$

o la sua generalizzazione da 2 a p componenti sono problemi NP-completi. Tuttavia, quando l'insieme da partizionare in un certo numero di componenti è soggetto ad ulteriori vincoli che legano fra loro alcuni elementi dell'insieme, lo spazio delle soluzioni ammissibili si riduce e gli stessi vincoli consentono di "guidare" un algoritmo in grado di ricavare in modo costruttivo una soluzione ottima del problema, permettendo di ottenere, in alcuni casi, algoritmi di complessità polinomiale. Se dunque l'insieme da suddividere in un certo numero di componenti è l'insieme dei vertici di un grafo e se viene imposto un vincolo di connessione delle componenti della partizione, allora il problema assume spesso le caratteristiche di un problema trattabile e risolubile in tempo polinomiale mediante un algoritmo di bassa complessità ². Con il proseguo della

²<http://www.mat.uniroma3.it/users/liverani> (Last access: 20/09/2019)

trattazione saranno introdotti alcuni algoritmi in grado di effettuare partizionamenti in componenti, che verranno poi identificate come comunità all'interno di una rete sociale.

2.4.1 Affinity Propagation

L'algoritmo di clustering *Affinity Propagation* é stato proposto da Frey e Duek [DF07] e viene trattato all'interno della documentazione della parte di clustering di *scikit-learn*, un modulo Python che integra un alto numero di algoritmi di machine learning sia per problemi supervisionati che non. Il punto centrale dell'algoritmo Affinity Propagation é nell'identificazione di un sottoinsieme di "esemplari rappresentativi".

In input viene presa una matrice di similaritá tra coppie di dati. I dati si scambiano valori reali come messaggi finché non emergono degli esemplari idonei e di conseguenza si ottengono dei buoni cluster.

L'algoritmo é stato scartato durante l'implementazione del lavoro di tesi perché generava troppi cluster e di conseguenza troppi topic anche per insiemi ridotti di tweet.

In statistica e data mining, affinity propagation (AP) é un algoritmo di clustering basato sul concetto di "passaggio di messaggi" tra punti (item). Diversamente dagli algoritmi di clustering quali il k-means, affinity propagation non richiede che sia definito a priori, o stimato prima di esegire l'algoritmo, il numero di cluster.

In breve, in Affinity Propagation, ciascun punto dati invia messaggi a tutti gli altri punti informando i propri bersagli dell'attrattiva (affinity) relativa di ciascun bersaglio per il mittente. Ogni target quindi risponde a tutti i mittenti con una risposta che informa ciascun mittente della sua disponibilitá ad associarsi al mittente, data l'attrattiva dei messaggi che ha ricevuto da tutti gli altri mittenti. I mittenti rispondono ai rispettivi target con messaggi che informano ciascun target dell'attrattività relativa rivista del target per il mittente, dati i messaggi di disponibilitá che ha ricevuto da tutti i target. La procedura di passaggio dei messaggi procede fino al raggiungimento di un consenso. Una volta che il mittente é associato a uno dei suoi obiettivi, tale obiettivo diventa l'esempio del punto. Tutti i punti con lo stesso esemplare vengono inseriti nello stesso cluster.

L'algoritmo procede alternando due step di passaggio di messaggi, per aggiornare due matrici dato un item X_k :

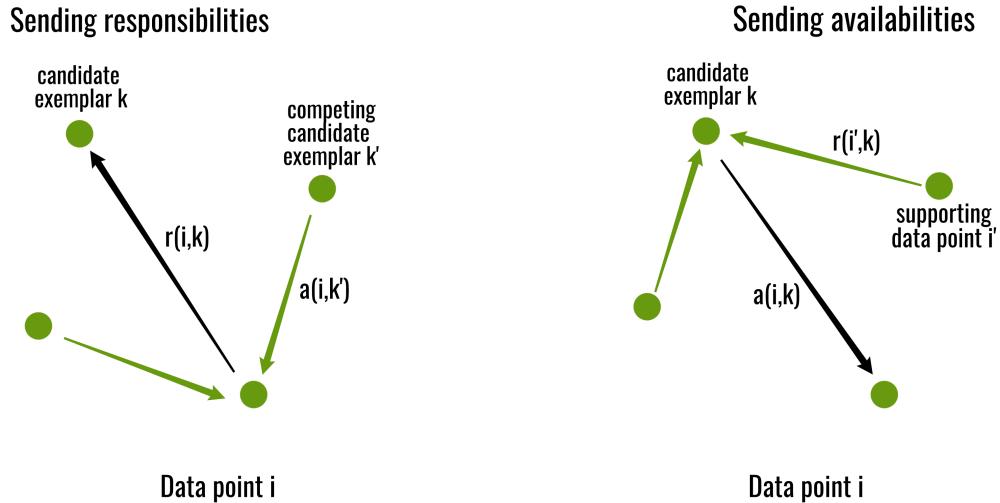


Figura 2.4: Affinity Propagation

- La matrice di "responsabilitá" ("responsibility") contiene valori che quantificano quanto un determinato item X_k sia adeguato come exemplar per gli altri punti, tenendo in considerazione tutti altri candidati exemplar per ogni item.
- La matrice di "disponibilitá" ("availability") contiene valori che rappresentano quanto sarebbe "appropriato" per gli altri item impostare come loro exemplar proprio il punto X_k tenendo in considerazione la preferenza degli altri punti.

2.4.2 Spectral Clustering

Tra gli algoritmi sperimentati vi é anche lo Spectral Clustering. Questo algoritmo suddivide l'insieme dato in cluster usando gli *autovettori* della matrice del grafo. In particolare, come spiegato da Luxburg et al.[UvLB08] prende in input la matrice di adiacenza del grafo che trasforma nella sua matrice di Laplace normalizzata. La matrice di Laplace viene calcolata per differenza tra la *matrice di grado* e la *matrice di adiacenza* dello stesso grafo. Questo processo porta all'interno della matrice, nodi molto connessi in regioni piú vicine. Poi analizzando la matrice raggruppa i nodi vicini con tecniche standard, come ad esempio *k-means*. Ci si potrebbe chiedere il motivo per cui sia necessario raggruppare i punti ottenuti attraverso gli autovettori, quando si puó raggruppare direttamente l'insieme iniziale, in base alla matrice di similaritá.

Il motivo é che il cambiamento di rappresentazione indotto dagli autovettori rende molto piú evidenti le proprietá del cluster dell'insieme di dati iniziale.

In questo modo, il clustering spettrale é in grado di separare i punti che non possono essere separati applicando direttamente tecniche di clustering come k-means, ad esempio, poiché quest'ultimo tende a fornire insiemi di punti convessi.

2.4.3 Louvain Method

L'analisi delle relazioni incognite presenti nei dati riveste un ruolo fondamentale nello studio delle reti: gli archi che collegano i nodi di una rete delineano una struttura fortemente interconnessa che si presta agevolmente, per esempio, alla creazione di gruppi latenti sottostanti. Tale peculiaritá trova fondamento nel concetto - anch'esso nato in ambito sociologico - di omofilia, ossia la tendenza da parte di un nodo a connettersi maggiormente con altri attori simili per una o piú caratteristiche. In questo contesto, un'approccio particolarmente utilizzato é il metodo di Louvain [VDB08]. Un algoritmo di tipo agglomerativo capace di suddividere un grafo in partizioni ed eseguire computazioni con una complessitá temporale lineare rispetto al numero di nodi del grafo. La metodologia é incentrata sul concetto di *modularitá* di una partizione: con modularitá si indica la differenza tra la frazione di archi che connettono nodi appartenenti allo stesso gruppo e il valore atteso della medesima quantitá qualora le connessioni all'interno della rete fossero casuali. Pertanto tale indicatore puó essere utilizzato come un efficace strumento per valutare la qualitá di una data partizione, e dunque puó essere considerata come una funzione obiettivo da massimizzare. Sfortunatamente, l'ottimizzazione esatta risulta spesso computazionalmente problematica e per questo motivo il metodo di Louvain propone un algoritmo capace di approssimare il valore massimo, rendendo quindi il calcolo possibile anche in presenza di reti contraddistinte da un numero elevato di nodi. Tale procedimento iterativo inizialmente assegna ogni nodo ad una comunità diversa, per poi alternare due fasi ciclicamente:

- *Prima fase:* Per ogni nodo $i \in V$ si considera ciascun vicino $j \in V$ e si sposta i nella comunità di j permettendo un guadagno massimo e positivo di modularitá, qualora ciò non fosse possibile si lascia i nella propria comunità.

Questo si ripete per ogni nodo fintanto che è possibile rilevare un miglioramento della modularità.

- *Seconda fase:* prevede invece che ogni comunità rilevata venga considerata come un nodo di una nuova rete, dove il peso della connessione tra due nodi è dato dalla somma dei pesi degli archi tra i nodi delle due comunità e gli archi interni contribuiscono alla formazione di un loop sullo stesso gruppo.

Definendo come passo la combinazione delle due fasi appena descritte, l'algoritmo procede iterativamente finché la prima fase di un determinato passo non propone alcuna modifica nella partizione rilevata allo step precedente.

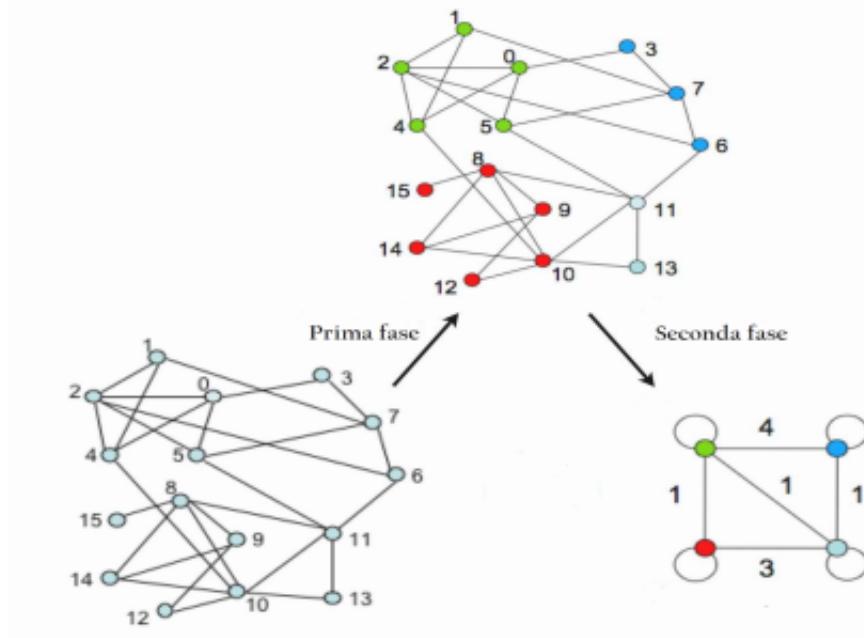


Figura 2.5: Algoritmo Louvain: esempio di passo.

La velocità di esecuzione e la precisione dei risultati generati fa del Louvain Method uno degli algoritmi più usati negli ultimi anni per l'identificazione non supervisionata di comunità in reti di grandi dimensioni. Per la prima volta si parla della possibilità di analizzare reti con milioni di nodi e miliardi di archi in tempi molto brevi.

2.4.4 Hyperlink Induced Topic Search (HITS)

Usando la rappresentazione a grafo è facile definire due criteri per individuare le comunità all'interno del grafo:

1. le entità (nodi) della comunità dovrebbero essere simili fra di loro secondo una certa misura di similarità;
2. le entità della comunità dovrebbero interagire fra loro più di quanto interagiscano con altre entità;

Il primo criterio assicura che le entità condividano effettivamente le stesse caratteristiche. Il secondo criterio garantisce che le stesse interagiscano fra loro in maniera significativa, rendendole in tal modo una comunità effettiva piuttosto che un set di utenti con le stesse caratteristiche che non interagiscono mai fra loro. *Hyperlink-Induced Topic Search* (HITS) è un algoritmo di link analysis precursore di *PageRank*, sviluppato da Jon Kleinberg e inizialmente usato nel ranking di pagine web [Kle99]. I due algoritmi sono molto simili, tranne per il fatto che HITS è *query-dependent*. In particolare HITS stima il valore del *contenuto* di una pagina (*authority score*) e il valore complessivo dei suoi *link* alle altre pagine (*hub score*). Si basa fondamentalmente su di un insieme circolare di assunzioni:

- buoni hub puntano a buone authority;
- buone authority sono puntate da buoni hub;

Esiste quindi una interdipendenza, cioè l'authority score dipende dal hub score, che a sua volta dipende dall'authority score. Entrambi i valori sono calcolati, tramite un algoritmo iterativo basato soltanto sulla struttura dei link e può essere impiegato per trovare comunità nella social network analysis. Dato un set di authority score e hub score, HITS aggiorna gli score secondo le seguenti equazioni:

$$A(p) = \sum_{q \rightarrow p}^{H(q)} \quad (2.3)$$

$$H(p) = \sum_{p \rightarrow q}^{A(q)} \quad (2.4)$$

dove $A(p)$ é la somma degli hub score delle entitá che puntano a p , $H(p)$ é la somma degli authority score delle entitá puntate da p e $p \rightarrow q$ indica che esiste un arco fra l'entitá p e l'entitá q . Come giá anticipato si tratta di un algoritmo iterativo per il quale inoltre se ne puó dimostrare la convergenza mediante un numero ridotto di iterazioni. L'algoritmo Hyperlink-Induced Topic Search (HITS) puó essere impiegato per trovare comunità, e in particolare per effettuare una stima di quanto effettivamente ciascuna entitá possa essere considerata significativa, attendibile e in grado di influenzare gli altri membri all'interno di ogni singola comunità. Non a caso all'interno del lavoro di tesi questo algoritmo viene utilizzato a valle di un primo processo di clustering al fine di identificare gli utenti piú autorevoli, i cosiddetti *influencer* all'interno delle comunità identificate tramite *Louvain Method*. Pertanto dato un grafo di entitá, occorre identificare preliminarmente un sottoinsieme di entitá che potrebbero essere membri della comunità, dette entitá candidate e date le entitá candidate, HITS puó essere usato per trovare il *core* della comunità, ovvero le entitá piú autorevoli. La convergenza é ottenuta quando gli authority score e gli hub score di tutte le entitá candidate variano, fra un'iterazione e la successiva, di un valore inferiore alla quantitá ε prestabilita. Una volta che gli authority e gli hub score sono stati calcolati, le entitá possono essere classificate secondo i loro authority score. Tale lista conterrá le entitá piú authoritative all'interno della comunità. Tali entitá sono probabilmente i leader o formano il core della comunità, sulla base delle loro interazioni con gli altri membri della comunità.

Capitolo 3

Topic Modeling

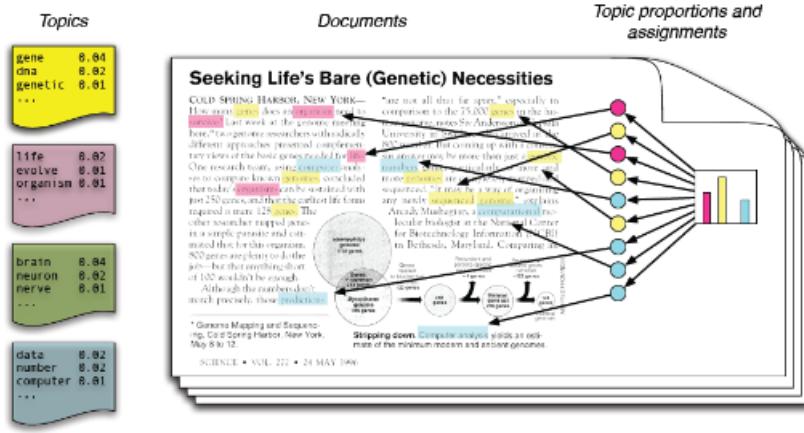
3.1 Introduzione al problema

I social network hanno cambiato radicalmente la quotidianità, semplificandone ed arricchendone diversi aspetti. Ma non solo. Ogni giorno sui social network vengono prodotte quantità enormi di dati, dai quali si possono ricavare numerose informazioni. Si è dunque assistito ad una spinta, causata dall'aumento esponenziale degli utilizzatori di social network, a settori dell'informatica che si occupano di sviluppare tecnologie in grado di dare un significato in maniera automatica a questi dati.

Una volta identificate quindi le comunità, all'interno di una rete sociale, è necessario determinare di cosa queste comunità si occupano e quali interessi accomunano i partecipanti. Per far questo ci si basa sui contenuti generati dagli utenti.

Il topic modeling, ovvero l'estrazione di cluster lessicali co-occorrenti in una collezione di documenti sulla base di calcoli statistico-probablistici, è uno tra i più innovativi e diffusi metodi analitici computazionali tra quelli che vengono comunemente ricompresi nell'insieme dei metodi del *Natural Language Processing*. Più specificamente, dato un insieme di documenti, la stima dei parametri in questi modelli estrae un insieme ristretto di distribuzioni sulle parole, definite *topic*. Come illustrato in alto, Figura 3.1, un generico testo è costituito da un mix di argomenti, z , a cui sono associate parole specifiche di un vocabolario. Ognuno di questi argomenti ha una relativa distribuzione di probabilità sulle parole del vocabolario.

¹<https://www.analyticsvidhya.com/blog/2016/08/>

Figura 3.1: Generazione topic.¹

Ovviamente il topic model non é in grado di dirci quale sia l'argomento, ma attraverso le distribuzioni sulle parole otterremo dei cluster di termini, ove ognuno di questi cluster corrisponde ad un topic a cui saranno associati i termini con probabilitá piú alta (colonna sinistra in Figura 3.1). Le parole con probabilitá piú alta forniscono un'idea dei temi trattati nella collezione di documenti. Quello del topic detection é dunque un contesto vivace: in esso si va a collocare questo lavoro di tesi. Verrá infatti proposto un sistema che risolva il task di topic detection, prendendo in input i dati raccolti dal social network *Twitter*. A valle di questo processo si andrá ad analizzare il sentimento degli utenti rispetto ai temi rilevati.

3.2 Algoritmi per Topic Modeling

3.2.1 Latent Semantic Analysis

L'analisi della semantica (LSA) é una tecnica di elaborazione del linguaggio naturale, introdotta da Jerome Bellegarda nel 2005, in grado di analizzare e determinare il grado di relazione tra documenti all'interno di un corpus. L'obiettivo é quello effettuare dimensionality reduction per la classificazione. [Bel05] LSA assume che parole di significato simile risiedano in porzioni di testo relativamente vicine. Viene costruita una matrice contenente il conteggio relativo alla presenza di ogni termine all'interno di ogni paragrafo all'interno di un documento e tramite la tecnica *singular value decomposition*

(SVD) si riduce il numero di righe preservando la struttura di somiglianza tra le colonne. LSA fornisce in output l'informazione relativa alla similitudine tra documenti sotto forma di matrice utilizzando come metrica la *coseno similaritá*. I valori vanno da -1 a 1, dove -1 rappresenta due documenti che sono completamente opposti e 1 rappresenta un altissima similaritá tra due porzioni di testo. Questo output é dunque sufficiente per creare una matrice le cui colonne rappresentano ciascuna un documento e le cui righe contengono documenti nel loro ordine di somiglianza con il documento associato alla colonna in cui si trovano.

3.2.2 Latent Dirichlet allocation

Entrambe LSA e *Latent Dirichlet Allocation* (LDA) prendono in input del testo rappresentato secondo il modello Bag of Words: un metodo utilizzato nell'Information Retrieval e nell'elaborazione del linguaggio naturale per rappresentare documenti ignorando l'ordine delle parole. In questo modello, ogni documento é considerato in quanto contiene parole, analogamente a una borsa; ciò consente una gestione di queste basata su liste, dove ogni borsa contiene determinate parole di una lista. LSA é incentrata sul restituire e ridurre in output le dimensioni di una matrice contenente il rapporto di similaritá tra documenti all'interno di un corpus. LDA invece si focalizza maggiormente sul risolvere il problema relativo al *Topic Modeling*. Un documento viene considerato come un insieme di argomenti, ognuno dei quali é caratterizzato da una particolare distribuzione di termini e parole. Si tratta di un modello di analisi del linguaggio naturale che permette di prendere in considerazione il significato semantico del testo analizzando la somiglianza tra la distribuzione dei termini all'interno di un documento con quella di un argomento specifico (*topic*) o di un'entitá. Il modello LDA é stato ideato da David Blei, Andrew Ng e Michael Jordan. É stato presentato in un articolo del 2003 su Journal of Machine Learning Research. [BNJ03] Piú di recente, la Latent Dirichlet allocation ha conquistato una certa notorietá anche nell'ottimizzazione SEO semantica come possibile fattore di ranking del search engine Google.

Il modello Latent Dirichlet Allocation é un modello probabilistico generativo che rappresenta ogni documento come una mistura di topic, dove ogni topic assume una distribuzione multinomiale sulle parole del vocabolario, perció ogni documento del corpus



Figura 3.2: Esempio di applicazione di Latent Dirichlet Analysis.

é composto da un insieme di parole ("bag of word") generate da un insieme di topic.

Ognuno di questi topic costituisce una distribuzione multinomiale sulle parole, quest'ultime raggruppate in un vocabolario definito in precedenza sulla base dei testi analizzati: le parole con probabilitá piú alta forniscono un'idea dei temi trattati nel corpus di documenti. In altre parole, si assume che vi sia una distribuzione di topic nel corpus di documenti e che ad ogni topic sia associata una sequenza di parole. L'analisi LDA perció, presuppone un processo di generazione del testo in due stadi: prima si sceglie il topic e poi si sceglie un gruppo di parole per discutere quel determinato topic.

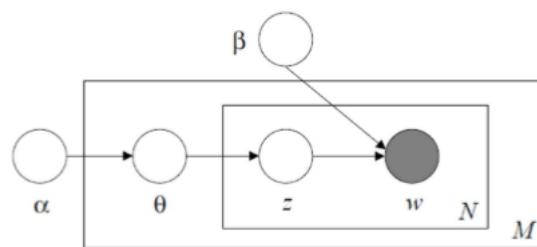


Figura 3.3: Modello grafico LDA

In Figura 3.3 vediamo rappresentato graficamente il modello LDA, dove M rappresenta il numero di documenti e N il numero di parole nei documenti. Mentre α e β rappresentano rispettivamente la distribuzione di Dirichlet di un topic per ogni documento e di una parola per ogni argomento (sono parametri a livello di corpus). Ogni parola è rappresentata dalla variabile w con distribuzione multinomiale, ogni topic è rappresentato dalla variabile z con distribuzione multinomiale e un documento è rappresentato dalla variabile θ con distribuzione di Dirichlet.

Capitolo 4

Opinion Mining

4.1 Introduzione al Problema

Opinion Mining è una disciplina molto recente. Un incrocio tra l'information retrieval e la linguistica computazionale. Si concentra non sull'argomento di cui parla un documento ma sull'opinione che il documento stesso esprime. Sentiment Analysis, Sentiment Classification, Opinion Extraction sono altri nomi usati per identificare questa disciplina in letteratura. È un settore dell'intelligenza artificiale che si occupa del processamento del linguaggio naturale allo scopo di determinare la presenza di soggettività all'interno di espressioni. Per semplicità si potrebbe assumere che queste espressioni siano testuali, ma sono numerosi gli studi che riguardano l'estrazione di sentimento all'interno di altri contenuti, come video e immagini. L'analisi del sentiment è utilizzata in molteplici settori: dalla politica ai mercati azionari, dal marketing alla comunicazione, dall'ambito sportivo a quello delle scienze mediche e naturali, dall'analisi dei social media alla valutazione delle preferenze del consumatore.

4.2 Approcci: Apprendimento automatico e dizionari

La *sentiment analysis* costituisce un campo di studi che analizza le opinioni, i sentimenti, le valutazioni, le stime, le attitudini e le emozioni delle persone nei confronti di prodotti, servizi, organizzazioni, individui, problemi, eventi e topic, e delle loro caratteristiche. Molte delle tecniche di sentiment analysis attuali possono essere ricondotte al machine

learning *supervisionato*. L'apprendimento automatico supervisionato é una metodologia di apprendimento che mira a istruire un sistema informatico in modo da consentirgli di risolvere dei compiti in automatico. Questa tecnica é fortemente usata all'interno del progetto di sentiment analysis trattato. Sulla base di questa definizione si possono quindi definire:

- I dati in ingresso come insieme I , (tipicamente vettori).
- L'insieme dei dati in uscita come insieme O (gli output possono essere valori continui (regressione), o un'etichetta numerica).
- Una funzione h che associa ad ogni dato in ingresso (I) la sua risposta corretta (O).

Tutti gli algoritmi di apprendimento supervisionato partono dal presupposto che nel caso in cui si forniscano all'algoritmo un numero adeguato di esempi, questo sarà in grado di creare una funzione f che approssimerá la funzione h . Se l'approssimazione di h risulterá adeguata quando proporremo ad f dei dati in ingresso mai analizzati la funzione dovrebbe essere in grado di fornire delle risposte in uscita simili a quelle fornite da h e quindi corrette o quasi. Questo tipo di apprendimento tipicamente é piú veloce di quello non supervisionato, nonostante problemi come *overfitting* e la necessità di fornire in input dei dati etichettati correttamente. L'apprendimento non supervisionato consiste invece in una classe di problemi in cui si cerca di determinare automaticamente il modo in cui i dati sono organizzati. Al contrario dell'apprendimento supervisionato, durante l'apprendimento vengono forniti all'apprendista solo esempi non annotati, in quanto le classi non sono note a priori ma devono essere apprese automaticamente. Per quanto riguarda gli algoritmi di classificazione, i piú usati risultano essere SVM *Support Vector Machines* e classificatori probabilistici come *Naive Bayes*. Ultimamente inoltre con la crescente importanza di approcci basati su deep learning sono ampiamente utilizzate anche reti neurali ricorrenti. In particolare, l'approccio sviluppato e verificato nel corso di questo lavoro di tesi riguarda un modello *Long Short-Term Memory*. Relativamente all'analisi dei termini contenuti nella frase, le tecniche piú ricorrenti sono quelle che fanno uso di *bag-of-words*, approccio utilizzato per l'implementazione di una rete neurale ricorrente LSTM descritta in seguito, e *N-Grammi* utilizzata per l'implementazione di

un classificatore basato su SVM, anch'esso descritto nel proseguo della trattazione. La prima tecnica consiste nel rappresentare il documento con una collezione non ordinata di parole, eventualmente con le occorrenze. Questo metodo si dimostra utile utile per ottenere in modo rapido un'analisi sulla frequenza delle parole, ma perde completamente la grammatica, e quindi anche il contesto, delle frasi a cui le parole appartenevano. La seconda tecnica fa uso degli N-grammi, semplicemente una lista di N parole consecutive estratte da una frase. In questo modo si è in grado di tenere in considerazione l'ordine delle parole all'interno di una frase e quindi attribuire un valore semantico alle parole in maniera coerente con il loro utilizzo. Un altro approccio comune è quello basato su dizionari. Le metodologie di sentiment analysis basate su dizionari vengono classificate come tecniche semi-automatiche. In generale, prevedono la presenza di un insieme seed di opinion word iniziale, che viene incrementato nel tempo attraverso l'utilizzo, ad esempio, di sinonimi. Il concetto di dizionario deriva da alcune tecniche di sentiment analysis manuale, che prevedono un elenco prefissato di polar word, ad ognuna delle quali viene attribuita una polarità. A fronte di un'implementazione relativamente semplice, le tecniche basate su dizionari sono meno flessibili degli algoritmi di machine learning: ad esempio, parole dipendenti dal dominio e frasi comparative sono classificate con più precisione dalle tecniche di apprendimento supervisionato, una volta a regime. Tool come *SentiStrength* che saranno trattati in seguito, utilizzano un approccio combinato di queste due metodologie, al fine di fornire un'estrazione del sentiment che possa essere eseguita in tempi ragionevoli senza pregiudicare in maniera eccessiva le prestazioni del classificatore.

4.2.1 Tipologie

L'analisi del sentimento a grana fine fornisce un livello preciso di polarità suddividendolo in diverse categorie, di solito da molto positivo a molto negativo. Questo può essere considerato l'equivalente dell'opinione delle valutazioni su una scala a 5 stelle.

Allo stesso modo il rilevamento delle emozioni identifica emozioni specifiche piuttosto che positività e negatività. Gli esempi potrebbero includere felicità, frustrazione, shock, rabbia e tristezza.

L'analisi basata sugli intenti invece riconosce le azioni dietro un testo oltre all'opi-

nione. Ad esempio, un commento online che esprime frustrazione per la sostituzione di una batteria potrebbe spingere il servizio clienti a contattare per risolvere quel problema specifico.

L'analisi basata sull'aspetto raccoglie il componente specifico che viene citato positivamente o negativamente. Ad esempio, un cliente potrebbe lasciare una recensione su un prodotto dicendo che la durata della batteria era troppo breve. Quindi, il sistema restituirà che il sentimento negativo non riguarda il prodotto nel suo insieme, ma la durata della batteria.

4.2.2 Sfide

Tra le sfide più interessanti in merito all'opinion mining rimane quella che ruota attorno al problema dell'*accuracy* del modello che viene addestrato. La scissione netta tra soggettività e oggettività, o un commento con un sentiment neutrale, ci mettono di fronte ad un problema non banale che il sistema adottato deve risolvere per noi, e spesso purtroppo non risulta esserne in grado. Un esempio basico potrebbe essere il seguente: Il caso in cui un consumatore abbia ordinato un item su di un sito di ecommerce e lo abbia ricevuto di un colore diverso da quello richiesto. Questo provocherebbe una recensione che potrebbe essere identificata come neutra nel caso in cui fosse espressa in questo modo: "Il prodotto era blu, l'avevo ordinato rosso". Nel testo non sono presenti termini che esprimono sentimento anche se la recensione esprime ovviamente un sentimento negativo.

Studiare il sentimento inoltre può essere sfidante nell'identificare le cause per le quali avvengono le misclassificazioni. La categorizzazione di termini come "niente" o "tutto" potrebbe infatti essere confusa dal sistema nel momento in cui il contesto all'interno del quale si opera non sia dato. Allo stesso modo l'ironia come il sarcasmo potrebbero essere non riconosciuti e pertanto portare ad errori nell'etichettatura.

Di rilevanza non certamente inferiore, il problema relativo al rumore presente nei testi. Specialmente nel caso in cui si vadano ad analizzare porzioni di testo intrinsecamente rumorose come sono di certo i Tweet. Un'attenzione particolare infatti va riservata all'allenamento del modello nel caso in cui il training set sia composto di tweet comprendenti emoji e informazioni irrilevanti.

Infine gli utenti, specialmente nel caso in cui siano chiamati ad esprimere le proprie opinioni, possono essere fortemente contradditori nei loro statements.

A tal proposito nell'ambito di questo lavoro di tesi vengono indagate varie tecniche che tentano di ridurre il problema tramite l'analisi semantica della frase in questione. Dalla presenza di subordinate avversative, alla presenza di negazioni.

4.3 Strumenti

4.3.1 SentiStrength

SentiStrength¹ è un classificatore *lexicon-based* che utilizza in abbinamento informazioni linguistiche sintattiche e regole al fine di individuare la forza e la polarizzazione di opinioni espresse in piccoli frammenti di testo informali. Questo strumento per la sentiment analysis è pensato e progettato per testi derivanti dal Social Web, ivi compresi dunque contenuto generato da utenti come lo sono i tweet [ea10].

Per ogni testo, lo strumento sopracitato restituisce in output due interi: il primo da 1 a 5 ad indicare la forza di un sentiment positivo, ed uno score separato, sempre da 1 a 5 per rappresentare la forza della componente negativa del sentiment espresso all'interno del testo. Un valore vicino ad 1 indica sostanzialmente un sentiment nullo, un valore più vicino al 5 indica invece una forte componente di sentiment espressa, in senso positivo o negativo che sia [The12]. SentiStrength riesce nell'intento di combinare un approccio lexicon-based con un altro basato invece su tecniche di apprendimento automatico, ed è in grado di operare in due modalità: supervised and unsupervised. In unsupervised mode, utilizza i pesi predefiniti relativi ai termini polarizzanti presenti all'interno della base di conoscenza. In supervised mode, invece tramite un training set aggiorna automaticamente i pesi relativi ai temini allo scopo di fornire risultati più accurati. Ad esempio, nel caso in cui il termine "pauroso" non sia utilizzato con un'accezione negativa (e.g. discussione su film horror) allora il processo sarà in grado di modificare il peso relativo a quel determinato termine assegnandogli un peso neutro o anche positivo. Tuttavia tramite test sperimentali è stato dimostrato che entrambi gli approcci, supervisionato e non, forniscono un simile livello di accuracy all'interno

¹<http://sentistrength.wlv.ac.uk/>

di un set composto da testi provenienti da social network [The13]. L'algoritmo di detection delle emozioni sul quale si basa SentiStrength è stato sviluppato su un set iniziale di 2.600 classificazioni MySpace utilizzate per i test piloti. Gli elementi chiave di SentiStrength sono elencati di seguito:

Come è stato già accennato in precedenza il nucleo fondamentale dell'algoritmo è costituito dall'elenco di parole che funge da base per la classificazione preliminare del peso di polarizzazione da attribuire a ciascun termine. Questa è una raccolta di 298 termini positivi e 465 termini negativi classificati per forza sentimentale positiva o negativa con un valore da 2 a 5. Le classificazioni predefinite si basano su giudizi umani durante la fase di sviluppo, con modifiche automatiche che si verificano successivamente durante la fase di allenamento, come spiegato nel prosegua.

I pesi attribuiti manualmente, infatti, vengono modificati da un algoritmo di training con l'obiettivo di ottimizzare l'attribuzione della forza polarizzante per ciascun termine. Questo algoritmo inizia l'esecuzione sfruttando i pesi predefiniti e quindi per ciascun termine valuta se un aumento o una riduzione del peso di 1 aumenterebbe l'accuratezza delle classificazioni. Ogni modifica che aumenta l'accuratezza complessiva di almeno 2 viene mantenuta. L'aumento minimo potrebbe anche essere impostato su 1, ma ciò potrebbe provocare overfitting, tuttavia con un fattore 2 si potrebbe rischiare di perdere utili modifiche a parole rare. L'algoritmo verifica casualmente tutte le parole dell'elenco dei sentimenti e viene ripetuto fino a quando tutte le parole non sono state verificate senza che i loro pesi relativi siano stati modificati.

Un algoritmo di correzione ortografica identifica lo spelling esatto delle parole che sono state scritte in modo errato a causa dell'inclusione di lettere ripetute. Ad esempio 'ciaoooooo' verrebbe identificato come "ciao" da questo algoritmo. L'algoritmo (a) elimina automaticamente le lettere ripetute più di due volte (ad esempio, helllo -> ciao); (b) elimina le lettere ripetute che si verificano due volte per le lettere che si verificano raramente due volte (in inglese ad esempio, niice -> nice) e (c) elimina le lettere che si verificano due volte se sono inserite all'interno di una parola non correttamente formata che tuttavia mediante l'eliminazione della ripetizione formerebbe una parola standard (ad esempio, nnice -> nice ma non hoop -> hop).

Una *booster word list* inoltre contiene le parole in grado di aumentare o ridurre

la polarizzazione delle parole seguenti, che siano positive o negative. Ciascuna parola aumenta la forza della polarizzazione emozionale di un fattore 1 o 2 in base al rispettivo potere (e.g. 'molto', 'estremamente') oppure lo decresce di 1 (ad esempio, 'qualche').

Un elenco di parole negative in grado di invertire la polarizzazione dell'emozione espressa tramite termini successivi (comprese le parole booster precedenti). Ad esempio, se "molto felice" avesse una forza positiva 4, allora "non molto felice" avrebbe una forza negativa 4. La possibilità che alcuni termini negativi non neghino non è stata incorporata in quanto ciò non sembra accadere spesso nel set di dati pilota.

Le lettere ripetute più volte, necessarie per l'ortografia corretta, vengono utilizzate per dare un potenziamento della forza di 1 ai termini, purché ci siano almeno due lettere aggiuntive. L'uso di lettere ripetute è un mezzo comune per esprimere emozione o energia nei commenti di MySpace o Twitter, ma una lettera ripetuta spesso indica la presenza di un errore di battitura.

Un elenco di 'emoticon' con i relativi pesi, in grado di aggiungere o rimuovere peso pari a 2, completa l'elenco delle liste utili alla rilevazione di peso emozionale aggiuntivo.

A qualsiasi frase con un punto esclamativo è stata assegnata una forza positiva minima di 2.

La punteggiatura ripetuta con incluso almeno un punto esclamativo, aumenta di 1 il peso emozionale relativo al termine o frase immediatamente precedente.

L'emozione negativa è stata ignorata nelle domande. Ad esempio, la domanda "sei arrabbiato?" sarebbe classificata come non contenente sentimento, nonostante la presenza della parola "arrabbiato". Questo non è stato applicato al sentimento positivo perché molte frasi di domande sembravano contenere un sentimento lievemente positivo. In particolare, frasi come "whats up?" erano generalmente classificate come contenenti un sentimento lievemente positivo (forza 2). I suddetti fattori vengono applicati separatamente ad ogni frase. La frase viene divisa per interruzioni di riga nei commenti o dopo punteggiatura diversa dalle emoticon.

4.3.2 Reti neurali ricorrenti: Long Short-Term Memory

Una rete neurale ricorrente (*recurrent neural network*, RNN) è una classe di rete neurale artificiale in cui i valori di uscita di uno strato di un livello superiore vengono utiliz-

zati come ingresso ad uno strato di livello inferiore. Quest'interconnessione tra strati permette l'utilizzo di uno degli strati come memoria di stato, e consente, fornendo in ingresso una sequenza temporale di valori, di modellarne un comportamento dinamico temporale dipendente dalle informazioni ricevute agli istanti di tempo precedenti. Ciò le rende applicabili a compiti di analisi predittiva su sequenze di dati, quali possono essere ad esempio il riconoscimento della grafia o il riconoscimento vocale. In Figura

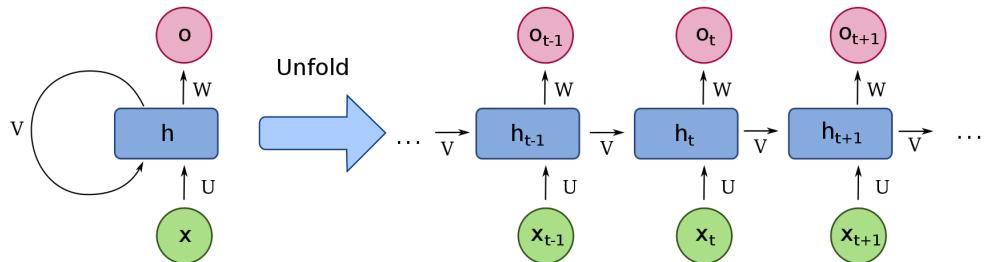


Figura 4.1: Modello di rete neurale ricorrente.

4.1 viene presentato il diagramma di una RNN a una unitá. Dal basso verso l'alto, gli stati in input, gli stati nascosti e gli stati di output. U, V e W sono i pesi della rete. Sulla sinistra il diagramma compresso, sulla destra la sua versione estesa. Ad ogni step della sequenza (ad esempio a ogni istante temporale t) lo strato riceve oltre all'input $x_{(t)}$ anche il suo output dello step precedente $x_{(t-1)}$. Questo consente alla rete di basare le sue decisioni sulla storia passata (effetto memoria) ovvero su tutti gli elementi di una sequenza e sulla loro posizione reciproca. In una frase non é infatti rilevante solo la presenza di specifiche parole ma é importante anche come le parole sono tra loro legate (posizione reciproca). Una *cella* é una parte di rete ricorrente che preserva uno stato (o memoria) interno h_t per ogni istante temporale. É costituita da un numero prefissato di neuroni (puó essere vista come un layer).

In una cella base di RNN lo stato h_t dipende dall'input x_t e dallo stato precedente x_{t-1} :

$$h_t = \phi(x_{(t)}^T \cdot W_x + h_{(t-1)}^T \cdot W_h + b) \quad (4.1)$$

dove:

- W_x e W_y sono i vettori dei peri e b il bias da apprendere.
- ϕ è la funzione di attivazione (ad esempio ReLU).

e l'output y_t corrisponde allo stato h_t :

$$y_{(t)} = h_{(t)} \quad (4.2)$$

Le celle base trovano difficoltà nel ricordare/sfruttare input di step lontani: la memoria dei primi input tende a svanire. D'altro canto sappiamo che in una frase anche le prime parole possono avere un'importanza molto rilevante. Per risolvere questo problema e facilitare la convergenza in applicazioni complesse, sono state proposte celle più evolute dotate di un effetto memoria a lungo termine: *LSTM* e *GRU* (Gated Recurrent Unit) sono le più note tipologie.

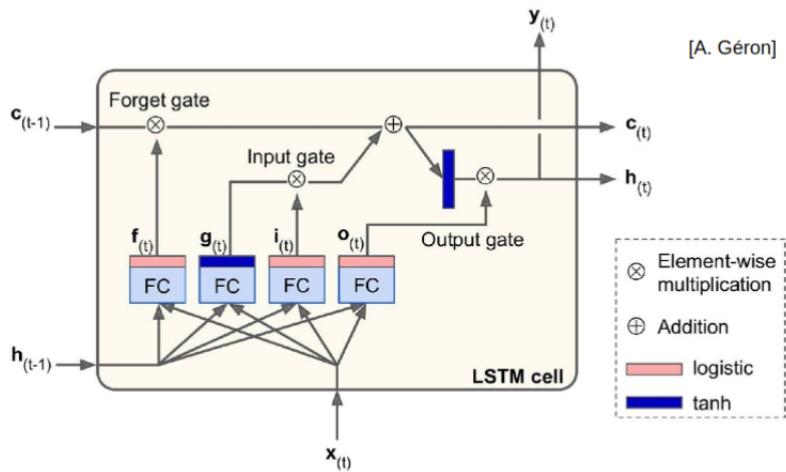


Figura 4.2: Modello LSTM

Nella LSTM (Long Short-Term Memory) lo stato $h_{(t)}$ è suddiviso in due vettori $h_{(t)}$ e $c_{(t)}$:

- $h_{(t)}$ é uno stato (o memoria) a breve termine; anche in questo caso uguale all'output della cella y_t .
- $c_{(t)}$ é uno stato (o memoria) a lungo termine.
- la cella apprende durante il training cosa é importante dimenticare (forget gate) dello stato passato $c_{(t-1)}$ e cosa estrarre e aggiungere (input gate) dall'input corrente $x_{(t)}$.
- per il calcolo dell'output $y_{(t)} = h_{(t)}$ si combina l'input corrente (output gate) con informazioni estratte dalla memoria a lungo termine.

4.3.3 Support Vector Machine

Le macchine a vettori di supporto (*Support Vector Machine, SVM*) sono dei modelli di apprendimento supervisionato associati ad algoritmi di apprendimento per la regressione e la classificazione. Dato un insieme di esempi per l'addestramento (training set), ognuno dei quali etichettato con la classe di appartenenza fra le due possibili classi, un algoritmo di addestramento per le SVM costruisce un modello che assegna i nuovi esempi ad una delle due classi, ottenendo quindi un classificatore lineare binario non probabilistico.

Un modello SVM é una rappresentazione degli esempi come punti nello spazio, mappati in modo tale che gli esempi appartenenti alle due diverse categorie siano chiaramente separati da uno spazio il piú possibile ampio. I nuovi esempi sono quindi mappati nello stesso spazio e la predizione della categoria alla quale appartengono viene fatta sulla base del lato nel quale ricade.

L'SVM é basato sull'idea di trovare un iperpiano che divida al meglio un set di dati in due classi. Per comprenderne il funzionamento é bene definire alcuni concetti chiave:

Support Vector: definiti anche vettori di supporto in italiano, essi sono i punti dati piú vicini all'iperpiano. Tali punti dipendono dal set di dati che si sta analizzando e se vengono rimossi o modificati alterano la posizione dell'iperpiano divisorio. Per questo motivo, possono essere considerati gli elementi critici di un set di dati.

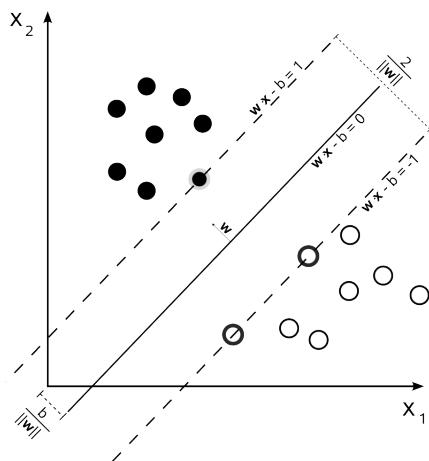


Figura 4.3: Esempio di separazione lineare mediante le SVM.

Margine: é definito come la distanza tra i vettori di supporto di due classi differenti piú vicini all'iperpiano. Alla metá di questa distanza viene tracciato l'iperpiano, o retta nel caso si stia lavorando a due dimensioni. Il Support Vector Machine ha l'obiettivo di identificare l'iperpiano che meglio divide i vettori di supporto in classi. Per farlo esegue i seguenti step:

1. Cerca un iperpiano *linearmente separabile* o un limite di decisione che separa i valori di una classe dall'altro. Se ne esiste piú di uno, cerca quello che ha margine piú alto con i vettori di supporto, per migliorare l'accuratezza del modello.
2. Se tale iperpiano non esiste, SVM utilizza una mappatura *non lineare* per trasformare i dati di allenamento in una dimensione superiore (se siamo a due dimensioni, valuterá i dati in 3 dimensioni). In questo modo, i dati di due classi possono sempre essere separati da un iperpiano, che sará scelto per la suddivisione dei dati.

Nel caso in cui si abbia a disposizione un dataset non lineare, e che quindi non esista un limite di decisione lineare (una singola linea retta che separa le classi), si utilizza un approccio alternativo tramite SVM chiamato *Kernel Trick* (Metodo del Kernel). Gli algoritmi SVM utilizzano un insieme di funzioni matematiche definite come kernel. Il suo scopo é di prendere i dati come input e trasformarli nella forma richiesta qualora

non sia possibile determinare un iperpiano linearmente separabile, come avviene nella maggior parte dei casi. In generale il Kernel puó essere definito come:

$$K(x, y) = \langle f(x), f(y) \rangle \quad (4.3)$$

dove K é la funzione del kernel, x, y sono vettori di input a dimensione n , f é usato per mappare l'input dallo spazio n dimensionale a quello m dimensionale (di livello più alto del livello n). Mentre $\langle x, y \rangle$ indica il prodotto scalare.

Normalmente, il kernel é costituito da una funzione lineare e dunque si ottiene un classificatore lineare. Tuttavia, usando un kernel non lineare si puó modellare un classificatore non lineare senza trasformare completamente i dati: si vanno infatti a modificare solo il prodotto scalare rispetto allo spazio desiderato. Poi SVM avrá il compito di trovare il miglior iperpiano: Il Kernel Trick infatti non é, in realtà, parte di Support Vector Machine. Puó essere utilizzato con altri classificatori lineari come la regressione logistica. L'algoritmo di support vector machine si occupa solo di trovare il limite decisionale, o l'iperpiano migliore. La scelta della funzione del kernel tra gli altri fattori potrebbe influire notevolmente sulle prestazioni di un modello SVM. Le funzioni Kernel piú diffuse sono le seguenti:

- *kernel lineare*, kernel piú semplice e che funziona bene per la classificazione del testo.
- *kernel polinomiale*, che contiene due parametri: una costante c e un grado di libertá d . Un valore d con 1 rappresenta il kernel lineare. Un valore maggiore di d renderá il limite decisionale più complesso e potrebbe comportare un overfitting dei dati.
- *kernel RBF*, (Radial Basis Function) anche chiamato il kernel gaussiano. Risulta in un limite decisionale piú complesso. Il kernel RBF contiene un parametro γ : un piccolo valore di γ fará sì che il modello si comporti come un SVM lineare, mentre un grande valore di γ renderá il modello fortemente influenzato dagli esempi dei vettori di supporto.

Nel corso di questo lavoro di tesi sono stati messi a confronto modelli SVM lineari e non, con tecniche basate su LSTM. I risultati mostrano evidenti differenze derivanti dalla scelta del kernel.

Capitolo 5

Architettura del Sistema

5.1 Introduzione al caso di studio

Negli ultimi anni la crescita e l'ininterrotto flusso di informazioni e dati che ogni giorno sono memorizzati e possono essere sfruttati all'interno del web in generale, e piú nello specifico all'interno dei Social Network Websites, ha portato ad una conseguente importanza sempre maggiore relativa alle metodologie e approcci in grado di fornire vantaggi competitivi non trascurabili alle aziende che scelgono di avvalersi di tali tecnologie. Tali siti infatti forniscono agli utenti registrati diversi strumenti con cui possono comunicare, commentare, replicare e trovare altre persone con le quali interagire al fine di condividere informazioni. Gli utenti e le loro relazioni nei Social Network sono rappresentabili mediante grafi in cui è, quindi, possibile parlare di individuazione di comunità tramite l'identificazione di partizionamenti e l'analisi conseguente centrata su queste community online. L'estrazione o di queste comunità relative ad un argomento online si rivela utile in diversi ambiti, come ad esempio il Digital Marketing, in cui la conoscenza di gruppi specifici di utenti maggiormente influenti al suo interno può costituire un fattore di importanza fondamentale al fine di promuovere prodotti sul mercato in maniera efficace e quindi di incrementare il business relativo. Il presente lavoro di tesi dunque, intende proporre un approccio specifico di Analisi di una rete sociale, in particolare, a partire dal social network *Twitter*.

5.2 Architettura della soluzione

L’obiettivo del sistema sviluppato nell’ambito di questo lavoro di tesi è quello di effettuare l’analisi di una rete sociale basata sulla modellazione delle relazioni tra utenti derivata esclusivamente dai contenuti da essi generati in modo completamente non supervisionato. Sono stati applicati e testati tutti i metodi descritti nei capitoli precedenti e quindi ne sono stati scelti i piú efficienti nonché piú efficaci per l’implementazione dei vari moduli che ne costituiscono l’architettura. In questo capitolo verranno forniti i dettagli dell’architettura del sistema e descritte le modalità con cui sono state prese le scelte implementative. Per la complessità totale del sistema, in fase di analisi dei requisiti, è stato scelto di dividere il progetto in quattro moduli separati. Tale divisione è stata operata principalmente per due motivi. In primo luogo, si è voluto lavorare su moduli coesi. Lo sviluppo di un progetto complesso come questo richiede continui miglioramenti iterativi del codice ed operare in moduli piccoli facilita l’intervento di modifica e la successiva fase di testing, piuttosto che intervenire in un unico progetto enorme. In secondo luogo, la scelta di operare la partizione è stata dettata dalla volontà di creare dei moduli a sé stanti che potessero essere eseguiti in modo separato e in tempi diversi.

Nonostante questi due aspetti i moduli seguono un flusso logico che rispecchia quello della dipendenza di un modulo dall’altro. Tale dipendenza è rappresentata dal senso delle frecce in Figura 5.1.

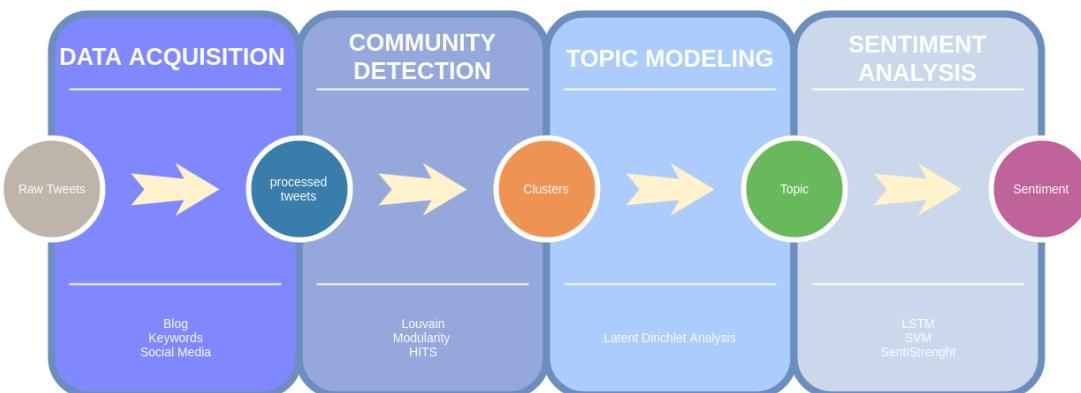


Figura 5.1: Moduli del sistema.

5.2.1 Schema processo

Come anticipato, l'obiettivo di questa ricerca é stato effettuare Social Network Analysis, tramite la combinazione e il test di varie tecniche di Machine Learning, allo scopo di estrarre informazioni utili in modalità completamente non supervisionata. Il processo descritto di seguito é composto dai seguenti step, divisi all'interno dei 4 moduli specificati anticipatamente:

- Acquisizione e analisi del dataset.
- Modellazione del grafo sociale e analisi delle relazioni tra utenti.
- Applicazione metodo di Louvain per la detection delle community.
- Applicazione algoritmo HITS, per la scoperta dei cosiddetti utenti 'influencer'.
- Filtraggio Tweet utenti autorevoli.
- Word Embedding e pre-processing del testo.
- Applicazione Latent Dirichlet Analysis per ogni set di utenti influencer presenti in ogni comunità identificata.
- Detection del topic predominante all'interno delle tre comunità mediante base di conoscenza.
- Analisi del sentiment relativo.

5.2.1.1 Acquisizione e analisi del dataset

Step fondamentale é stata la modellazione del grafo sociale. In genere le informazioni che possono essere estratte da un social network di qualsiasi tipologia sono limitate in base alle normative vigenti in merito alla privacy e alla gestione dei dati. Informazioni relative a relazioni dirette (rapporti di amicizia espliciti - follower/following), in Twitter ad esempio, sono piuttosto difficili da estrarre in gran quantitá, se non sfruttando API a pagamento che il social network mette a disposizione per scopi commerciali. Inoltre questa ricerca é volta ad identificare comunità online implicite basate sui gusti, gli

interessi e le preferenze degli utenti. Senza che essi siano legati obbligatoriamente da relazioni esplicite, ma basandosi esclusivamente sui contenuti da essi generati.

5.2.1.2 Modellazione del grafo sociale e analisi delle relazioni tra utenti

Il dataset utilizzato è costituito da tweet di utenti, sprovvisto di relazioni dirette ('follow') tra gli stessi, ma con informazioni essenziali a corredo, come la presenza di *retweet*, *mention* e *hashtag*.

A partire dunque dai dati a disposizione è stato necessario modellare il grafo sociale ideando possibili relazioni estraibili dal particolare tipo di tweet e dagli identificatori utilizzati al suo interno. I particolari di questa modellazione sono trattati all'interno di un paragrafo ad hoc. Il risultato di questa modellazione è stato un grafo diretto che ha costituito la base per le analisi successive.

5.2.1.3 Applicazione metodo di Louvain per la detection delle community

Con il grafo a disposizione, all'interno del quale i nodi rappresentano gli utenti e gli archi le relazioni implicite tra gli stessi, lo step successivo è stato quello di determinare le comunità. Sono stati dunque testati svariati algoritmi di clustering, e dopo questa valutazione iniziale, tramite l'algoritmo di Louvain, sono state individuate diverse comunità. Louvain sfrutta la funzione di modularità come metrica fondamentale per la quantificazione della qualità della divisione della rete in moduli.

Tre di esse sono risultate essere maggiormente rappresentative in termini di connessioni e numero di nodi, e quindi sono state prese come input per gli step successivi, eliminando i nodi classificati come appartenenti ad altre comunità, allo scopo di agevolare il carico computazionale successivo.

5.2.1.4 Applicazione algoritmo HITS, per la scoperta utenti 'influencer'

Lo step successivo ha riguardato la scoperta degli utenti più autorevoli all'interno delle tre comunità individuate. Tramite l'algoritmo Hyperlink Induced Topic Search (HITS), pertanto è stato prodotto un ranking basato sull'influenza di ciascun utente all'interno della community, ed è stata effettuata una nuova esclusione dal grafo relativa ai nodi aventi un valore di Authority inferiore ad una determinata soglia.

5.2.1.5 Word Embedding e pre-processing del testo

Una volta filtrati e recuperati i tweet relativo agli utenti piú autorevoli, si effettua una fase di pre-processamento dei dati che termina con la rappresentazione del testo secondo il modello *Doc2Bow*. La fase di pre-processing dei testi permette di semplificare la struttura del linguaggio attraverso l'eliminazione dell'informazione derivante dall'ordine delle parole. Le parole diventano quindi "dati" indipendenti tra loro, andando a formare quello che prende il nome di "bag of words". Le trasformazioni eseguibili sono innumerevoli, nel seguito son indicate una serie di operazioni usate nel corso dell'analisi.

Tutte le lettere vengono rese minuscole: generalmente la modalità di scrittura non porta alcuna informazione aggiuntiva, perciò si riduce il testo a sole lettere minuscole per avere una rappresentazione univoca, tenendo presente che in alcuni casi come i blog a volte la scrittura in maiuscolo può accentuare ad esempio il senso di approvazione/disapprovazione che si vuole esprimere.

Vengono rimosse le "stopwords", ovvero tutte quelle parole che non portano informazioni importanti per la comprensione del sentiment come gli articoli, le congiunzioni, le preposizioni semplici e composte, i verbi ausiliari, ecc.

Viene eliminata l'intera punteggiatura che in genere è fondamentale per la comprensione del testo, ma non per l'utilizzo di un metodo statistico.

Vengono rimossi gli spazi bianchi in eccesso (se tra due parole sono presenti piú spazi bianchi, vengono ridotti all'unità) e possono essere rimossi anche i termini che compaiono troppo frequentemente nel corpus (per esempio nel maggiore del 90% dei testi) o troppo raramente (inferiore al 5%). Questo processo di cleaning del testo è stato applicato nella fase relativa all'applicazione degli algoritmi per la topic detection. Quando poi è stato necessario analizzare il sentiment dei contenuti, questa operazione di cleaning è stata modificata in modo tale da non permettere una perdita di informazione significativa. Ad esempio la punteggiatura non è stata rimossa completamente e soprattutto sono state considerate le emoticon, come vettore principale per l'espressione di un'emozione relativa a ciò che si scrive.

5.2.1.6 Applicazione Latent Dirichlet Analysis

A questo punto, avendo a disposizione un input strutturato e pulito dalla maggior parte delle fonti in grado di introdurre rumore è stato eseguito lo step di topic modeling tramite Latent Dirichlet Analysis. Il risultato dell'applicazione di questo algoritmo alle tre differenti comunità ha dimostrato l'esistenza di un argomento trattato in maniera trasversale all'interno delle community relativo alla politica americana. In particolare relativo alle elezioni del 2016, con protagonista il presidente degli Stati Uniti d'America, Donald J. Trump.

5.2.1.7 Sentiment Analysis

Per la fase relativa alla sentiment analysis sono state perseguiti due strade. In particolare infatti ho voluto mettere in risalto la doppia applicazione dei risultati ottenibili da un'analisi di questo tipo. Una volta individuato infatti l'argomento predominante all'interno delle community in questione è possibile, mediante un apposito filtraggio, monitorare l'opinione riguardo l'entità protagonista del topic. Inoltre si può studiare la polarizzazione degli utenti in relazione all'entità protagonista e ad una antagonista. Nel caso preso in esame infatti è stata prima analizzata l'opinione degli utenti centrata sulla singola entità "Donald Trump", poi è stata studiata la polarizzazione degli utenti in merito alla contrapposizione con l'entità antagonista "Hillary Clinton", famosa rivale nelle elezioni presidenziali del 2016. Essendo il dataset estratto da una comunità prettamente giovanile e per lo più universitaria, molto probabilmente sarà affetto da bias. Pertanto i risultati estratti potrebbero non rappresentare l'opinione nel complesso, ma sicuramente sono attendibili per quanto riguarda lo specifico dataset. Tutte le analisi relative al sentiment sono state effettuate mediante il tool SentiStrength, descritto in precedenza, dopo un attenta valutazione e confronto con altri metodi di classificazione presentati nel proseguito della trattazione.

5.2.2 Data Acquisition: Twitter API

L'intero lavoro di tesi è basato sui dati provenienti da Twitter. Tramite infatti le API esposte dal noto servizio di microblogging è possibile infatti estrarre informazioni relative agli utenti, alle loro relazioni e i contenuti da essi generati, che hanno costituito

la base per il dataset necessario alle analisi descritte nel proseguo della trattazione. Le API di Twitter sono realizzate tramite servizi REST.

Richiedono un'autenticazione con token (tramite OAuth2), che può essere immerso direttamente nel codice della particolare libreria utilizzata per le chiamate, ottenibile mediante registrazione sull'apposita pagina dedicata dal social network mediante l'utilizzo di un account Developer.

Un limite importante imposto dalle API è legato al numero di chiamate possibili in un intervallo di tempo: esse infatti sono organizzate in finestre temporali di quindici minuti e, a seconda del servizio richiesto, sono possibili tra le 15 e le 180 chiamate in una singola finestra. Una chiamata che abbia superato il limite massimo ritorna l'errore HTTP 429 - Too Many Requests. Tuttavia, per 'aggirare' in un certo senso tale limite, le librerie utilizzate per lo sfruttamento delle suddette API, nella maggior parte dei casi forniscono un meccanismo di 'sleep' del modulo in esecuzione.

5.2.3 Persistenza del dataset

Al fine di rendere disponibili i dati per le analisi realizzate successivamente, si è progettata la persistenza in modo tale che consentisse memorizzazione e fruizione dei dati in modo veloce e flessibile. Si è deciso dunque di utilizzare una base di dati non relazionale, MongoDB, per la memorizzazione dei tweet, data anche la particolare struttura a documento dei microtesti. I tweet infatti possono avere informazioni, e quindi campi, diversi. Inoltre, le API di Twitter potrebbero cambiare, magari rendendo disponibili nuovi servizi per i tweet: utilizzare MongoDB consente di non definire uno schema a priori, come si dovrebbe in una base di dati relazionale. Infine, essendo libero da vincoli di chiave e integrità referenziale. Inoltre la memorizzazione/lettura del dato è, in generale, più veloce.

5.2.4 Modellazione del Grafo Sociale

L'approccio scelto utilizza una modellazione del grafo sociale orientata alla propagazione con un modello di *rete omogenea di diffusione dell'informazione time-dependent*, dove i nodi della rete rappresentano gli utenti e gli archi interazioni tra gli stessi. Al momento esistono pochi approcci in letteratura che tengano conto delle feature ricavabili dal

conto sociale e della loro evoluzione temporale; d'altro canto le potenzialità invece di questo tipo di approccio sono molto interessanti. La rete quindi è stata modellata come un grafo orientato $G = (V, E)$ dove i nodi $v \in V$ sono rappresentati da utenti della rete sociale e gli archi $(u, v) \in E$ sono rappresentati da interazioni tra gli stessi utenti come *mention*, *retweet* oppure *hashtag*:

Gli archi relativi ad una qualsiasi interazione di *mention* tra un utente $u \in V$ che menziona un altro utente $v \in V$ sono rappresentati da un arco orientato $u \rightarrow v$. La rappresentazione grafica in Figura 5.2.



Figura 5.2: Modellazione Mention.

Gli archi relativi ad una qualsiasi interazione di *retweet* tra un utente $u \in V$ che fa un retweet da un post di un altro utente $v \in V$ sono rappresentati da un arco orientato $v \rightarrow u$. La rappresentazione grafica in Figura 5.3.



Figura 5.3: Modellazione Retweet.

Infine, gli archi relativi ad una qualsiasi interazione di *hashtag* tra un utente $u \in V$ che utilizza un hashtag Temporalmente prima di un altro utente $v \in V$ sono rappresentati da un arco orientato $u \rightarrow v$. La rappresentazione grafica in Figura 5.4.



Figura 5.4: Modellazione Hashtag.

La rete così modellata è costituita da circa 260000 vertici e circa 300000 archi orientati.

5.3 Analisi

5.3.1 Louvain Method per Community Detection

Modellata la rete, il passo successivo è stato quello di riuscire ad individuare un buon partizionamento dei nodi secondo quelle che potrebbero essere delle comunità sociali che si instaurerebbero all'interno della rete d'interazione di ogni utente monitorato. Per riuscire nell'intento abbiamo avuto la necessità è stato necessario il testing di vari algoritmi di community detection al fine di scegliere quello che meglio ripartisca i nodi in maniera omogenea, senza suddividere il grafo in troppo piccole comunità e che riesca a terminare le computazioni in tempi ragionevoli, data la complessità della struttura dati su cui si opera. Gli algoritmi utilizzati per tale scopo sono stati i seguenti:

- Louvain Method.
- K-Means, con embedding Word2Vec e Doc2Vec.
- Density Based SCAN.

Nel prossimo capitolo saranno analizzati i punti di forza e punti deboli relativi agli approcci utilizzati e ai risultati ottenuti. Saranno pertanto esposte le motivazioni che hanno portato alla scelta del metodo di Louvain come algoritmo fondamentale per lo step relativo alla Community Detection.

L'algoritmo *Louvain Community Detection* è stato l'algoritmo che ha prodotto i risultati migliori sia in termini dimensionali in merito alle comunità trovate, che in termini di complessità spaziale e temporale. In Figura 5.5 sono mostrate le community identificate tramite l'applicazione del metodo di Louvain. Oltre alle tre chiaramente più evidenti in quanto formate da un numero maggiore di nodi (colori rosso, nero e giallo), ne sono state trovate anche altre. Dopo un'analisi preliminare del dataset basata anche su conoscenze pregresse, tuttavia, il numero atteso di comunità è risultato essere tra le due e le cinque. Pertanto nel proseguo dell'analisi sono state prese in considerazione

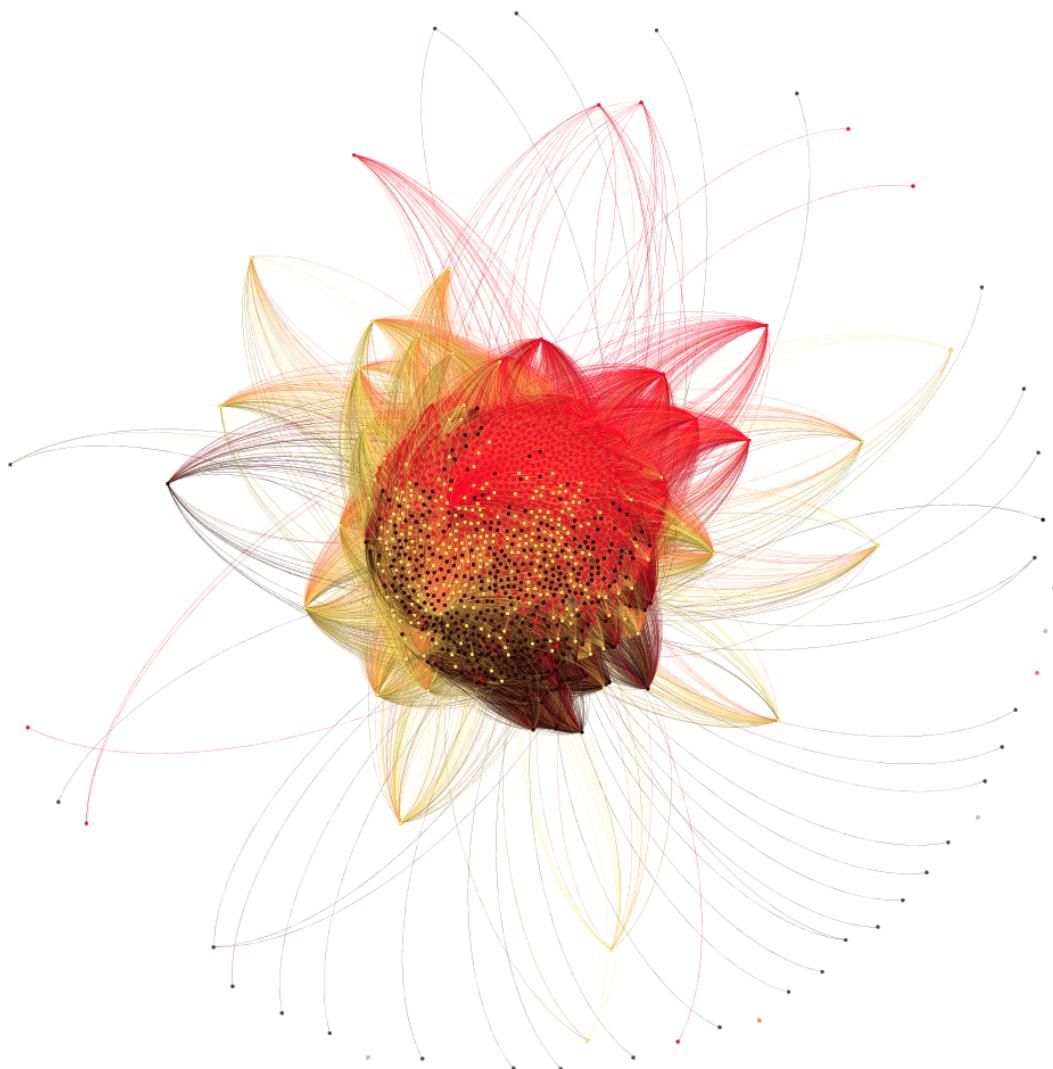


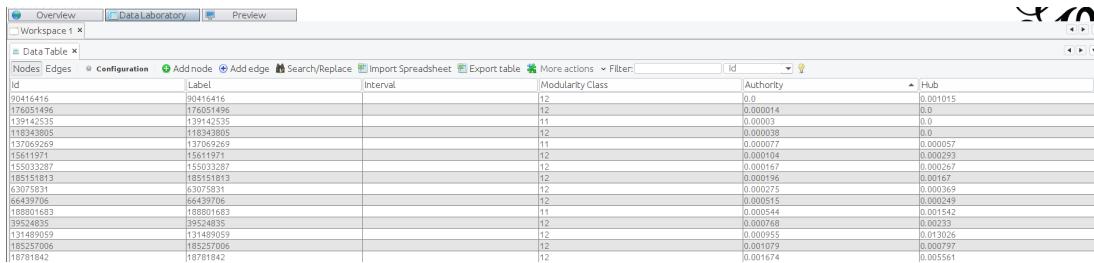
Figura 5.5: Comunità rilevate.

queste tre community maggiormente dominanti. Data anche la necessità di ridurre il numero di nodi, in modo tale da agevolare le computazioni richieste in seguito. Tra le più esigenti sicuramente quella relativa all'esecuzione della tecnica LDA per il task di topic modeling.

5.3.2 Authority Analysis

Per ogni comunità si è inoltre andati ad eseguire un'analisi ulteriore dei nodi rilevanti al suo interno considerandola come se fosse un sotto-grafo indipendente rispetto alle altre. Per ogni comunità è stato eseguito l'algoritmo di *Hyperlink Induced Topic Search* (HITS), per identificare quindi quei nodi (utenti della rete) che rappresentano nodi autorità per la comunità d'utenti. L'algoritmo HITS è stato utilizzato nel progetto per identificare, all'interno delle comunità trovate, gli utenti che ne costituiscono i centroidi e sono appunto degli utenti che caratterizzano al meglio la comunità. Con la loro identificazione è infatti stato possibile aumentare considerevolmente l'accuratezza dei classificatori finali.

Per ogni cluster identificato tramite l'applicazione di Louvain è stata effettuata la misura dei valori di Authority e Hub.



Id	Label	Interval	Modularity Class	Authority	Hub
90416416	90416416	12	0.0	0.001015	0.0
176051496	176051496	12	0.000014	0.0	0.0
1291535	1291535	11	0.000003	0.0	0.0
118343805	118343805	12	0.000038	0.0	0.0
137069269	137069269	11	0.000077	0.000057	0.0000293
15611971	15611971	12	0.000104	0.000104	0.000027
155033287	155033287	12	0.000167	0.000167	0.000056
188801683	188801683	12	0.000196	0.000196	0.000057
63075831	63075831	12	0.000275	0.000369	0.000049
66439706	66439706	12	0.000515	0.000515	0.0000249
188801683	188801683	11	0.000544	0.001542	0.00233
39524835	39524835	12	0.000768	0.000768	0.0013026
1311059	1311059	12	0.000955	0.000955	0.000097
188257000	188257000	12	0.001079	0.001079	0.000097
18781842	18781842	12	0.001474	0.001474	0.000094

Figura 5.6: Valori di Authority e Hub.

In Figura 5.7 è possibile visualizzare in modo agile un filtraggio, rispettivamente relativo ai cluster 1,3 e 2,3, sulla base del quale è stato poi applicato l'algoritmo HITS. I nodi con un valore di Authority più alta hanno dunque dimensione maggiore.

Sono stati scelti, per ogni comunità, i dieci nodi con valore di Authorities per l'algoritmo HITS più alto; questi nodi rappresentano nodi di particolare interesse e verranno utilizzati anche per le classificazioni descritte in seguito.

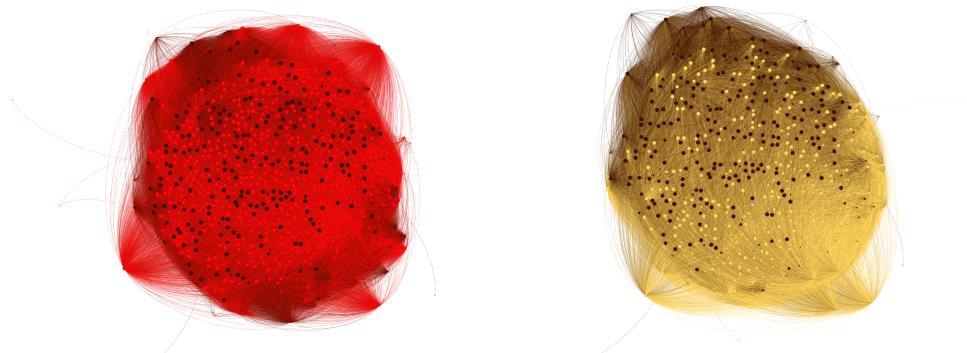


Figura 5.7: Visualizzazione Authority.

5.3.3 Topic Coherence

Al fine di determinare l'argomento maggiormente trattato all'interno delle comunità individuate è stata applicata la tecnica *Latent Dirichlet Analysis*. Dopo aver effettuato un preprocessamento del testo, come spiegato in precedenza, si è stato necessario definire a priori il numero di topic da ricercare. Questa decisione è stata presa in base ad una serie di test attraverso i quali si è andati a misurare l'andamento del *Coherence Value* al variare del numero di topic K. La topic coherence è una metrica per la valutazione della bontà topic estratti. Ciascun topic generato è costituito da parole e la topic coherence dell'argomento viene misurata in base alle prime N parole del topic. Il coherence score dunque viene definito come la media dei punteggi di somiglianza di parole a coppie, rispetto ai termini più significativi nell'argomento.

Un buon modello genererà argomenti coerenti, ovvero topic con punteggi di coerenza elevati. I topic di qualità elevata sono argomenti che possono essere descritti da una breve etichetta, quindi questo è ciò che la misura di coerenza dell'argomento dovrebbe catturare. Il mio approccio per trovare il numero ottimale di argomenti è stato quello di costruire molti modelli LDA con valori diversi di numero di argomenti (k) e scegliere quello che restituisse il valore di coerenza più alto.

Scegliere una “k” che segna la fine di una rapida crescita della coerenza degli argomenti di solito offre argomenti significativi e interpretabili. Scegliere un valore ancora più elevato può talvolta fornire argomenti secondari più granulari.

A tal proposito è bene specificare che sono state effettuate due analisi separate: una

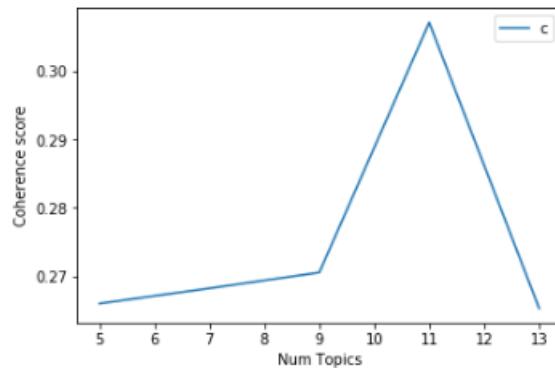


Figura 5.8: Coherence Score.

sull'intero corpus, per determinare gli argomenti trattati a livello globale dagli utenti presenti nel dataset, e un'altra analisi simile centrata sulle comunità singole. A tal proposito come si evince dal grafico in Figura 5.8 è stato scelto un numero di topic pari a 9. Per le singole comunità k è stato scelto pari a 3.

TOPIC cluster 9

```
In [88]: from gensim import models

docs = dfcl9.originalTweet

myStopWords1=set(stopwords.words('english')+stopwords.words('spanish')+list(['http','https']))

documents = [tokenize_and_stem(s, stopwords=myStopWords1) for s in docs]

dictionary = corpora.Dictionary(documents)

corpus = [dictionary.doc2bow(text) for text in documents]

lda = models.LdaMulticore(corpus, id2word=dictionary, num_topics=3, passes=100, workers=9)

#Show first n important word in the topics:
lda.show_topics(7,5)
```

```
Out[88]: [(0,
  '0.020*"trump" + 0.014*"https" + 0.009*"hillary" + 0.008*"clinton" + 0.004*"donald"'),
 (1,
  '0.037*"https" + 0.005*"mais" + 0.004*"jornaloglobo" + 0.004*"earning" + 0.004*"estadao"'),
 (2,
  '0.019*"https" + 0.004*"today" + 0.004*"people" + 0.003*"world" + 0.002*"police")]
```

Figura 5.9: Topic cluster 9.

In Figura 5.9 e 5.10 sono mostrati i risultati relativi all'analisi dei cluster 9 e 11 dai quali si evince chiaramente la preminenza dei topic relativi alla politica americana.

TOPIC cluster 11

```
In [79]: lda = models.LdaMulticore(corpus, id2word=dictionary, num_topics=3, passes=100, workers=9)

#Show first n important word in the topics:
lda.show_topics(7,5)
```

```
Out[79]: [(0,
    '0.014*"https" + 0.006*"retweeted" + 0.005*"police" + 0.004*"auspol" + 0.003*"video"'),
(1,
    '0.016*"小池百合子" + 0.015*"都知事選" + 0.012*"豊洲市場" + 0.010*"royals" + 0.009*"auspol"),
(2,
    '0.026*"trump" + 0.009*"retweeted" + 0.008*"https" + 0.007*"clinton" + 0.005*"donald")]
```

Figura 5.10: Topic cluster 11.

Capitolo 6

Risultati

Questo capitolo descrive l'attività di sperimentazione posta in essere per valutare le prestazioni dei sistemi ideati e realizzati nell'ambito del lavoro di tesi.

6.1 Modellazione centrata su utente

Nel corso di questo lavoro di tesi sono state perseguiti, in modalità essenzialmente parallela, due strade. Sono infatti state concepite due idee di modellazione relativa ai grafi sociali. In prima battuta, tramite un crawler in grado di estrarre automaticamente le relazioni di amicizia tra utenti su Twitter è stata modellata una rete sociale basata per l'appunto su questo tipo di relazioni. In particolare la ricerca è stata centrata su un utente specifico e a partire da questa entità sono stati estratti gli utenti follower e gli utenti following in maniera ricorsiva. Ovvero per ciascun utente risultante all'interno dell'unione delle liste "follower" e "following" dell'utente centrale sono stati modellati tramite archi i collegamenti con le altre entità fino al secondo grado di "parentela". Dal grafo è stato rimosso il nodo relativo all'utente sul quale è stata centrata l'analisi per facilità di visualizzazione.

Questo tipo di rete sociale, costituente un grafo di dimensioni nettamente inferiori rispetto alla rete sociale scaturita dal dataset descritto in seguito, ha costituito la base per i test necessari in un primo periodo di verifica delle tecnologie.

A questo grafo è poi stato applicato l'algoritmo HITS (vedi Capitolo 2) per determinare i nodi con valori di Authority più alta, visibile in base alla dimensione degli stessi,

ed è stato effettuato un partizionamento sulla base della modularità. In Figura 6.1 ogni comunità è rappresentata con un colore differente.

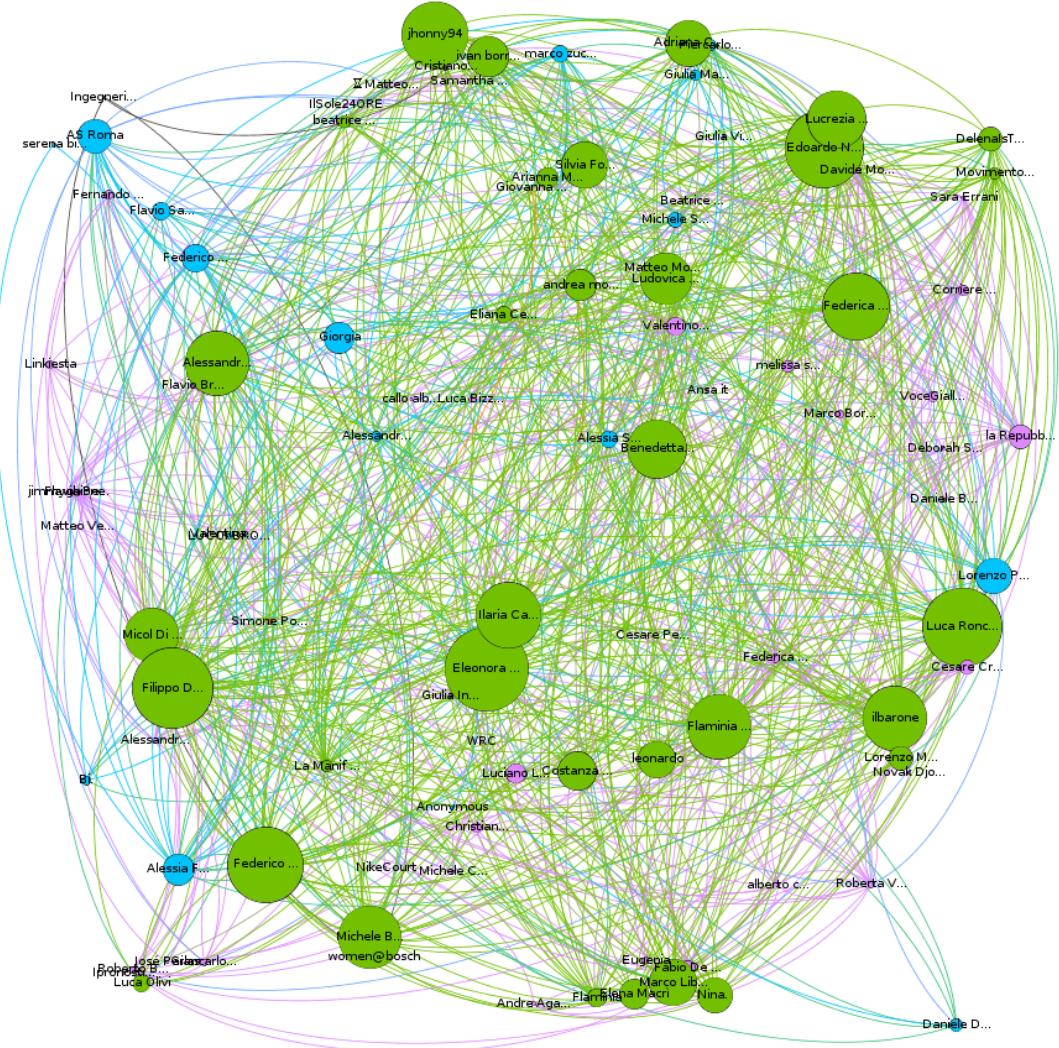


Figura 6.1: Rete sociale centrata su un singolo utente (il cui nodo relativo è stato rimosso per una migliore visualizzazione).

Il risultato dell'applicazione di queste tecniche ha presentato una divisione netta tra utenti afferenti a categorie e gruppi "reali" di persone differenti nella realtà. Ad esempio, sono state classificate come utenze appartenenti a categorie diverse, i collegamenti relativi a persone famose, oppure compagni di scuola, o di gruppi sportivi.

6.1.1 Confronto fra algoritmi di clustering

In fase di valutazione delle differenti tecniche di clustering applicabili per il partizionamento della rete sociale sono stati testati essenzialmente due approcci, descritti nei precedenti paragrafi: quello relativo ad un grafo i cui archi rappresentassero le relazioni implicite tra utenti basate su UGC, e l'approccio relativo al partizionamento di un grafo espresso tramite relazioni esplicite tra utenti e dunque sulla base dei rapporti di amicizia. I risultati migliori sono stati conseguiti mediante l'applicazione del metodo di Louvain sul grafo basato su UGC.

Allo stesso modo, è stata effettuata una fase preliminare di test per valutare quale tecnica fosse la più efficace ed efficiente per lo step relativo alla fase di topic modeling, che ha portato all'utilizzo della tecnica LDA. Di seguito vengono illustrati i risultati relativi ai test effettuati in merito all'applicazione delle tecniche K-Means e DBSCAN, con word embedding Word2Vec e Doc2Vec.

```
Counter({4: 9672, 0: 3908, 3: 1448, 5: 604, 2: 468, 1: 167})
silhouette score: 0.47913548
```

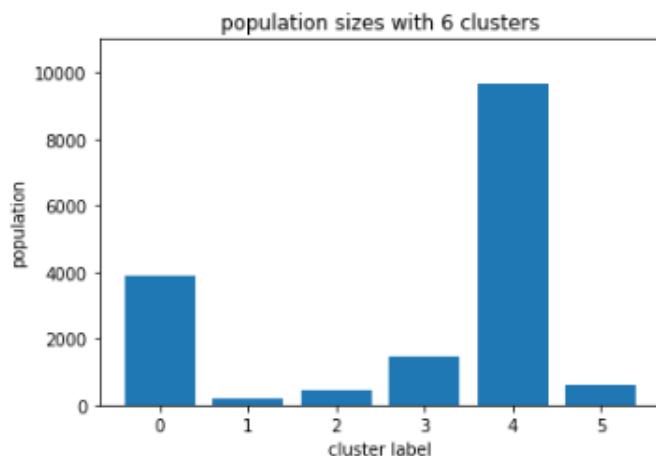


Figura 6.2: Distribuzione tweet K-means con Word2Vec embedding.

In Figura 6.2 sono riportati i risultati relativi all'applicazione di K-Means con Word2Vec embedding. Si può notare a primo impatto come il partizionamento identificato non sia neanche lontanamente soddisfacente. La distribuzione dei tweet appartenenti ai cluster è infatti completamente sbilanciata. Viene identificato un singolo cluster contenente per lo più rumore e gli altri gruppi pressoché inesistenti. Un silhouette score di 0.48.

La situazione migliora leggermente riguardo la distribuzione utilizzando la tecnica di embedding Doc2Vec in grado di tenere in considerazione anche la semantica della frase presa in analisi e non solamente il termine stesso. Tuttavia il coefficiente di S silhouette crolla del tutto.

```
Counter({3: 2552, 0: 1092, 4: 1023, 1: 953, 2: 636, 5: 177})
silhouette score: 0.17361765
```

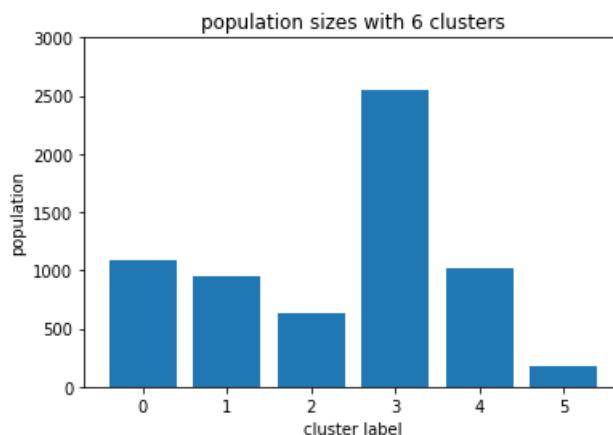


Figura 6.3: Distribuzione tweet K-means con Doc2Vec embedding.

Il valore di "k" impostato a 6 è stato stabilito tramite *Elbow Method* sulla base della somma dei quadrati delle distanze (vedi Figura 6.4).

Il metodo "a gomito" costituisce uno dei metodi più immediati ed utilizzati per la stima del parametro relativo al numero di cluster da identificare: quando la curva ha un angolo a gomito, significa che dopo una prima fase in cui la distanza dei punti dai relativi centroidi è relativamente alta, scende progressivamente fino ad un livello ($k = 6$ in questo caso) in cui i miglioramenti successivi non sono consistenti. Come è ben visibile dal grafico relativo all'elbow method, la linea che viene disegnata non presenta chiaramente una curva di questo tipo. Per ovviare dunque alla decisione relativa alla

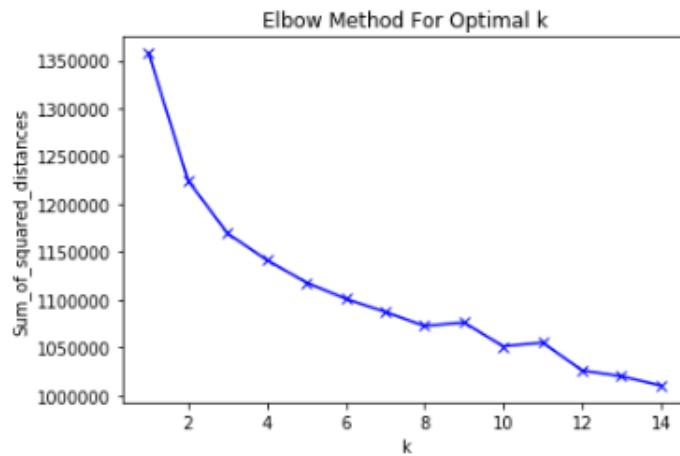


Figura 6.4: Risultati dell’Elbow Method.

stima a priori del parametro k , è stata valutata l’efficacia di un metodo di clustering gerarchico. Una tecnica di questo tipo infatti presenta un duplice vantaggio: non è necessario il parametro k , e soprattutto è un metodo di clustering in grado di classificare i data point come rumore e dunque escluderli dalla predizione finale. Nelle Figure 6.5 e 6.6 sono presentati i risultati relativi all’applicazione di DBSCAN all’insieme di tweet rappresentato tramite Doc2Vec embedding.

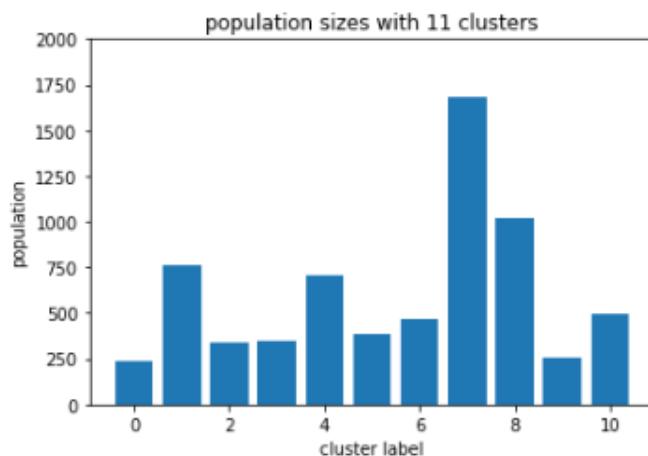


Figura 6.5: Distribuzione tweet DBSCAN con Doc2Vec embedding.

```

Cluster 0 : {'great': 1.0, 'video': 1.0, 'best': 1.0, 'live': 1.0, 'take': 1.0, 'many': 1.0}
Cluster 1 : {'know': 1.0, 'back': 1.0, 'last': 1.0, 'happy': 1.0, 'week': 1.0, 'many': 1.0}
Cluster 2 : {'trump': 1.0, 'world': 1.0, 'great': 1.0, 'think': 1.0, 'year': 1.0, 'president': 1.0}
Cluster 3 : {'great': 1.0, 'make': 1.0, 'year': 1.0, 'made': 1.0, 'also': 1.0, 'change': 1.0}
Cluster 4 : {'people': 1.0, 'http': 1.0, 'could': 1.0, 'police': 1.0, 'watch': 1.0}
Cluster 5 : {'video': 1.0, 'make': 1.0, 'could': 1.0, 'donald': 1.0, 'america': 1.0, 'vote': 1.0}
Cluster 6 : {'company': 1.0, 'https': 0.0}
Cluster 7 : {'today': 1.0, 'could': 1.0, 'work': 1.0, 'black': 1.0, 'game': 1.0}
Cluster 8 : {'could': 1.0, 'stories': 1.0, 'service': 1.0, 'listen': 1.0, 'group': 1.0, 'friend': 1.0}
Cluster 9 : {'thanks': 1.0, 'still': 1.0, 'going': 1.0, 'year': 1.0, 'work': 1.0, 'times': 1.0}
Cluster 10 : {'https': 1.0, 'years': 1.0, 'morning': 1.0, 'game': 1.0, 'might': 1.0, 'early': 1.0}

```

Figura 6.6: Termini principali dei cluster identificati tramite DBSCAN.

6.2 Descrizione Dataset

6.2.1 Considerazioni quantitative

Il dataset è composto da 4.934.827 tweet, raccolti tra il 01/01/2016 e il 12/12/2016, condivisi da 1619 utenti.

Più del 44% dei tweet sono in lingua inglese.

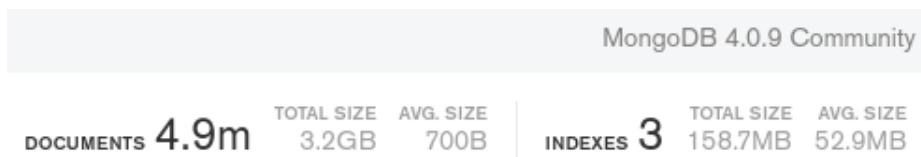


Figura 6.7: Dimensioni del dataset.



Figura 6.8: Linguaggio dei tweet.

Il grafico in Figura 6.9, invece, suddivide gli utenti a seconda di intervalli di quantità di tweet. Come si può notare, la stragrande maggioranza di utenti ha meno di 100 tweet nel periodo di osservazione.

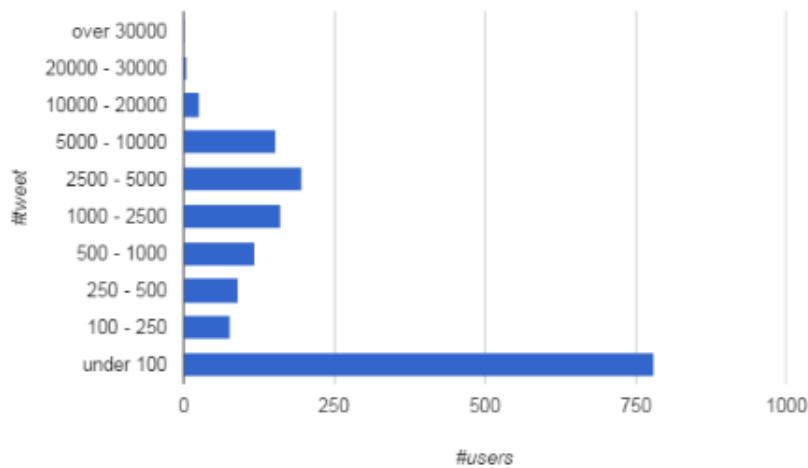


Figura 6.9: Distribuzione utenti rispetto a UGC.

6.2.2 Considerazioni qualitative

Il dataset è composto da 793.596 concept distinti, estratti dagli hashtag degli utenti. Ciascun concept, tuttavia, può essere presente in più tweet. La Figura 6.10 evidenzia, quelli che sono stati i trend del 2016. In effetti, gli hashtag usati corrispondono agli eventi che hanno destato più interesse: la guerra in Siria, la campagna politica americana per l'elezione del nuovo presidente (e i commenti sui risultati), le Olimpiadi di Rio De Janeiro e così via. Queste considerazioni confermano anche l'efficacia dell'utilizzo della tecnica LDA che ha rivelato essere proprio la politica americana il topic maggiormente trattato all'interno del dataset a disposizione.

6.3 Analisi opinion Trump

Lo step conclusivo dell'analisi effettuata è stato finalizzato al determinare in che modo le community identificate parlassero del determinato topic estratto dai contenuti generati dagli utenti appartenenti alle stesse. Dopo una fase di valutazione e implementazione di

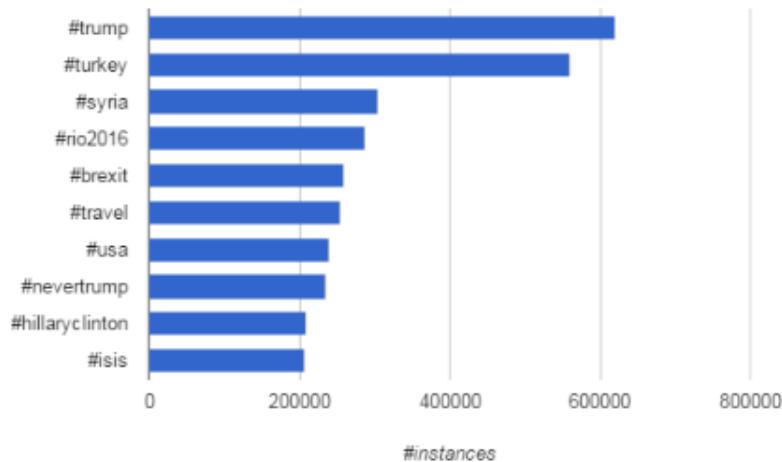


Figura 6.10: I dieci hashtag più utilizzati nel periodo di osservazione.

modelli di classificatori utili ad eseguire sentiment analysis, SentiStrength - dall’alto di un 78% di score misurato tramite F1-measure - è stato giudicato come il più adatto in termini di efficacia ed efficienza. È stato dunque utilizzato il suddetto classificatore per analizzare una parte del dataset costituita dai soli tweet relativi al presidente americano. A tal proposito è stato effettuato un filtraggio mirato basato su mention, hashtag e keyword significative per l’entità in questione.

```
trumpTweet = dfAll[dfAll.originalTweet.str.contains("(?:^|\W)#Trump(?::$|\W)|\\
(?:^|\W)#DonaldTrump(?::$|\W)|\\
(?:^|\W)#trump(?::$|\W)|\\
(?:^|\W)@realDonaldTrump(?::$|\W)|\\
(?:^|\W)#donaldtrump(?::$|\W)\\
(?:^|\W)Trump(?::$|\W)|\\
(?:^|\W)DonaldTrump(?::$|\W)|\\
(?:^|\W)trump(?::$|\W)|\\
(?:^|\W)realDonaldTrump(?::$|\W)|\\
(?:^|\W)donaldtrump(?::$|\W)")]
```

Figura 6.11: Filtraggio tweet inerenti Trump.

Il classificatore pertanto ha operato in modalità binaria, ossia restituendo due valori per ogni testo analizzato, la componente positiva e la componente negativa. È stata infine effettuata una semplice somma algebrica delle componenti per classificare il tweet come positivo, negativo o neutrale (nel caso in cui le due componenti si equivalgano). Il risultato è riportato in Figura 6.12.

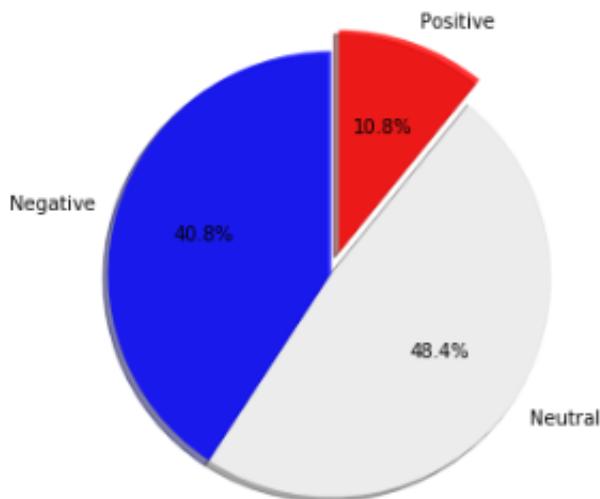


Figura 6.12: Diagramma opinion inerente Trump.

6.4 Polarizzazione utenti: Democratici Vs Repubblicani

Come anticipato nel Capitolo 5 la fase di sentiment analysis è stata costituita da due step fondamentali. Il primo, illustrato nel paragrafo precedente, relativo alle opinioni relative ad una entità target. Il secondo invece ha riguardato l'analisi relativa alla polarizzazione degli utenti relativamente ad un'opposizione tra due entità rivali. Il dataset a disposizione, a tal proposito, si è dimostrato essere particolarmente utile in quanto contenente informazioni relative allo scontro politico tra democratici e repubblicani nelle elezioni del 2016 del presidente degli Stati Uniti d'America.

La procedura che ha portato ad un analisi di questo tipo è la seguente: all'interno del dataset sono stati filtrati, in maniera molto simile rispetto allo step relativo al solo Trump, i tweet relativi prima al candidato Repubblicano poi al candidato del partito Democratico Hillary Clinton. A questo punto sono stati mantenuti soltanto quegli utenti che avessero espresso un'opinione, positiva o negativa che fosse, relativa ad entrambi i candidati. Quindi per ciascun utente rimasto sono stati analizzati tutti i singoli tweet relativi alle due entità rivali. Per ogni tweet classificato come "ProTrump" si va ad incrementare il contatore relativo e lo stesso processo viene effettuato per i tweet "ProClinton". In conclusione quindi, per maggioranza, si va a classificare l'utente come

entità incline alle idee portate avanti dal partito Democratico o Repubblicano. Il grafico in Figura 6.13 mostra il risultato dell’analisi. Più la curva presenta un valore elevato verso le estremità del grafico, più utenti si dimostrano pienamente convinti dell’ideologia politica portata avanti. Più ci si avvicina al centro del grafico, dove si nota anche una leggera sovrapposizione, più si entra in quella zona dove ci si sbilancia leggermente verso un fronte, ma non si ha una piena classificazione relativa all’appartenenza ad una determinata fazione politica.

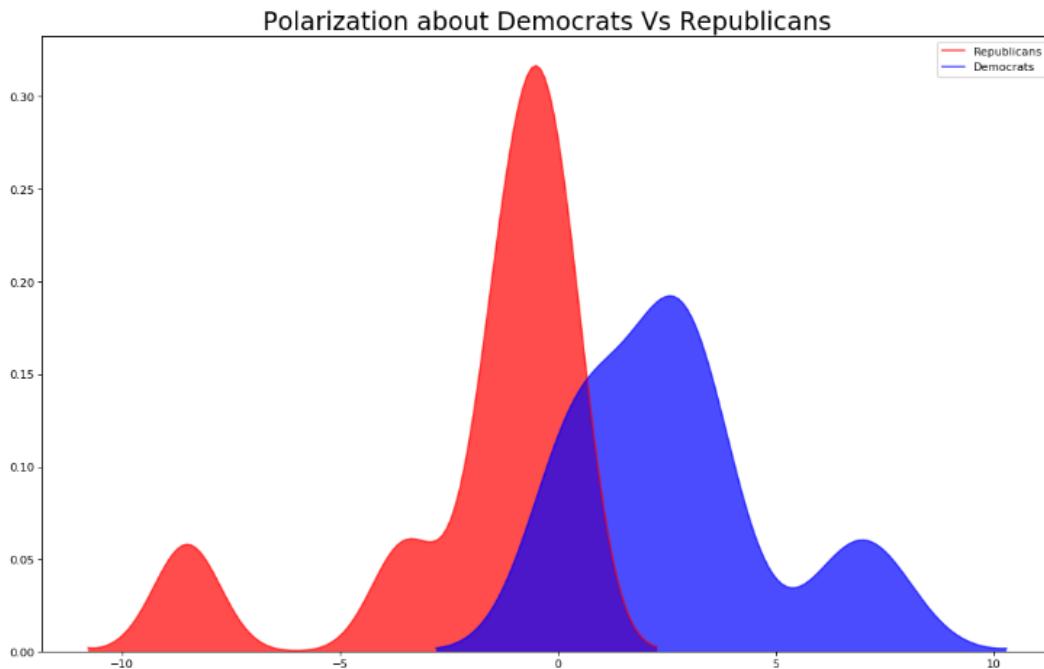


Figura 6.13: Polarizzazione utenti Trump vs. Clinton.

6.5 Confronto algoritmi Opinion Mining

Nella fase relativa alla valutazione delle tecniche utili allo step di Sentiment Analysis sono stati testati diversi algoritmi e modelli, illustrati nei paragrafi successivi. I modelli oggetto di sperimentazione sono stati:

- Long Short-Term Memory (LSTM) Recurrent Neural Network.

- Support Vector Machines (SVM) Lineare.
- Support Vector Machines con kernel Radial Basis Function.
- Approccio misto basato sull'utilizzo di dizionari e tecniche di apprendimento automatico, tramite SentiStrength.

Come set di training per i modelli addestrati è stato utilizzato un dataset¹ specifico prelevato dal noto host web Kaggle. Il seguente insieme di tweet è costituito dai seguenti campi: ItemID (ID del tweet), Sentiment (label relativa al sentiment rilevato) e SentimentText, campo destinato a presentare l'effettivo testo del tweet. Le label relative al sentiment sono di due tipi: "0" ad indicare un sentiment negativo e "1" l'opposto. In Figura 6.14 è riportata una porzione esemplificativa del training set, dopo aver effettuato le consuete operazioni di text pre-processing.

	SentimentText	Sentiment
0	is so sad for my apl friend	neg
1	i missed the new moon trailer	neg
2	omg its already 730 o	pos
3	omgaga im sooo im gunna cry ive be...	neg
4	i think mi bf is cheating on me ...	neg

Figura 6.14: Estratto del training set.

6.5.1 LSTM

Il primo approccio testato è costituito da un modello di rete neurale ricorrente LSTM. La topologia delle reti, guidata da un approccio puramente empirico, è formata da: un primo strato di embedding, che trasforma numeri interi positivi in vettori densi di dimensione fissa. Poi uno strato di dropout che permette di impostare a 0 il tasso di apprendimento per una frazione di unità in input casuale a ogni aggiornamento durante il periodo di addestramento, il che aiuta a prevenire un eccesso di adattamento

¹<https://www.kaggle.com/c/twitter-sentiment-analysis2/overview>

(overfitting). Come penultimo strato abbiamo quello che caratterizza effettivamente la rete neurale come ricorrente, ovverosia lo strato LSTM. A completare la struttura della rete neurale uno strato denso fully-connected utile per restituire in output un vettore di due unità.

Layer (type)	Output Shape	Param #
<hr/>		
embedding_1 (Embedding)	(None, 85, 128)	256000
spatial_dropout1d_1 (Spatial)	(None, 85, 128)	0
lstm_1 (LSTM)	(None, 196)	254800
dense_1 (Dense)	(None, 2)	394
<hr/>		
Total params: 511,194		
Trainable params: 511,194		
Non-trainable params: 0		

Figura 6.15: Topologia della Rete Neurale Ricorrente.

Il modello addestrato si è rivelato particolarmente efficace nella classificazione dei tweet categorizzati come positivi, raggiungendo un 80% di accuracy. Per i negativi invece ha raggiunto un livello di accuracy pari al 70%. Ciononostante lo score, misurato tramite F1-Measure non è neanche lontanamente accettabile, in quanto ha raggiunto uno score pari a 0.49. Difatti il risultato dell'utilizzo del suddetto classificatore sul dataset relativo a Donald Trump ha restituito dei risultati assolutamente non veritieri (vedi Figura 6.16).

6.5.2 SVM lineare

Dati gli scarsi risultati ottenuti mediante l'utilizzo del classificatore basato su LSTM, è stato dunque tentato un approccio tramite Support Vector Machine. In prima analisi è stato addestrato un modello basato su SVM lineare. Sono stati effettuati diversi test, variando essenzialmente la dimensione del validation set in fase di tuning dei parametri tramite k-fold cross validation, e lo score ottenuto non è andato oltre il 71%.

Il problema maggiormente riscontrato nell'utilizzo di un classificatore di questo tipo è stato relativo all'incapacità del modello di riconoscere l'entità protagonista del tweet e basare la classificazione del sentiment sui termini polarizzanti ad esso relativi. Ad

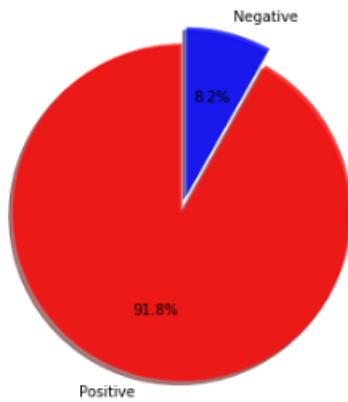


Figura 6.16: Risultati della sentiment analysis tramite LSTM.

esempio, in Figura 6.17 è mostrato un esempio di chiara misclassificazione, dove vengono confuse le entità relative al presidente americano e al sole.

```
In [19]: se = count_vectorizer.transform(["Trump is the worst president ever, but the sun is still awesome!"])
clf30.predict_proba(se)
Out[19]: array([[0.36206397, 0.63793603]])
```

Figura 6.17: Esempio di misclassificazione.

Il classificatore, infatti, in questo caso indica una probabilità pari al 65% circa di sentimento positivo, un risultato chiaramente errato viste le entità presenti nel tweet. Il risultato finale è leggermente meno sbilanciato rispetto al modello LSTM, ma comunque non accettabile. Il modello ha raggiunto un score complessivo pari al 71%.

6.5.3 SVM non lineare

L'utilizzo di un kernel non lineare, gaussiano in questo caso, permette al modello nel caso in cui un singolo iperpiano costituisca un'ipotesi troppo semplicistica per separare linearmente le osservazioni presenti nel dataset di identificare relazioni implicite in maniera più approfondita rispetto ad un modello lineare, proiettando i dati dell'input space in un nuovo spazio detto feature space, a più alta dimensionalità. I vettori che prima non erano linearmente separabili hanno più probabilità di esserlo in uno spazio a più dimensioni. Effettivamente in questo caso l'utilizzo di un kernel RBF fornisce un

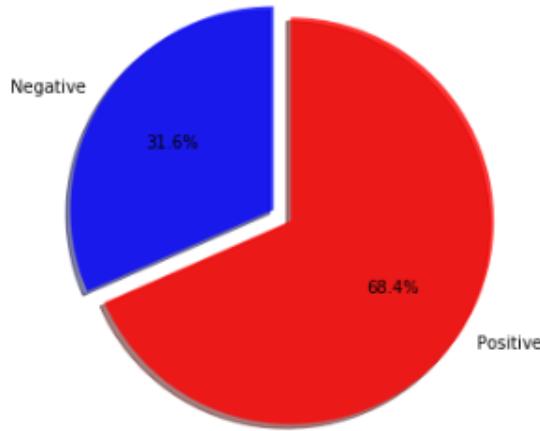


Figura 6.18: Risultati della sentiment analysis tramite linear SVM.

contributo notevole alla classificazione dei tweet. Infatti viene risolto il problema della misclassificazione delle entità presenti nel tweet, come mostrato in Figura 6.19.

```
In [28]: se = count_vectorizer.transform(["Trump is the worst president ever, but the sun is still awesome!"])
clf.predict_proba(se)
Out[28]: array([[0.75702126, 0.24297874]])

In [29]: se = count_vectorizer.transform(["Trump is the worst president ever"])
clf.predict_proba(se)
Out[29]: array([[0.82408381, 0.17591619]])

In [30]: se = count_vectorizer.transform(["Trump is the best president ever"])
clf.predict_proba(se)
Out[30]: array([[0.22238346, 0.77761654]])

In [36]: se = count_vectorizer.transform(["Sun is awesome"])
clf.predict_proba(se)
Out[36]: array([[0.09207318, 0.90792682]])
```

Figura 6.19: Classificazione corretta con entità multiple.

Per l'implementazione del suddetto modello è stata effettuata una fase di tuning degli iperparametri necessari per l'esecuzione di un modello basato su SVM. La stima dei parametri "C" e "Gamma" è stata effettuata tramite k-fold cross validation. Partizionare i dati in dati di training, validation e test, richiede in genere di generare (in maniera random) gli indici degli elementi da assegnare ai diversi insiemi per poi procedere alla suddivisione facendo attenzione a dividere nello stesso modo anche le etichette (in classificazione) o valori target(nella regressione).

La ricerca di valori ottimali per gli iperparametri può essere lunga e noiosa. Quando possibile pertanto è necessario automatizzare questo processo. A tal proposito è stata utilizzata la funzione GridSearch messa a disposizione dalla libreria ScikitLearn: per ogni iperparametro si definisce un insieme di valori da testare. Il sistema è valutato su tutte le combinazioni di valori di tutti gli iperparametri. Può essere molto costoso, come in questo caso.

```

1 | 2019-10-01 10:53:15.032446 : CVCrossGrid Started
2 | Fitting 3 folds for each of 169 candidates, totalling 507 fits
3 | 2019-10-02 10:02:46.381132 : CVCrossGrid Done
4 | The best parameters are {'C': 10.0, 'gamma': 0.01} with a score of 0.77
5 |

```

Figura 6.20: Risultato della k-fold cross validation.

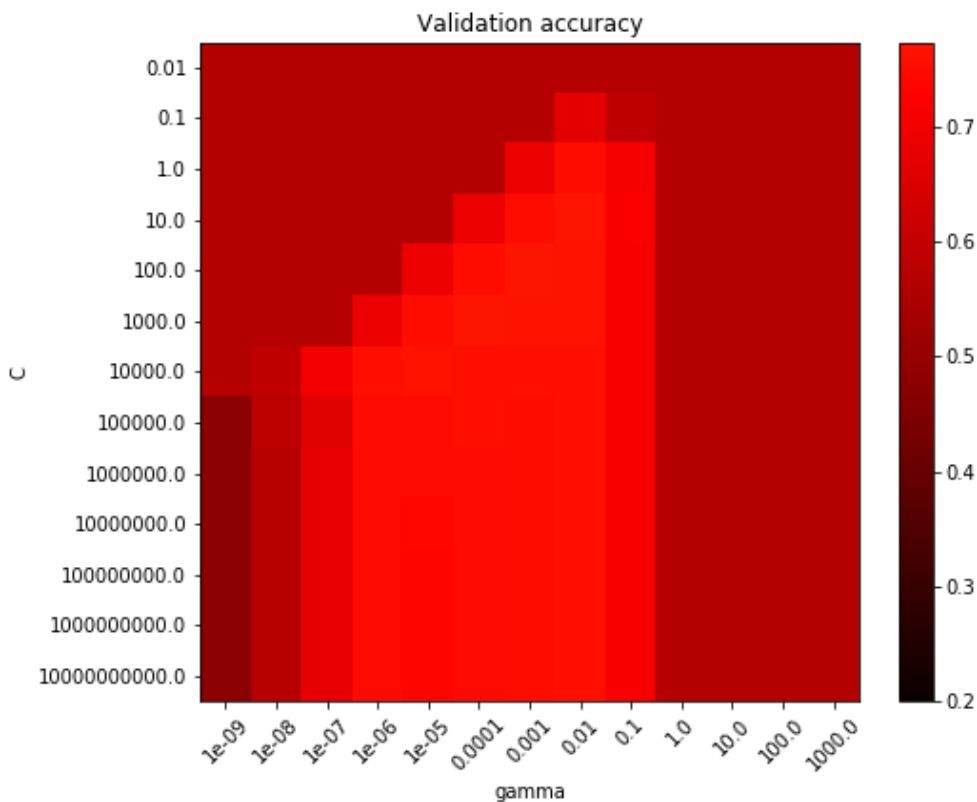


Figura 6.21: Heatmap CVGrid.

Relativamente alla cross validation, ScikitLearn mette a disposizione inoltre la funzione `cross_val_score` che a partire da un unico training set e il numero K di fold, valuta K volte il sistema e ritorna un array di K score (es. accuratezze di classificazione).

I risultati di questo processo sono illustrati in Figura 6.20 e in Figura 6.21 tramite heatmap. Il valore da selezionare è quello relativo alla "punta" del grafico che, come di evince dalla colorazione, presenta un risultato di accuracy più alto rispetto ai parametri scelti. Ovviamente si sceglie quella combinazione che permette il minor sforzo computazionale in fase di addestramento, ossia un valore per C più basso possibile. Il risultato finale della classificazione, sempre più vicina al modello utilizzato come riferimento, ovvero SentiStrength, è illustrato in Figura 6.22.

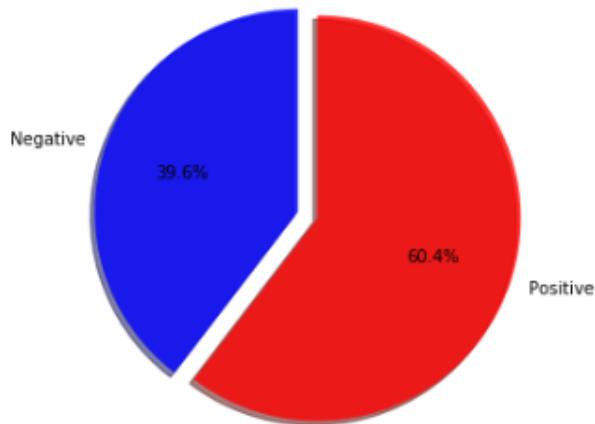


Figura 6.22: Risultati della sentiment analysis tramite SVM con kernel gaussiano.

Conclusioni e sviluppi futuri

Conclusioni

L'obiettivo di questo lavoro di tesi è consistito nella ideazione, realizzazione e valutazione di un sistema in grado di effettuare, in maniera automatica ma soprattutto non supervisionata, step relativi alla Community Detection, al Topic Modeling e alla Sentiment Analysis, sulla base dei contenuti generati esclusivamente dagli utenti presi come campione. Questi step eseguiti in sequenza possono infatti costituire lo scheletro di un'analisi in grado di facilitare l'identificazione di particolari pattern all'interno di specifici gruppi di utenti interessanti.

Per realizzare tale sistema ci si è avvalsi di tecniche di Social Network Analysis e di Machine Learning. In particolare, sono state modellate reti omogenee di utenti sulla base delle loro relazioni implicite espresse dai contenuti da essi generati individuando al loro interno delle comunità individuate tramite tecniche di partizione basate sul valore di modularità.

Queste comunità sono state polarizzate e sono stati identificati al loro interno dei nodi authority, secondo l'algoritmo Hyperlink Induced Topic Search (HITS). Questi utenti, con il relativo UGC, sono stati infine utilizzati come base per l'addestramento e il test dei modelli classificatori sperimentali utilizzati, vale a dire Long Short-Term Memory Recurrent Neural Network, Support Vector Machines Lineare e Support Vector Machines con Kernel RBF, oltre ai modelli per il clustering tramite K-Means e DBSCAN.

Sviluppi futuri

Fra i possibili sviluppi futuri, figura il miglioramento della rilevazione delle entità target all'interno del testo da classificare tramite una segmentazione testuale del tweet e l'isolamento delle entità rilevanti nell'analisi, in modo tale da rendere la classificazione del sentimento più accurata.

Sarebbe, altresì, interessante cambiare il modello di rete utilizzata ed esplorare differenze e vantaggi dell'utilizzare un diverso modello come, ad esempio, una rete omogenea di amicizia o basata sempre sulle relazioni espresse nei tweet, ma monitorata nel tempo. Si potrebbe adottare un graph database come Neo4J² per poter eseguire query dirette su topic interessanti identificati mediante il processo descritto all'interno di questo lavoro di tesi, e poter utilizzare queste reti in parallelo, ottenendo ulteriori informazioni dinamiche sul processo di propagazione dell'informazione.

Sarebbe, inoltre, interessante applicare tali approcci ad altri siti di review, social network o servizi di social bookmarking e sfruttare le diverse funzionalità integrate fra loro per conseguire l'obiettivo proposto.

Si potrebbero, quindi, adottare modelli di Reinforcement Learning per consentire agli approcci implementati un progressivo miglioramento automatico delle prestazioni. Per tale scopo, si potrebbero integrare architetture basate su Convolutional Neural Network, ad esempio per identificare schemi nascosti tra le parole e le immagini, all'interno di un contesto in fortissima espansione come quello di Instagram.

Il sistema sviluppato nell'ambito di questo lavoro di tesi potrebbe, infine, costituire la spina dorsale di un sistema di raccomandazione in grado di rilevare comunità e gusti degli utenti ai quali poi suggerire informazione interessante derivata da analisi dello stesso tipo ma eseguite in parallelo e su diversi insiemi di utenti e contesti.

²<https://neo4j.com>

Appendice

Strumenti software

Tutte le analisi effettuate sono state implementate in Python v3.6, data la versatilità del linguaggio e la propensione dello stesso verso applicazioni relative al Data Mining, al Machine Learning e tecniche per la gestione dei Big Data.

Librerie

Tra le tante librerie utilizzate in fase di analisi spiccano decisamente quelle utilizzate per l'applicazione delle tecniche di Machine Learning sopracitate. In particolare SciKitLearn, Sklearn e Keras per l'implementazione della rete neurale ricorrente. Inoltre per la visualizzazione dei dati e dei risultati annessi sono state utilizzate le famose librerie per grafici Mathplotlib e Seaborn, oltre alla più classica per la gestione dei dataframe Pandas. Per effettuare il preprocessamento del testo invece si è dimostrata di fondamentale importanza la libreria "Natural Language Processing" (NLTK), comprendente anche liste modificabili di StopWords in diverse lingue.

Ambiente

Il processo di analisi è avvenuto in due ambienti sostanzialmente identici dal punto di vista software ma abbastanza diversi per quanto riguarda la potenza dell'hardware a disposizione. Per le analisi computazionalmente meno intensive relativamente a CPU e RAM è stato utilizzato un sistema basato su Ubuntu 19.04 con 15.5GB di memoria utilizzabile, processore Intel Core i7-7700HQ CPU @ 2.80GHz x 8 e scheda grafica GeForce GTX 1050/PCIe/SSE2.

Per le analisi in cui sostanzialmente si è dimostrato utile aumentare la capacità della memoria a disposizione, unico collo di bottiglia importante, oltre ovviamente alla velocità di esecuzione degli algoritmi è stato utilizzato la macchina *Barry* equipaggiata con 60GB di memoria utilizzabile, processore Intel Core i7-6700HQ CPU @ 2.80GHz x 12 e scheda grafica GeForce GTX 1050/PCIe/SSE2, con Ubuntu 19.04 installato.

Tool

Gephi

Con Gephi, un tool per la visualizzazione di grafi e network, è stato possibile visualizzare in forma grafica i risultati ottenuti dalle elaborazioni svolte sui dati. sono stati costruiti dei grafi i cui nodi sono gli utenti caratterizzati dal loro userId e gli archi tra i vari nodi rappresentano le relazioni tra di essi. Grazie a Gephi è stato possibile attuare operazioni di filtraggio dei nodi, nonchè l'applicazione di algoritmi per la visualizzazione notevoli tra cui il layout OpenOrd utilizzato in Figura 5.5.

Bibliografia

- [Bel05] Jerome R. Bellegarda. Unsupervised, language-independent grapheme-to-phoneme conversion by latent analogy. *Speech Communication*, 46(2):140–152, 2005.
- [BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.
- [DA07] Sergei Vassilvitskii David Arthur. k-means++: The advantages of careful seeding. *JO - Proc. of the Annu. ACM-SIAM Symp. on Discrete Algorithms*, 2007.
- [DF07] Dueck and Frey. Non-metric affinity propagation for unsupervised image categorization. *Int. Conf. Computer Vision*, 2007.
- [ea10] Thelwall et al. Sentiment strength detection in short informal text. *American Society for Information Science and Technology*, 2010.
- [EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD’96, pages 226–231. AAAI Press, 1996.
- [FH16] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *ArXiv*, abs/1608.00163, 2016.
- [Kle99] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, September 1999.

- [ME96] Jiirg Sander Xiaowei Xu Martin Ester, Hans-Peter Kriegel. A density-based algorithm for discovering clustersin large spatial databases with noise. *Institute for Computer Science, University of Munic*, 1996.
- [The12] Buckley-K. Paltoglou G. Thelwall, M. Sentiment strength detection for the social web. *American Society for Information Science and Technology*, 2012.
- [The13] Buckley-K. Thelwall, M. Topic-based sentiment analysis for the social web: The role of mood and issue-related words. *American Society for Information Science and Technology*, 2013.
- [UvLB08] Mikhail Belkin Ulrike von Luxburg and Olivier Bousquet. Consistency of spectral clustering. *Max Planck Institute for Biological Cybernetics, Ohio State Universityand Pertinence*, 2008.
- [VDB08] Renaud Lambiotte Etienne Lefebvre Vincent D. Blondel, Jean-Loup Guillaume. Fast unfolding of communities in large networks. *Department of Mathematical Engineering, Université catholique de Louvain*, 2008.