Analisi dei tweet dei talk politici: prima e durante la crisi di governo

Bellotti Lorenzo* - 795192 - l.bellotti5@campus.unimib.it
Mera Ivan*- 783086 - i.merafranco@campus.unimib.it
Tosi Giovanni* - 873263 - g.tosi12@campus.unimib.it
*Equal contributors

Abstract

Negli ultimi anni Twitter, così come altri social network, sta diventando degli strumenti sempre più utilizzati per il dibattito delle questioni politiche e sociali. L'interazione tra il pubblico, con post e/o foto sui social, e i programmi in diretta è molto comune. Sulla base di questo si è deciso di studiare se ci fosse un legame tra le interazioni su Twitter e gli ascolti televisivi. Per lo sviluppo di questo lavoro si è utilizzata un'architettura Lambda che ha permesso lo streaming dei tweet durante la diretta dei programmi oggetto d'indagine. Ai dati ottenuti sono state applicate tecniche di data preparation al fine di sfruttarli per arricchire i dati di ascolto Auditel. Si è anche deciso di applicare tecniche di sentiment analysis per analizzare l'umore del pubblico di ogni programma. Infine si è deciso di visualizzare i risultati ottenuti mediante Tableau.

CONTENTS

1 INTRODUZIONE

Nel corso degli ultimi anni Twitter è diventato uno tra i più importanti strumenti di condivisione sia per le persone comuni sia per quanto riguarda i soggetti politici, i quali hanno accentrato la comunicazione di proposte e idee politiche online. Inoltre il fenomeno di condivisione di opinioni da parte del pubblico della tv generalista, durante la diretta, è sempre più diffuso e Twitter rappresenta uno degli strumenti principali per questo genere di interazioni. A causa di questi cambiamenti abbiamo deciso di focalizzare il nostro progetto sull'analisi dei tweet durante la messa in onda dei principali talk politici, trasmessi in prima serata sulle reti in chiaro della tv, in relazione ai dati di share e amr di tali programmi. In particolare i talk show oggetto della nostra indagine sono:

- **Carta bianca**
- **Di martedì**
- **Fuori dal coro**
- **Quarta repubblica**
- **Dritto e rovescio**
- **Piazza pulita**

Per approfondire il tema dell'attuale crisi politica, che ha coinvolto il governo italiano, abbiamo deciso di utilizzare come periodo di riferimento le settimane comprese tra l'11 e il 24 gennaio durante le quali l'Esecutivo ha perso la fiducia in Parlamento.

2 ARCHITETTURA

Per quanto concerne lo sviluppo del sistema che ha permesso di eseguire lo streaming dei tweet durante le dirette, si è utilizzato un'architettura Lambda che sfrutta Kafka, Python, con le relative librerie tweepy, pymongo e kafka-python e le API Twitter. Per evitare problemi di *hard-coding*, si è fatto uso di diversi *Config_File* in formato yml in modo da poter automatizzare il più possibile l'esecuzione delle pipeline.

```
########################--- 0. CONFIGURATION ---########################
general:
  pgr_data: programmi.xlsx
  ck: uxwLWE19KK5qW2a4YeqxhPb7d
  cs: T49ip2UVkaWFqbPNXWicqfCoMNC7fjmtIYUdwueU4SPVb89uIs
  at: 1295319664038490118-75dEUCA2IDxPvetZU0Hf9GxkdxwMih
  ats: lmbNbzdzADNR4MCmBSVqfo5XAYD2NuHz3lzxdlCSlZSXI
  hours_gtm: 1
  shr_fasce: ["SHR giovani 15-24", "SHR Eta'(25/34)",
                                   "SHR Eta'(35/44)", "SHR Eta'(45/54)",
                                   "SHREta 55/64", "SHR Individui 65+"]
  amr_fasce: ["AMR giovani 15-24", "AMR Eta'(25/34)",
                                   "AMR Eta'(35/44)", "AMR Eta'(45/54)",
                                   "AMREta 55/64", "AMR Individui 65+"]


pgr1:
  programma: quarta repubblica
  tipologia: attualità
  link: https://www.superguidatv.it/programmazione-canale/oggi/guida-programmi-tv-rete-4/189/
  topic: qr
  delta_before: 2
  leng: it

pgr2:
  programma: fuori dal coro
  tipologia: attualità
  link: https://www.superguidatv.it/programmazione-canale/domani/guida-programmi-tv-rete-4/189/
  topic: cb
  delta_before: 30
  leng: it
```

Figure 1. Config_File

Di seguito il dettaglio delle varie componenti:

2.1 Scraping

Per effettuare lo scraping si è creato il notebook *1_scraping_pgr_orari.ipynb* utilizzando la libreria BeautifulSoup di Python. In questo modo si sono ottenuti i dati relativi ai programmi. Si è deciso di utilizzare come pagina *www.superguidatv.it* in quanto fornisce la programmazione di tutti i canali con la stessa struttura. Per ogni giornata di streaming si scaricavano i dati relativi ai programmi in onda. Bisogna impostare nel *Config_File* il nome del programma, la tipologia e il link da dove

scaricare i dati. La pipeline è fatta in modo da filtrare la programmazione e ottenere i soli programmi di interesse. Una volta eseguito lo scraping si ottengono tre attributi: nome programma, durata con la tipologia e l'orario di inizio. Partendo da questi tre attributi si sono creati altri come l'hashtag ufficiale del programma, l'orario di fine e il giorno di trasmissione. Il file così ottenuto sarà utilizzato dal producer di Kafka. Di seguito un esempio di dataframe in output dal notebook:

Programma	Orario inizio	Orario fine	Giorno settimana	Hashtags
Quarta repubblica	2021-02-08 21:25:00	2021-02-09 00:51:00	Lunedì	quartarepubblica

Figure 2. Dati di scraping

2.2 Kafka

Kafka è una piattaforma streaming distribuita che permette di archiviare ed elaborare flussi di record in tempo reale. All'interno di Kafka agiscono principalmente tre attori: producer, consumer e il cluster. Il primo si occupa di pubblicare e scrivere messaggi sui topic, il secondo è incaricato di leggere i messaggi da un topic a cui si è iscritto, mentre il terzo utilizza tecniche di message queuing per organizzare i messaggi nei vari topic. Si è impostato un producer, *2_Producer.ipynb* che legge alcuni dati dal *Config_File* mentre altri dal file di scraping. Dal *Config_File* il producer legge le chiavi di Twitter API, quanti minuti prima dell'evento iniziare lo streaming e la lingua dei tweet. Dal file di scraping prende il nome del programma da aggiungere come attributo ad ogni tweet. Abbiamo configurato il consumer, *3_Consumer.ipynb* che ascolta i messaggi in arrivo dal topic per tutta la durata della trasmissione. Quando arriva un messaggio modifica l'orario di pubblicazione del tweet. Di default viene considerata l'ora solare con fuso orario inglese, quindi bisogna aggiungere 1 ora per uniformarlo all'ora italiana. Sulla base del processo appena descritto, si sono eseguiti in modo simultaneo producer e consumer, in funzione della programmazione settimanale dei talk da noi selezionati. Ogni coppia di producer/consumer fa riferimento, quindi, ad un topic diverso, ognuno dei quali è associato al programma.
Questa implementazione ci ha permesso di ottenere un'architettura scalabile che permetta di eseguire lo streaming contemporaneamente su diversi programmi in onda su emittenti diverse nella stessa fascia oraria.

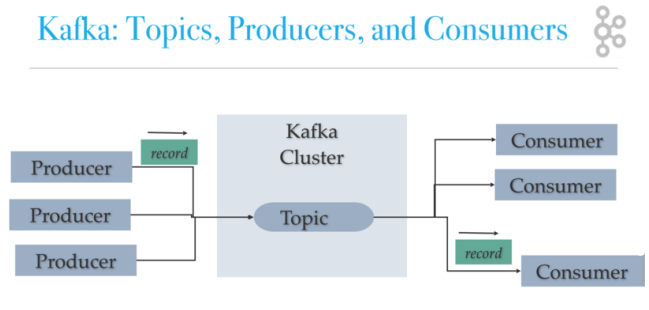

Figure 3. Kafka

2.3 MongoDB

Per archiviare i tweet letti dal consumer si è deciso di utilizzare MongoDB, un database non relazionale (NoSQL) di tipo documentale. Il motivo di tale scelta è che i tweet sono già in formato JSON e quindi è molto semplice memorizzare gli oggetti senza modificare la loro struttura nelle collection del db. Considerato che il team era composto da tre persone e i programmi erano sei, si è deciso che ogni membro del team avrebbe creato un database contenente due collection, una per ogni talk a lui assegnato. L'archiviazione viene effettuata sfruttando la libreria Python *PyMongo*. Essa permette in maniera molto semplice e intuitiva di interfacciarsi con MongoDB e archiviare i tweet con la funzione *insert_one*. Prima di esportare i dati così ottenuti, si è deciso di effettuare alcune operazioni di pulizia. In questo modo, riducendo la dimensione dei file, si velocizza il caricamento dei JSON su Python. Sulla base di conoscenze pregresse, si sono ripuliti i dati duplicati facendo riferimento alla chiave *user_id* e *text*, è risaputo infatti che uno stesso utente pubblica più volte lo stesso tweet. Inoltre sono stati eliminati i tweet fuori dall'orario d'interesse. Per avere dati uniformi i tweet devono iniziare alle 21.20 e finire alle 23.59.
Per ogni collection sono stati eliminati tweet nell'ordine delle migliaia. Queste due operazioni sono contenute nel notebook *4_preprocessing_mongodb.ipynb* e sono state eseguite con l'utilizzo di *PyMongo*. A questo punto si sono esportati molto comodamente i dati da MongoDB alla VM tramite il seguente comando:
mongoexport –authenticationDatabase=admin –authenticationMechanism=SCRAM-SHA-1 –username=user –password= -d database -c collection -o path_file
Successivamente i file sono stati trasferiti in locale tramite il protocollo *scp* per le fasi successive.

3 Data Quality

Con i dati JSON in locale si è utilizzato il notebook *json_csv_xlsx* per appiattire i dati. La prima operazione consisteva in caricare ogni riga del JSON come elemento di una lista Python. Successivamente per ogni elemento nella lista si andava a ricavare solo alcuni attributi utili al nostro obiettivo.
Uno degli attributi più importanti è stato l'utente che viene menzionato in un tweet,*user_men* . Si è deciso infatti di tenere solo i tweet in cui venivano menzionati gli ospiti di un determinato programma. Questa scelta è stata fatta a seguito di un controllo sugli utenti menzionati nei tweet. La maggior parte dei tweet infatti, non aveva legami con il talk ma erano legati ad altri programmi in onda in quel momento.
Questo attributo viene letto da Python come una stringa. La prima operazione da fare dunque è stata quella di trasformarlo in una lista di valori.
Una parte degli utenti ha menzionato più persone contemporaneamente perciò bisognava trasformare questo attributo da lista a valori unici per ogni record. Python permette di effettuare ciò molto rapidamente tramite la funzione *explode*. Questo fa si che si creino dei duplicati se un utente menziona almeno due persone. In ogni caso

non rappresenta un problema in quanto l'attributo *_id* ci permette di identificare univocamente ogni tweet.
Sull'attributo *user_men* vengono eseguite alcune operazioni di normalizzazione come la rimozione di spazi, lowercase e rimozione di eventuali emoticons.
Infine vengono tenuti solo i tweet dove veniva menzionato almeno uno degli ospiti. Questo ha portato ad una riduzione considerevole del numero di tweet: per Quarta repubblica per esempio si passava da più di 300 mila tweet a circa 1800. Se da una parte si perdono dati, dall'altra si è sicuri che i tweet rimasti sono vere interazioni tra il pubblico e il talk in onda.
Visto che la maggior parte degli ospiti è sempre fisso, si è deciso di creare un *Config_File* che contenesse una lista degli ospiti per ogni programma.
Il risultato così ottenuto poteva essere utilizzato per le operazioni *core* di questo progetto.

4 Operazioni effettuate

Le due operazioni core del progetto sono state l'arricchimento dei dati di ascolto e la sentiment analysis. Di seguito il dettaglio delle operazioni.

4.1 Relazione Twitter - Ascolti

La società UPA ha gentilmente concesso dati sull'audience dei talk in esame, in particolare ci sono state fornite misure riguardanti lo share e l'AMR ufficiali di Auditel. [1] [2] I dati sono stati forniti in formato *.xlsx* e sono suddivisi per fasce d'età e sesso. Considerato che per possedere un profilo Twitter l'età minima è 14 anni, si è deciso di tenere i dati degli individui con almeno 15 anni.
Anche questo è stato fatto mediante l'uso di un parametro sul *Config_File*.
Tra le fasce di età rimaste si è calcolato la mediana in quanto più robusta agli outlier. Di seguito alcuni degli attributi del file di UPA:

Nome programma	AMR Bambini 4-14	AMR giovani 15-2	AMR Eta'(25/34)	AMR Eta'(35/44)	AMR Eta'(45/54)	AMREta 55/64	AMR Individui 65+
Quarta repubblica	17.941	31.102	24.932	73.669	150.139	188.978	549.582
Fuori dal coro	12.642	22.316	22.336	88.323	166.140	196.635	540.568
Dritto e rovescio	48.586	41.388	38.858	85.939	210.842	235.886	694.912
Carta bianca	13.057	23.324	30.982	77.761	165.978	261.731	675.878
Piazza pulita	22.733	39.026	53.517	123.940	186.584	315.338	721.313
Di martedì	27.899	58.436	69.272	115.532	225.725	331.552	761.549

Figure 4. Ascolti

Questi dati sono stati arricchiti con il numero di tweet unici per ogni talk. Il merge è stato effettuato sul nome del programma. Per garantire una maggiore robustezza durante la fase di merging, vengono eseguiti controlli sulla variabile di join che in questo caso è il nome del programma.

1. Share: è il rapporto tra il TTS del programma e il TTS del Totale TV in termini percentuali, dove il TTS è la somma del tempo che ciascun individuo ha dedicato a guardare i programmi.
2. AMR: è il numero medio di ascoltatori sull'insieme di ascolti, ovvero la media del numero di ascoltatori sui singoli minuti che compongono un programma.

4.2 Sentiment Analysis

La sentiment analysis è il campo del NLP che permette l'identificazione, l'estrazione e la valutazione di opinioni da un testo. Lo strumento utilizzato per calcolarla in questo progetto è VADER, una libreria Python "lexicon e ruled-based" particolarmente utile per l'analisi dei social in quanto considera anche le emoji che spesso sono utilizzate per esprimere un sentimento. Applicando l'algoritmo di VADER a un testo si ottengono 4 risultati: pos, neg e neu che stanno ad indicare quanto sia positivo, negativo o neutro la stringa analizzata (i valori sono compresi tra 0 e 1) mentre il compound è un indicatore più generale della valutazione che varia da -1 (molto negativo) a +1 (molto positivo). Nel progetto si è scelto di usare quest'ultimo come metrica di valutazione del sentiment. Sebbene VADER sia la scelta migliore quando si lavora con testi provenienti dai social presenta alcuni problemi. Un problema riguarda le emoji: VADER può gestirle, ma non ha un vocabolario apposito in cui ad ognuna di esse è associato un valore, bensì traduce il simbolo con la sua descrizione testuale e poi calcola il punteggio. Questo porta nella maggior parte dei casi ad ottenere un punteggio non realistico con il significato del testo.
Il secondo problema di VADER è che non è in grado di gestire testi in lingue diverse dall'inglese. La soluzione che propone è quella di collegarsi direttamente a Google Traduttore attraverso un wrapper e tradurre il testo.
Questa soluzione non è stata percorribile in quanto il server di Google ha un limite massimo richieste che un IP può effettuare. Per questo motivo si è deciso di utilizzare un tool online che permettesse di tradurre velocemente i tweet.
Con i tweet tradotti è stata applicata la sentiment analysis sui tweet tradotti in lingua inglese.

5 Risultati

Per visualizzare i risultati dei dati ottenuti si è deciso di utilizzare il software Tableau. Sono state create 4 visualizzazioni che permettono di capire velocemente i dati dello streaming. Di seguito la prima visualizzazione:

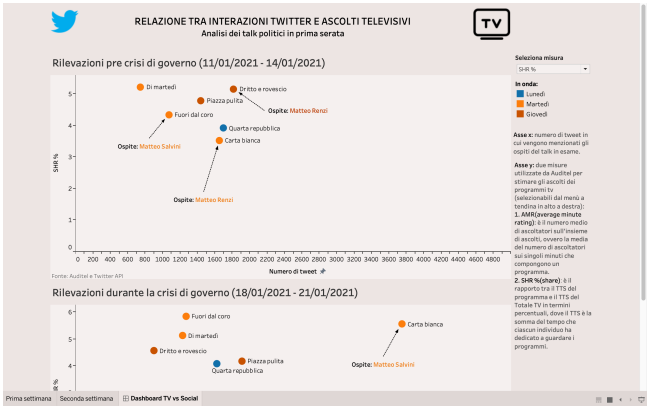

Figure 5. Relazione Twitter - Ascolti

Nella prima visualizzazione si riesce a studiare la relazione che intercorre tra il numero di interazioni Twitter

e le due misure di ascolti scelte.
Si possono fare diverse considerazioni. Si può notare come ogni qual volta che c'è tra gli ospiti un politico di rilievo il talk in questione ottiene il maggior numero di interazioni su Twitter. Per quanto riguarda un confronto tra le due settimane di studio, si evince un aumento per tutti i talk di share e AMR. Si evidenzia come Salvini, essendo l'unico politico ospite di rilievo abbia avuto un impatto degno di nota nelle interazioni di Carta bianca.

Di seguito viene mostrato l'andamento dei tweet nei programma in cui sono stati ospiti Renzi e Salvini. Sebbene l'intervento di Renzi sia stato inferiore rispetto a Salvini, è riuscito a far concentrare in un minuto un maggior numero di tweet. Difatti nel complesso in quella puntata *Carta bianca* aveva ottenuto più interazioni rispetto a *Fuori dal coro* dove Salvini era ospite.

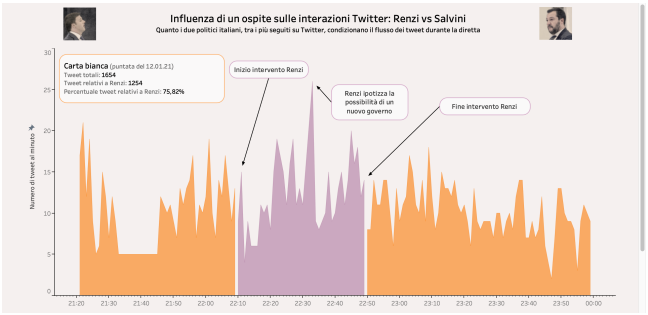

Figure 6. Andamento tweets con menzioni a Renzi

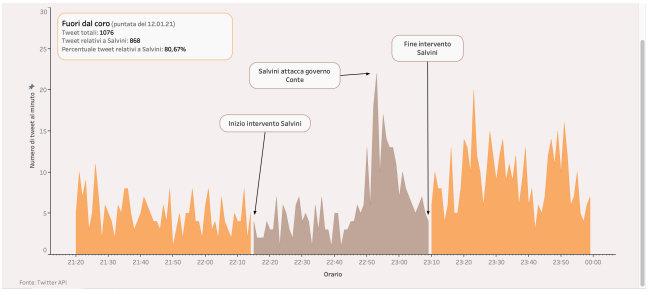

Figure 7. Andamento tweets con menzioni a Salvini

Nella terza e ultima visualizzazione si mostra il sentiment associato ad ogni talk. Si è deciso impostare l'analisi a livello di programma anziché a livello di ospite in quanto alcuni ospiti, come Salvini o Renzi, catturavano più del 75% delle interazioni.
Dall'analisi di questi grafici si evidenzia come *Quarta repubblica* abbia ottenuto una percentuale altissima di sentiment neutro la prima settimana, più del 70%. Dall'altra parte, durante la seconda settimana, *Di martedì* ha ottenuto un elevato livello di sentiment positivo, più del 51%.

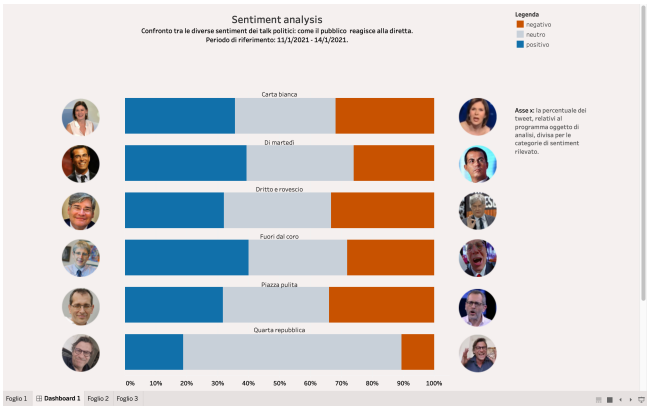

Figure 8. Sentiment settimana prima della crisi di governo

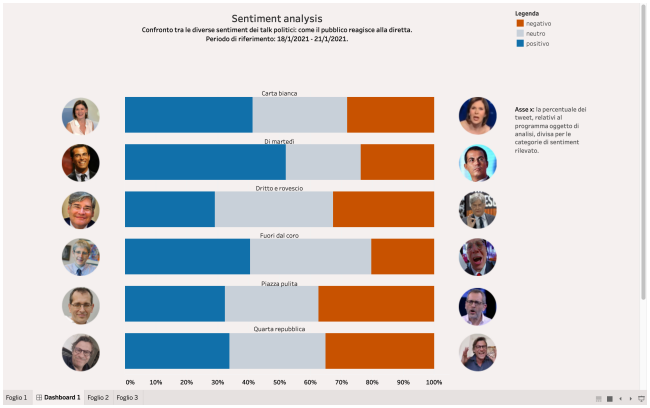

Figure 9. Sentiment settimana durante della crisi di governo

6 Conclusioni

Lo sviluppo di questo progetto ha portato alla realizzazione mediante l'utilizzo di Kafka di un'architettura Lambda attraverso la quale abbiamo memorizzato il flusso di tweet che hanno come oggetto i principali talk politici in prima serata in un database MongoDB. Una delle difficoltà principali riscontrate consisteva nella contemporaneità di più talk nello stesso giorno e durante la stessa fascia oraria, per ovviare a questa problematica abbiamo sviluppato un file config che ci permettesse di automatizzare lo streaming, evitando di cambiare direttamente il producer e limitandoci a modificare solamente il riferimento alla collezione di destinazione del flusso dei tweet, su MongoDB, e il topic contenuti nel consumer. Tale soluzione ci ha permesso di utilizzare gli stessi producer e consumer contemporaneamente per lo streaming di più programmi. L'obiettivo del nostro lavoro è quello di cercare una possibile relazione tra le interazioni su Twitter con i dati di ascolto relativi ai talk show politici selezionati. Ciò che emerge dalle nostre analisi è che non vi è una relazione diretta tra share/AMR e interazioni Twitter, come da noi inizialmente ipotizzato, ma ci sono più fattori che influenzano l'interazione su Twitter tra i quali si evidenziano: la presenza di ospiti di "peso", come ad esempio Matteo Salvini e Matteo Renzi, i quali hanno influenzato in maniera netta le interazioni Twitter relative al programma in cui erano presenti, come si

evince dai picchi dell'andamento temporale del flusso dei tweet. Un altro elemento rilevante riguarda la situazione politica vigente durante la trasmissione dei talk, che ha coinvolto un maggior numero di spettatori. Infine non è da sottovalutare l'affinità tra lo schieramento politico al quale appartiene l'ospite e il talk e l'orientamento del conduttore, laddove i due sono divergenti abbiamo notato una maggiore interazione su Twitter.

7 RIFERIMENTI

https://www.crummy.com/software/BeautifulSoup/bs4/doc/
https://developer.twitter.com.
https://kafka-python.readthedocs.io/en/master/index.html
https://www.tweepy.org
https://www.mongodb.com/it
https://pymongo.readthedocs.io/en/stable/
https://www.upa.it/it/index.html
https://www.auditel.it/glossario/
https://www.nielsen.com/it/it/solutions/measurement/social-tv/
https://www.tableau.com