La Casa de Papel: *El Fenomeno*

Stefano Colombo[1], Marco Savino[2], Antonella Zaccaria[3]

Sommario
La Casa di Carta (la *Casa de Papel* nella versione originale e *Money Heist* in quella internazionale) è una serie spagnola trasmessa in anteprima da *Antena 3* e distribuita nel resto del mondo da *Netflix*, grazie al quale è diventata un vero e proprio fenomeno, che con la sua simbologia ed i messaggi veicolati di libertà e rivoluzione ha empaticamente coinvolto persone di tutto il mondo.
In questo progetto sono stati raccolti e analizzati i *tweet* in lingua italiana, inglese e spagnola, relativi alla quarta stagione. Per lo *streaming* e *storage* dei *tweet* sono stati utilizzati rispettivamente *Kafka* e *MongoDB*. Successivamente sono state effettuate le analisi preliminari e la *Sentiment Analysis* sui contenuti postati, in modo da identificare i personaggi più amati e più odiati della serie. Infine il *database* è stato integrato con ulteriori *dataset* riguardanti le informazioni di ogni attore della serie tv. Ciò ha permesso il calcolo di uno *score* di importanza associato ad ogni personaggio.

Keywords
Velocità — Varietà

[1] *838941, Dipartimento di Informatica, Sistemistica e Comunicazione*
[2] *793516, Dipartimento di Informatica, Sistemistica e Comunicazione*
[3] *848647, Dipartimento di Informatica, Sistemistica e Comunicazione*

Indice

1. Caso di studio

La casa di carta (La *casa de papel* nella versione originale e *Money Heist* in quella internazionale) è una serie spagnola trasmessa in anteprima da *Antena 3* e distribuita nel resto del mondo da *Netflix*. E' stata concepita come una serie evento di soli 15 episodi della durata di circa 70 minuti ciascuno, ma il sorprendente successo internazionale ha incoraggiato il servizio di video in *streaming* (*Netflix*) ad acquisirne i diritti di esclusiva per produrne quattro stagioni per un totale di 38 episodi.
La trama racconta lo svolgimento, dalla sua ideazione all'esecuzione, di un incredibile piano criminale ideato dal Professore, per derubare la Zecca di Madrid (nelle prime due stagioni) e la Banca di Spagna (terza e quarta stagione). Per raggiungere il suo obiettivo, il Professore ha reclutato attentamente un gruppo di persone che possono utilizzare solo degli appellativi fittizi di nomi di città in modo da non instaurare relazioni tali da comprometterne il piano.
Le maschere di *Dalì* e le tute rosse, la canzone *Bella Ciao*, l'intelletto del Professore e la sua banda hanno avuto un incredibile impatto mondiale. Il volto del famoso pittore spagnolo ha tempestato le piazze di ogni Paese non solo durante i festeggiamenti del carnevale ma anche in occasione di proteste, seguito dall'inno partigiano, che ha riportato in auge una canzone che (almeno in Italia) tutti conoscevano, ma di cui forse troppi avevano dimenticato il significato. Dunque non si tratta semplicemente di una serie tv ma di un vero e proprio fenomeno che con la sua simbologia ed i messaggi veicolati di libertà e rivoluzione ha empaticamente coinvolto persone di tutto il mondo.

2. Obiettivi

Gli obiettivi che hanno guidato lo sviluppo del progetto sono:

- Implementare un'architettura che consentisse l'utilizzo di due (Velocità, Varietà) delle 3V caratteristiche dei

Big Data.

- Comprendere quali fossero le polarità suscitate negli utenti rispetto ai diversi personaggi citati, in modo da confrontare i personaggi più amati e quelli più odiati in riferimento ai *tweet* in lingua italiana, inglese e spagnola ed analizzare le caratteristiche degli attori che interpretano tali personaggi.

3. Velocità

3.1 Twitter

Twitter [1] è un *social network* che mette a disposizione delle API [2] per consentire agli utenti lo *streaming* dei *tweet* postati. Per l'acquisizione di tali dati è stata utilizzata *Tweepy*, una libreria *Python* che supporta l'autenticazione *OAuth* e ne consente il collegamento grazie a quattro *token* personali (*consumer key, consumer secret, access token* e *access secret*), messi a disposizione direttamente da Twitter dopo la creazione di una nuova applicazione e vengono utilizzati attraverso lo sviluppo di un'istanza *OAuthHandler* nel *Producer*.

3.2 Architettura Kafka e MongoDB

Per garantire la velocità richiesta dal progetto, è stato utilizzato, per l'acquisizione dei dati, l'ecosistema *Hadoop*, in particolare la piattaforma distribuita *Apache Kafka* [3]. L'architettura si serve di:

- un *Producer* per l'acquisizione dei *tweet* in *streaming* e l'invio ad un *topic* creato appositamente dall'utente;
- un *Consumer* che, effettuando la sottoscrizione al *topic* creato, riesce a recuperare in qualsiasi momento i dati immagazzinati.

I dati recuperati dal *Consumer* sono stati infine immagazzinati in *MongoDB* [4] in modo da garantire la persistenza dei dati anche in caso di malfunzionamenti e per effettuare in un secondo momento le analisi sugli stessi.

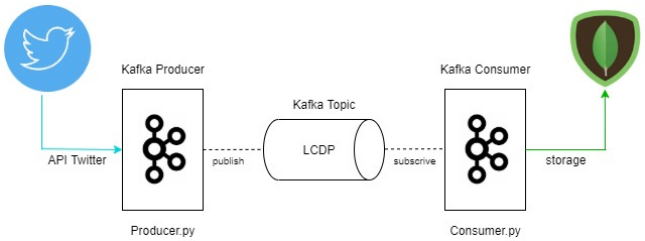

Figura 1. Architettura implementata

[1] https://twitter.com
[2] Application Programming Interface
[3] https://kafka.apache.org/
[4] https://www.mongodb.com/it

3.2.1 Streaming

L'architettura sviluppata permette di effettuare l'acquisizione dei *tweet* contenenti l'*hashtag* ufficiale della serie tv: *#lacasadepapel*. Lo *streaming* dei *tweet* è stato effettuato ricorrendo al metodo *on status* offerto dalla libreria *Tweepy*, definito all'interno della classe *listener* ereditata da *StreamListener*. I *tweet* sono scaricati in formato *JSON* e vengono convertiti in stringa attraverso il comando *json.dumps* per poter essere inviati dal *Producer* al *topic LCDP*, creato precedentemente con il seguente comando:

```
kafka-topics --zookeeper 127.0.0.1:2181 --topic
LCDP --create --partitions 1 --replication-factor 1
```

Il *Consumer* ha il compito di ascoltare i messaggi in arrivo dal *topic* e, all'avvenuta ricezione del messaggio, convertirlo in formato *JSON*. Il nuovo messaggio viene dunque aggiunto come nuovo *record* nella *collection* di *MongoDB*.

3.2.2 MongoDB

Ultimato il processo di *streaming* si è ottenuto un *database MongoDB* composto da 42.821 documenti caratterizzati dalla seguente struttura:


```
_id: ObjectId("5f297e8c2e5ecfd0fd965cf5")
user_id: "703251077047689216"
created_at: "2020-04-08 21:02:10"
screen_name: " Marina_RomaK10"
text: "Ti amo di Umberto Tozzi proprio non me l'aspettavo<U+0001F923> #LaCasa..."
source: "Twitter for Android"
lang: "it"
hashtags: Array
  0: "LaCasaDiCarta4"
  1: "LaCasaDiCarta "
  2: "LaCasaDePapel"
```


Figura 2. Esempio documento *MongoDB*

- *id*: identificativo univoco generato da MongoDB
- *user_id*: identificativo univoco dell'autore del *tweet*
- *created_at*: data e orario di pubblicazione del *tweet*
- *screen_name*: *username* dell'autore del *tweet*
- *text*: testo del *tweet*
- *source*: dispositivo utilizzato dall'autore del *tweet*
- *lang*: lingua del *tweet*
- *hashtags*: *array* contenente gli *hashtags* presenti nel *tweet*.

I dati memorizzati sono stati successivamente esportati tramite il comando *'mongoexport'* per essere analizzati in *Python*.

4. Analisi preliminari

I *tweet* raccolti tra il 01/04/2020 e il 21/04/2020 sono 42.821, di cui:

- 4.226 in lingua italiana

- 15.298 in lingua inglese
- 23.297 in lingua spagnola.

Inizialmente si è analizzata la distribuzione dei *tweet* sulla base della loro data di pubblicazione. Dalla Figura 3 si può

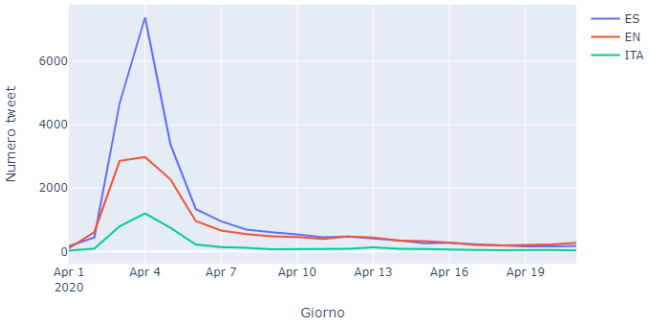

Figura 3. Andamento tweet

notare che il picco dei *tweet* nelle tre diverse lingue, è stato raggiunto il 4 aprile 2020, giorno dell'uscita della quarta stagione della *Casa de Papel*. *Netflix* rende subito disponibili tutte le puntate della serie tv; vi sono tendenzialmente due tipologie di spettatori: chi guarda tutte le puntate consecutivamente e chi le suddivide in diversi giorni. Per questo motivo si è precedentemente deciso di effettuare lo *streaming* dei *tweet* in un arco temporale di venti giorni (dal 01-04-2020 al 21-04-202).
Successivamente sono stati analizzati gli *hashtag* più utilizzati:

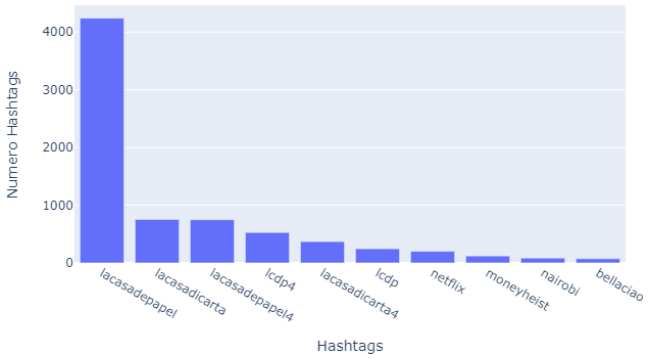

Figura 4. 20 Hashtags più utilizzati nei *tweet* in lingua italiana

Si può notare che, per quanto riguarda la lingua italiana ed inglese (Figura 4, Figura 5), i venti *hashtag* più presenti all'interno dei *tweet*, siano prettamente inerenti alla serie tv. Alcuni degli *hashtag* contenuti invece nei *tweet* in lingua spagnola (Figura 6), come per esempio *quedateencasa* e *cuarentena*, fanno riferimento alla fase di *lockdown* dovuta al *COVID-19*, a dimostrazione che *Netflix* sia stato un ottimo strumento di intrattenimento.

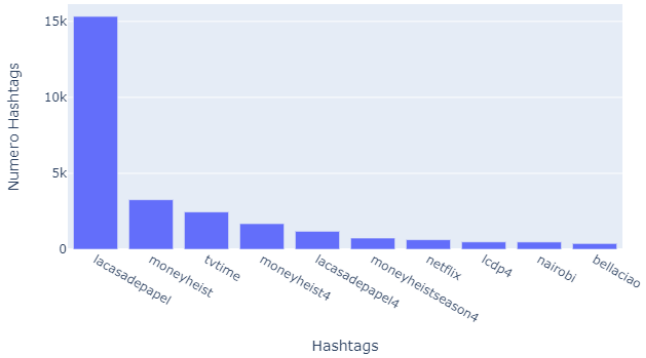

Figura 5. 20 Hashtags più utilizzati nei *tweet* in lingua inglese

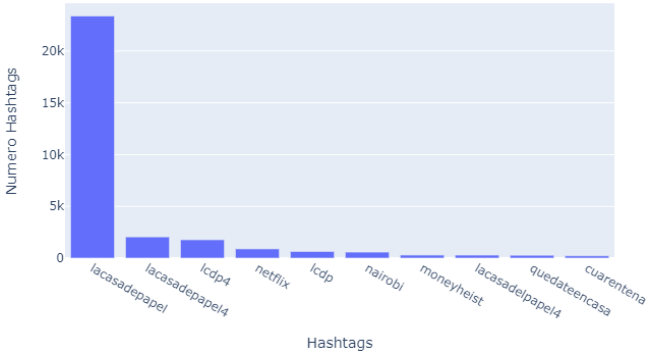

Figura 6. 20 Hashtags più utilizzati nei *tweet* in lingua spagnola

I principali dispositivi utilizzati dagli utenti di Twitter per postare i *tweet* riguardanti la quarta stagione della *casa de papel* sono i seguenti:

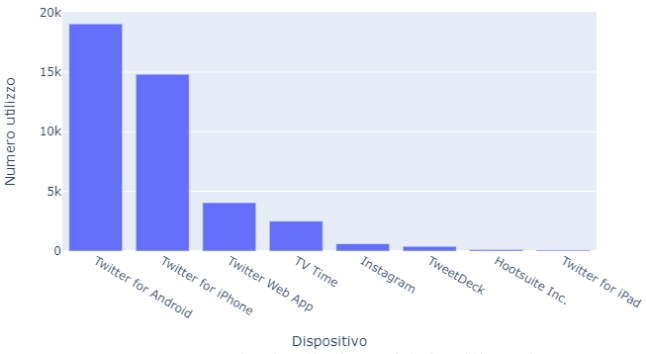

Figura 7. Principali dispositivi utilizzati

In tutte e tre le lingue il dispositivo maggiormente utilizzato è *Twitter for Android*. *TV time* è caratteristico dei *tweet* postati in lingua italiana ed inglese, mentre *Hootsuite Inc.* e *TweetDeck* sono utilizzati solo nei *tweet* in lingua spagnola.
Le analisi proseguono con i personaggi citati nei *tweet* (Figura 8, Figura 9, Figura 10): il numero di citazioni conferma la rilevanza di ciascun personaggio all'interno della quarta stagione. *Nairobi* si conferma infatti il personaggio più citato all'interno dei *tweet* nelle tre diverse lingue.

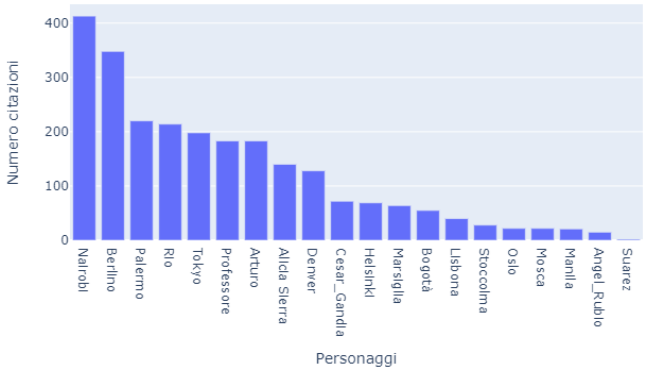

Figura 8. Personaggi citati nei *tweet* italiani

Particolare è il caso di *Tamajo*, che prende il posto del colonnello *Alfonso Prieto* per condurre le operazioni per la rapina alla Banca di Spagna ed è citato solo nei *tweet* spagnoli.

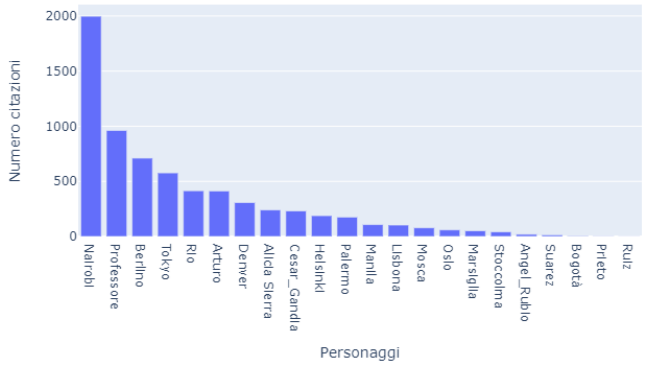

Figura 9. Personaggi citati nei *tweet* inglesi

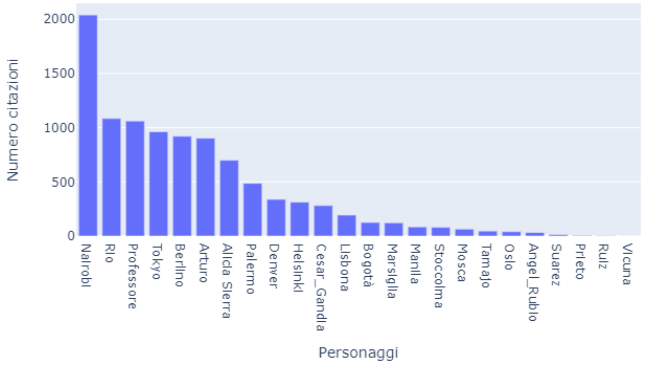

Figura 10. Personaggi citati nei *tweet* spagnoli

Le analisi preliminari si concludono con la *wordcloud*, creata utilizzando i *tweet* in lingua inglese, per far risaltare le parole più utilizzate dagli utenti. Come ci si aspettava emergono termini come *lacasadepapel*, *moneyheist* e termini relativi ai personaggi della serie tv:

Figura 11. *Wordcloud tweet* inglesi.

5. Sentiment Analysis

5.1 *Tweet* in lingua inglese

La *Sentiment Analysis* sui *tweet* in lingua inglese è stata effettuata utilizzando il lessico *VADER* [5]. *VADER* (*Valene Aware Dictionary for Sentiment Reasoning*) è un modulo utilizzato per la *sentiment analysis*, specialmente sui *social media*, ma adatto anche ad altri domini. Contiene un dizionario di parole già annotate come positive e negative che, in combinazione con un set di regole lessicali, permette di sapere non solo se una frase contiene un *sentiment positivo o negativo*, ma anche quanto è positivo o negativo tale *sentiment*, restituendo un punteggio compreso tra -1 e +1.

Watching Nairobi take a bullet through the head broke me. Fuck you to the writers of money heist [**-0.8689**]
We will always miss u Nairobi. [**-0.1531**]
For Nairobi #moneyheist #lacasadepapel [**0.0**]
Nairobi was my only favourite character [**0.25**]
Nairobi is the best character on lacasadepapel [**0.6369**]

Tabella 1. Esempi di *Sentiment Analysis* applicata ai *tweet* in lingua inglese

Il punteggio restituito è stato classificato nel seguente modo:

- -1 $\leq$ *sentiment* $\leq$ -0.5: fortemente negativo
- -0.5 $<$ *sentiment* $<$ 0: negativo
- *sentiment* = 0: neutro

[5] https://github.com/cjhutto/vaderSentiment

- $0 < sentiment < 0.5$: positivo
- $0.5 \leq sentiment \leq 1$: fortemente positivo.

Una volta effettuato il *preprocessing* del testo sui *tweet* relativi ai personaggi citati, è stata svolta la *Sentiment Analysis*. Dato che la frequenza di citazione è diversa per ogni personaggio, è stata effettuata una normalizzazione in modo da effettuare un corretto confronto. I risultati ottenuti sono riassunti nella seguente figura:

Figura 12. *Sentiment Analysis tweet* inglesi

Come si evince dalla Figura 12, tra i personaggi meno apprezzati dagli utenti c'è *Arturo*, che racchiude in sé tutti i difetti, debolezze e paure dell'uomo medio. Interessante è il caso di *Nairobi* in quanto ci si aspetterebbe un *sentiment* fortemente positivo/positivo, dato lo svolgimento degli eventi che l'hanno vista protagonista, ma, poiché i *tweet* associati a *Nairobi* contengono termini quali *kill* e *death* (dato l'epilogo triste di questo personaggio al termine della quarta stagione), il punteggio relativo al suo *sentiment* ne risulta influenzato.

5.2 *Tweet* in lingua italiana e spagnola

A causa della mancanza di lessici validi per la lingua italiana e spagnola, si è deciso di utilizzare la funzione di traduzione presente nel pacchetto *googletrans* [6] di *Python* per tradurre tali *tweet* in lingua inglese ed applicare poi il lessico *VADER* per effettuare la *Sentiment Analysis*. Come dimostrato in letteratura, la traduzione da una lingua alla lingua inglese tendenzialmente non causa il cambiamento del *sentiment* associato alla frase, ma vi può essere una perdita di accuratezza fino ad un massimo del 16%. Questa imprecisione è riscontrabile nell'incremento del numero di *tweet* a cui è associato una *sentiment* neutra.

5.2.1 *Tweet* in lingua italiana

I risultati ottenuti dopo aver tradotto i *tweet* in lingua inglese sono i seguenti:

Sappiate che avete tutto il mio odio profondo #lacasadepapel [**-0.5719**]
Vorrei solo sottolineare il fatto che anche questa volta è colpa di Tokyo, se non avesse lasciato Rio sull'isola tutto questo non sarebbe successo [**-0.0516**]
Ho visto l'episodio S04E06 di Money Heist! #lacasadepapel [**0.0**]
Comunque io aspetto solo che Denver e Stoccolma tornino insieme nella quinta stagione e adottino Rio [**0.1779**]
Non so voi ma la amo #aliciasierra #lacasadepapel [**0.7783**]

Tabella 2. Esempi di *Sentiment Analysis* applicata ai *tweet* in lingua italiana

Figura 13. *Sentiment Analysis tweet* italiani

Tra i personaggi meno amati vi sono *Arturo*, *Gandia* e *Mosca*, mentre i personaggi più apprezzati dagli utenti dei *tweet* in lingua italiana sono *Berlino*, *Manila*, *Lisbona* e *Bogotà*.

5.2.2 *Tweet* in lingua spagnola

I risultati ottenuti traducendo i *tweet* dalla lingua spagnola alla lingua inglese sono i seguenti:

[6] https://pypi.org/project/googletrans/

Tokio ql aweona quiere cagar todo el plan #lacasadepapel. [**-0.5574**]

Ya no voy a acceptar ninguna propuesta de matrimonio sin una escena como esta. He dicho. #lacasadepapel [**-0.1803**]

Acabo de ver el episodio s04e02 de money heist #lacasadepapel [**0.0**]

Igual esta buena la mina en jefe de la policía #lacasadepapel [**0.4404**]

Termine de ver la segunda temporada de #lacasadepapel que genial final wn la raja [**0.6249**]

Tabella 3. Esempi di *Sentiment Analysis* applicata ai *tweet* in lingua spagnola

Figura 14. *Sentiment Analysis tweet* spagnoli

Tra i personaggi meno apprezzati ci sono *Gandia*, *Mosca* e *Arturo*, mentre il più amato dagli utenti dei *tweet* in lingua spagnola è *Helsinki*.

6. Varietà

6.1 *Scraping*

Per ottenere i dati riguardanti la biografia di ogni attore della Casa di Carta è stato effettuato lo *scraping*[7] delle relative pagine di *Wikipedia* utilizzando la libreria *BeatifulSoap* di *Python*. I dati relativi ai premi, ai film e alle serie tv in cui l'attore ha recitato sono stati invece ottenuti utilizzando l'estensione *Web Scraper* di *Google Chrome*.

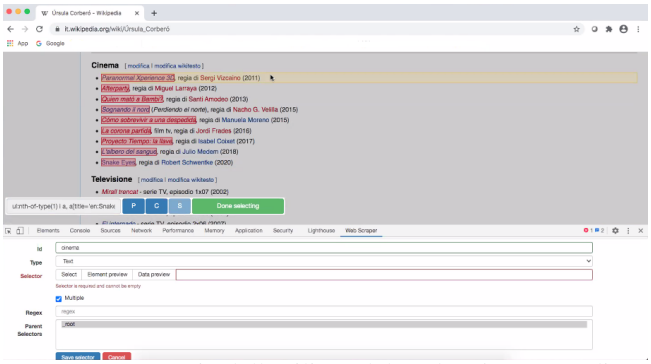

Figura 15. Esempio dell'utilizzo del *tool Web Scraper* di *Google Chrome*

Sono stati così ottenuti i seguenti *file csv*:

- Attori.csv
- Riconoscimenti.csv
- Cinema.csv
- Tv.csv

E' stato inoltre necessario effettuare delle operazioni di *Data Manipulation* in modo da ottenere dei dati in un formato adatto a *Neo4J*.

6.2 Integrazione - NEO4J

Per aggregare tutte le precedenti informazioni si è scelto di utilizzare *Neo4j* [8]. Il modello a grafo è costituito dai seguenti elementi:

- **Nodi**: nome_attore, cinema, riconoscimenti, TV, personaggio, tweet
- **Relazioni**: *ha_recitato_in* (che collega nome_attore e cinema), *ha_ricevuto_il_premio* (che collega nome_attore e riconoscimenti), *ha_recitato_nel_programma_tv* (che collega nome_attore e TV), *ha_interpretato_il_ruolo_di_ne_LCDP* (che collega nome_attore e Personaggio) e *citato_nel_tweet* (che collega tweet e Personaggio).

[7] tecnica informatica di estrazione di dati da un sito web

[8] `https://neo4j.com/`

La struttura a grafo si mostra estremamente comoda per la visualizzazione dei dati integrati ed efficiente nell'eseguire *query* complesse e dispendiose poiché in un database a grafo tutti i nodi sono già connessi. Un possibile esempio di *query* che mostra la flessibilità del linguaggio di interrogazione di grafi *Cypher* è la seguente:

```
MATCH (a:nome_attore)-[]->()-[s]->(t)
RETURN a.name, COUNT(s) AS nr_citazioni
ORDER BY nr_citazioni DESC
```

Il risultato restituito è il conteggio, in ordine decrescente, del numero di citazioni presenti nei *tweet*, associate al nome dell'attore che interpreta un determinato personaggio. *Neo4j* consente inoltre di recuperare velocemente i dati grazie alle funzionalità di *pattern matching* e attraversamento di nodi.

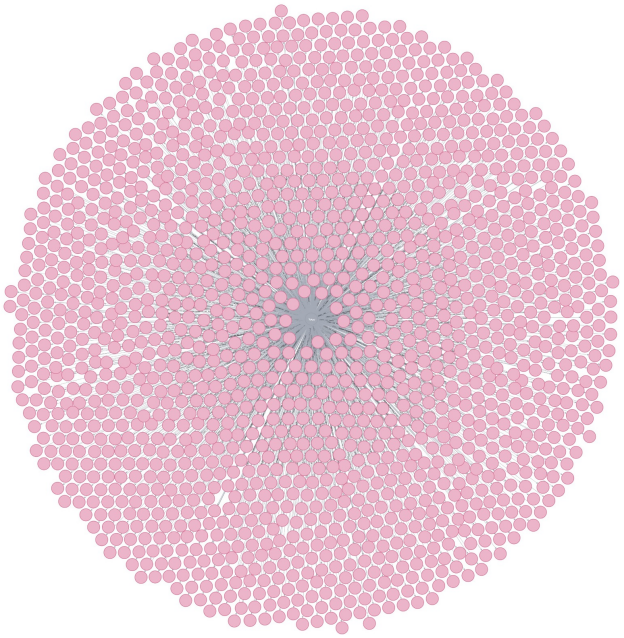

Figura 16. Grafo in *Neo4J* dei *tweet* in cui è citata *Tokyo*

7. *Score* di importanza dei personaggi

Lo *score* di importanza dei personaggi è stato creato considerando il numero dei *tweet* in lingua inglese riferito ad ogni personaggio, il corrispondente punteggio di *Sentiment Analysis* e il numero di premi, riferiti alla *Casa De Papel*, conferiti all'attore dalla critica. In particolare per calcolare questo punteggio è stata utilizzata la seguente formula:

$$Score = NTFP * 1 + NTP * 0.5 + NTN * 0.5 + NTFN * 1$$

dove:

- NTFP = Numero di Tweet Fortemente Positivi
- NTP = Numero di Tweet Positivi
- NTN = Numero di Tweet Negativi
- NTFN = Numero di Tweet Fortemente Negativi

Questo punteggio, nel caso in cui l'attore abbia ricevuto dei premi, deve essere moltiplicato per due volte il numero di premi ricevuti; altrimenti resta invariato. Dunque l'importanza di un personaggio dipende sia dal numero di *tweet* che si riferiscono a lui, in quanto significa che ha preso parte ad eventi chiave all'interno della quarta stagione, sia, in misura maggiore, dal fatto che abbia ricevuto un premio. Ciò significa che esperti del settore, il cui giudizio risulta sicuramente essere più strutturato e tecnico, hanno valutato che quell'attore ha ricoperto un ruolo di spicco all'interno della *serie tv*. Questo permette di definire quali siano i personaggi più importanti della quarta stagione della Casa di Carta.

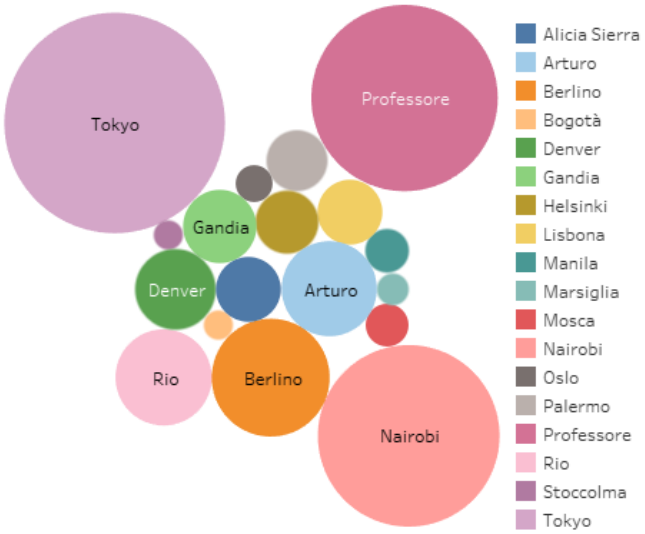

Figura 17. Lo *score* calcolato è proporzionale all'area delle *bubble* nel grafico: maggiore è l'area, maggiore è l'importanza associata al personaggio.

I personaggi più importanti nella quarta stagione della *Casa de Papel* risultano essere *Tokyo* (*score* = 1.338), il *Professore* (*score* = 956) e *Nairobi* (*score* = 908).

8. Conclusioni

La scelta di svolgere un progetto sulla *Casa De Papel* è legata alla curiosità di analizzare una serie tv così famosa da diventare un vero e proprio fenomeno mondiale, soprattutto durante il periodo di *lockdown* in cui le persone disponevano di più tempo da dedicare alla visione di serie tv.
La *Sentiment Analysis* ha evidenziato i personaggi preferiti e quelli meno apprezzati dagli utenti di *Twitter*, che rispecchiano le interpretazioni ed i comportamenti dei personaggi durante la quarta stagione della *Casa De Papel*.
La varietà dei dati raccolti ha inoltre permesso il calcolo di uno *score* di importanza dei vari personaggi, ottenuto tenendo conto di un numero maggiore di fattori.

Codice

L'intero codice implementato è disponibile al link:
`https://github.com/msavi1/Data-Management`

Riferimenti bibliografici

[1] Kafka. Kafka-python documentation. URL: `https://kafka-python.readthedocs.io/en/master/index.html`

[2] Twitter. Tweepy. URL: `https://www.tweepy.org/`

[3] MongoDB. Pymongo. URL: `https://pymongo.readthedocs.io/en/stable/`

[4] C.J Hutto. VaderSentiment. URL: `https://github.com/cjhutto/vaderSentiment`

[5] Alberto Poncelas, Pintu Lohar, Andy Way; *The Impact of Indirect Machine Translation on Sentiment Classification*, 2020.

[6] Thet Thet Zin, University of Computer Studies (Thaton),Faculty of Computer Science,Yangon,Myanmar; *Sentiment Polarity in Translation*, 2020

[7] Neo4J. Neo4J documentation. URL: `https://neo4j.com/docs/`

[8] Web Scraper. URL: `https://webscraper.io/tutorials`

[9] BeautifulSoap. URL: `https://pypi.org/project/beautifulsoup4/`