Rai social: un'analisi settimanale dei tweet

Alex Ceccotti
Matricola: 790497
CdLM Data Science
Università degli Studi
Milano Bicocca

Pietro De Simoni
Matricola: 790638
CdLM Data Science
Università degli Studi
Milano Bicocca

Gianmarco Stucchi
Matricola: 778772
CdLM Data Science
Università degli Studi
Milano Bicocca

***Abstract*—I social network stanno diventando sempre più rilevanti nella vita di tutti i giorni. Per questo motivo, anche le emittenti televisive devono monitorare le opinioni che gli utenti esprimono in rete durante i propri programmi TV. Per poter analizzare i dati e proporre campagne social adeguate è innanzitutto necessario acquisire e memorizzare in modo appropriato le informazioni grezze disponibili online. Lo scopo di questo progetto è quello di analizzare il seguito social di un'emittente televisiva attraverso la proposta di un'architettura che permetta di gestire le prime fasi del ciclo di vita dei dati (acquisizione, memorizzazione e manipolazione) nell'ambito del social network *Twitter*. In particolare sono stati presi in considerazione i tweet riferiti ai programmi TV *Rai* pubblicati durante la seconda settimana di aprile 2018. È stata infine svolta una breve analisi descrittiva.**

***Index Terms*—Twitter, Kafka. MongoDB, Rai, SocialTV**

INTRODUZIONE

Vista la continua evoluzione dei social network e la loro sempre più costante influenza nel mondo quotidiano, è senz'altro di notevole rilevanza la ricerca di metodi per tenere sotto controllo le informazioni che vengono immesse nel web. *Facebook*, *Twitter* e *Instagram* vantano infatti milioni di utenti attivi e una buona fetta della popolazione mondiale passa gran parte del proprio tempo a consultarli, postare commenti, mettere like ed inserire contenuti multimediali in queste piattaforme [1]. È dunque utile essere in grado di monitorare il flusso di dati di questi social per cogliere lo stato d'animo delle persone ed ottenere un'idea anche in tempo reale di come si evolve l'opinione pubblica riguardo alcuni temi. È stato dunque svolto un progetto che mira a gestire e ad analizzare dati provenienti da *Twitter* con particolare attenzione ai tweet riferiti a programmi TV emessi dai tre canali *Rai* principali: *Rai1*, *Rai2* e *Rai3*.

I. SCRAPING DELLA PROGRAMMAZIONE RAI

Tramite l'estensione *Web Scraper* [2] di *Google Chrome* abbiamo scaricato direttamente dal sito *www.raiplay.it/guidatv* la programmazione per i giorni di nostro interesse. Utilizzando questa estensione è possibile selezionare le componenti di un sito web che si vogliono scaricare. In questo modo, con un metodo molto efficiente, è stato possibile ottenere contemporaneamente l'intera programmazione di tutti i giorni di nostro interesse (dal 4 aprile 2018 all'11 aprile 2018) divisa per canale.
Il csv grezzo così ottenuto è stato successivamente ripulito in modo che ogni riga contenesse solamente il nome del programma, l'orario d'inizio e l'orario di fine. Molti dei programmi così ottenuti erano di poco interesse (ad esempio telegiornali e meteo), così abbiamo proceduto ad eliminarli. Per farlo abbiamo sviluppato un semplice script che per ogni riga dei nostri file csv andasse a leggere il nome del programma e ci chiedesse se tale programma andasse conservato. In caso positivo venivano richieste le keywords relative al programma in questione. Per ogni apparizione successiva alla prima di ciascun programma, lo script ricorda le scelte precedenti e automaticamente aggiunge le keywords relative al programma o lo elimina a seconda del caso. Alla fine di questo processo dunque, abbiamo ottenuto i nostri file csv contenenti i nomi dei programmi, gli orari a cui sarebbero andati in onda e le keywords da monitorare in quegli orari.

II. ARCHITETTURA SVILUPPATA

Una volta pulito il file csv ottenuto tramite *Web Scraper*, siamo passati alla progettazione e alla realizzazione di un'architettura che permetta di prendere in input il file csv contenente la programmazione di uno dei canali *Rai* e che automaticamente vada a scaricare, negli orari previsti, i tweet aventi le keywords selezionate in precedenza. Questi tweet dovranno poi essere opportunamente memorizzati per le future analisi. Abbiamo perciò utilizzato un'architettura *Lambda* tramite *Kafka* implementata nel seguente modo:

1) Un producer che accetti in input l'opportuno file csv e che acceda all'API di *Twitter* [3] per scaricare automaticamente i tweet da inviare ad un topic *Kafka*. I tweet vengono scaricati in formato JSON e dopo una selezione degli attributi da tenere vengono convertiti in stringa per poter essere inviati come messaggio al topic. Il producer rimane in esecuzione dall'orario di inizio del primo evento fino all'orario di fine dell'ultimo evento della settimana. Negli spazi temporali in cui non è presente alcun evento *Rai* rilevante, il producer resta attivo senza però accedere all'API di *Twitter*.
2) Un consumer batch che ascolta i messaggi in arrivo dal topic per tutta la durata della settimana. Quando arriva un messaggio lo converte in formato JSON, modifica l'orario di pubblicazione del tweet (di default viene considerata l'ora solare con fuso orario inglese, quindi bisogna aggiungere 2 ore per uniformarlo con l'ora legale italiana) e lo aggiunge come un nuovo record ad

una collection di *MongoDB*[4] unica per tutti gli eventi dello stesso canale *Rai*, ma diversa per canali diversi.

3) Seguendo questa logica, verranno eseguiti in modo simultaneo tre producer e tre consumer per tutta la durata della settimana. Ogni coppia di producer/consumer fa riferimento ad un topic diverso, ognuno dei quali è associato ad un canale *Rai*. Questa implementazione multipla è il modo che abbiamo ritenuto essere il più efficiente per realizzare un'architettura scalabile che permetta anche di effettuare analisi in real time.
4) In parallelo al consumer batch è stato implementato anche un consumer "stream" che, ascoltando un topic in un dato momento, converte i messaggi in formato JSON ed estrae il valore del campo "hashtag" il quale contiene una lista di dizionari. Dopo qualche semplice manipolazione è possibile estrarre per ogni tweet i relativi hashtag per creare poi un dizionario avente come chiavi gli hashtag utilizzati almeno una volta e come relativi valori il numero di volte che sono stati utilizzati. Viene infine visualizzato a schermo ogni N tweet analizzati la top 10 degli hashtag (vedi Figura 1).

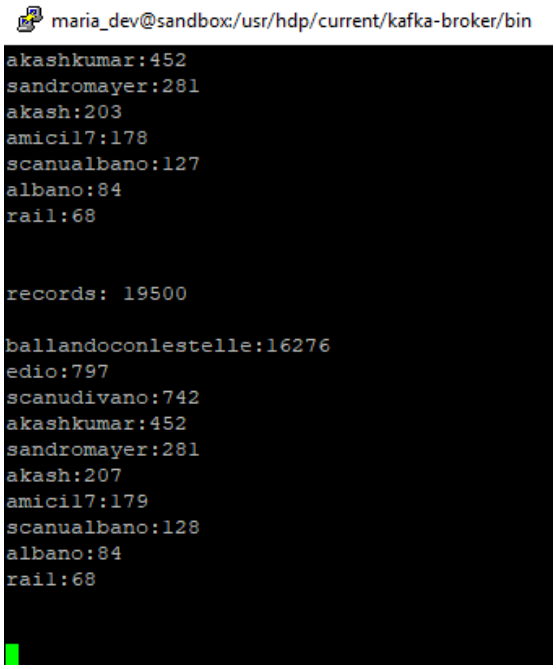

Fig. 1. Hashtag più twittati

5) Sempre facendo riferimento alle analisi real time, l'idea iniziale era quella di visualizzare graficamente l'andamento di alcune keyword predefinite per un dato programma (ad esempio i nomi dei partecipanti ad un talent show). Tuttavia, siamo riusciti a fare ciò solo in locale (e quindi senza architettura *Lambda*, ma scaricando esclusivamente i tweet tramite API) ottenendo come risultato il grafico in Figura 2 (che si aggiorna in tempo reale). Non siamo riusciti a far girare lo stesso script all'interno della nostra architettura definitiva in quanto risulta molto complicato realizzare grafici nella Virtual Machine. A nostro parere tale risultato è comunque un discreto spunto in ottica futura ed è quindi meritevole di nota.

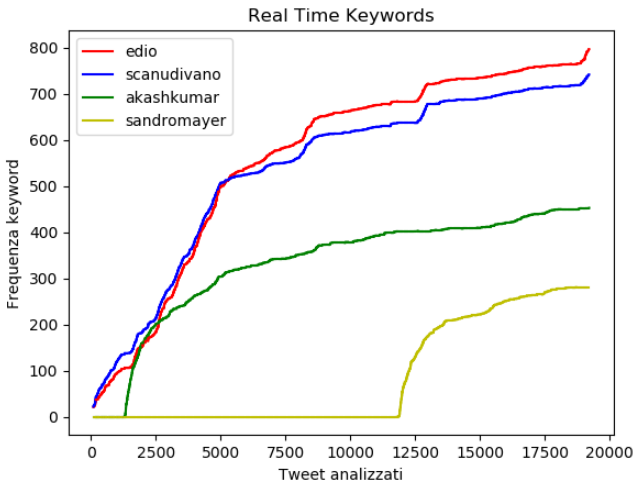

Fig. 2. Andamento keywords durante "Ballando con le stelle"

Nel suo complesso, l'architettura appena esposta può essere rappresentata come in Figura 3. Una volta ultimato il processo di download streaming dei tweet, abbiamo ottenuto un database *MongoDB* contenente tre collection, una per ogni canale *Rai*. La struttura di ogni record è rappresentata dall'esempio in Figura 4. Il campo "_id" è un identificativo univoco generato automaticamente da *MongoDB*, "screen_name" è il nome univoco dell'utente *Twitter*, "orario_fine"/"orario_inizio" è l'orario di fine/inizio dell'evento considerato, "text" è il testo del tweet, "created_at" è l'orario di pubblicazione del tweet, "followers" è il numero di seguaci dell'utente, "hashtag" è una lista di dizionari contenenti posizione e parola utilizzata come hashtag, "nome_evento" rappresenta il nome dell'evento considerato e "friends" è il numero di profili seguiti dall'utente. Ovviamente i campi relativi agli eventi vengono ripetutamente duplicati in molti record. Sebbene questo fatto rappresenti un'inefficienza in termini di spazio di memoria, la nostra struttura di storage rimane comunque appropriata in quanto scalabile e molto efficiente sia in scrittura sia in lettura.

III. Manipolazione ed integrazione dei dati

Per analizzare i dati memorizzati nel database *MongoDB* è possibile procedere in svariati modi. Per esempio, utilizzando *python* è possibile connettersi al database ed effettuare interrogazioni tramite la libreria *pymongo*. Questo metodo è molto rapido, ma lavorando nella Virtual Machine, come già accennato in precedenza, è difficile visualizzare graficamente i risultati delle proprie analisi. Il software *Tableau* [5] consente invece di ottenere rapidamente delle infografiche ed è possibile utilizzarlo da locale connettendosi ad un database remoto. Tuttavia, vista la struttura annidata del nostro ambiente virtuale

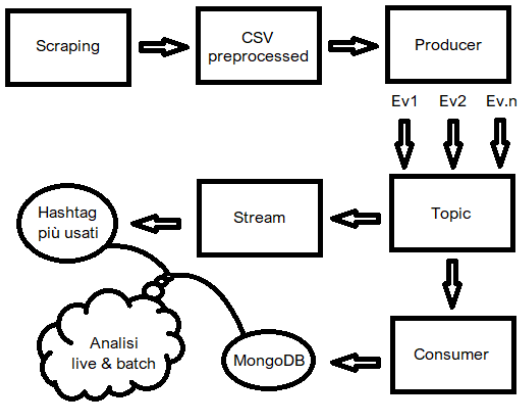

Fig. 3. Architettura sviluppata


```
{
"_id":{"$oid":"5aca78f0e5cde51c38e8e8fe"},
"screen_name":"lunacla74",
"orario_fine":"2018-04-9 00:04:00",
"orario_inizio":"2018-04-8 20:35:00",
"text":"Con i brividi che ballavo e cantavo in piedi sopra i
"created_at":"2018-04-08 22:17:52",
"followers":4799,
"hashtag":[{"indices":[84,96],"text":"miofratello"},{"indice
"keywords":["fazio","chetempochefa","litizzetto","saviano","
"nome_evento":"Che tempo che fa",
"friends":2613
}
```


Fig. 4. Esempio di un record salvato in MongoDB

(Sandbox dentro la Virtual Machine), tale connessione risulta scomoda e complicata. Dal momento che stiamo trattando dati di dimensioni contenute (meno di 20MB complessivi), la via più semplice ed efficiente è quella di esportare le collection in formato JSON e trasferire i file così ottenuti in locale. Questo ci ha permesso di svolgere l'analisi in modo rapido ed efficace. Abbiamo quindi esportato le tre collection tramite il comando "mongoexport" (ad esempio, per *Rai1*: "mongoexport -d prograi -c rai1 -o rai1.json") ottenendo in questo modo i file desiderati. Una volta portati in locale, abbiamo caricato i tre file in *Tableau* e li abbiamo uniti creando così un dataset unico. Da questo dataset sono stati esclusi i campi non atomici ("hashtag" e "keywords") in modo da appiattire i dati senza dover ripetere alcun record due o più volte. Questa scelta è supportata anche dal fatto che l'esclusione di questi attributi non condiziona in alcun modo il raggiungimento degli obiettivi che ci siamo prefissati per l'analisi batch.

In un secondo momento rispetto alla settimana di streaming dei dati, è stato deciso di integrare i dati degli eventi con quelli provenienti dalle pagine ufficiali dei programmi *Rai*. In particolare, sono state estratte da *Twitter* informazioni riguardanti il numero di followers delle pagine ufficiali e quanto queste abbiano twittato nella settimana di riferimento. Sono stati dunque generati tre file csv (uno per ogni canale) che, utilizzando *Tableau*, sono stati uniti alla versione piatta dei file JSON precedentemente considerati. Abbiamo così ottenuto due tabelle (una per i tweet streaming ed una per le pagine ufficiali) legate da una chiave comune, ovvero il nome dell'evento TV; tali tabelle sono state realizzate in modo da poter essere analizzate e visualizzate adeguatamente.

IV. Analisi e risultati

Una prima informazione che possiamo cercare di estrapolare dai dati è l'andamento del traffico *Twitter* (inteso come numero di tweet contenenti specifiche keyword) riferito ai programmi dei tre principali canali *Rai*. Aggregando per ora di pubblicazione e per canale si ottiene rapidamente il grafico in Figura 5. È poi facile andare ad individuare in concomitanza di quali eventi si è rilevato il maggior numero di tweet. In particolare, i programmi TV *Rai* con più seguito social nella settimana considerata sono "The Voice of Italy" (*Rai2*), "Ballando con le stelle" (*Rai1*) e "Che tempo che fa" (*Rai1*). Si registrano anche picchi di dimensioni ristrette per "Porta a Porta" (*Rai1*) e "NEMO" (*Rai2*). Si nota inoltre che per quanto riguarda *Rai3*, i relativi programmi TV non hanno generato molto traffico su *Twitter* se non per un leggero aumento durante la trasmissione "Ulisse".

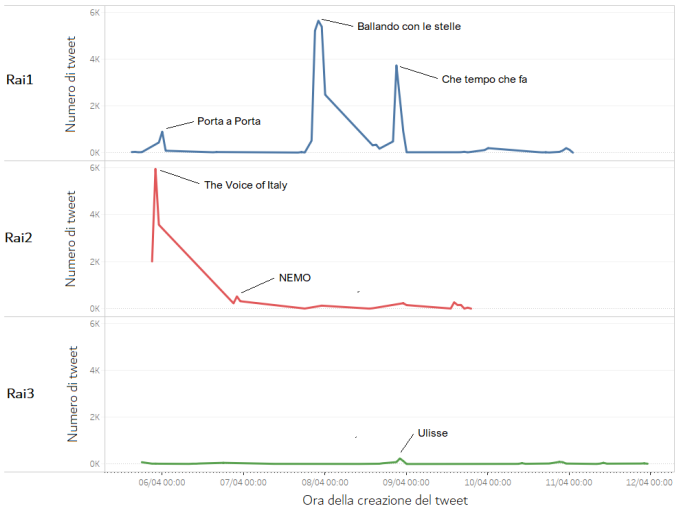

Fig. 5. Andamento orario dei tweet riferiti ai programmi Rai

Grazie all'integrazione dei dati riguardanti le pagine ufficiali, è possibile mostrare la relazione che intercorre tra comportamento dell'emittente sui social e relativa risposta degli utenti in termini di tweet. Come è possibile notare in Figura 6, un buon seguito in termini di followers di una determinata pagina non sempre porta ad un traffico consistente di tweet durante la trasmissione. È ad esempio il caso di *Report*. Possiamo notare invece come *Ballando con le stelle* ha generato molti tweet pur avendo pochi followers e avendo twittato poco nella settimana di riferimento. È interessante evidenziare invece la differenza di traffico generato dalle trasmissioni *Che tempo che fa* e *Che fuori tempo che fa*. Entrambe infatti fanno riferimento alla stessa pagina ufficiale, ma hanno una risposta in termini di tweet dagli utenti molto diversa. Risulta evidente dunque che le attività delle pagine ufficiali hanno un impatto limitato sul seguito social e non

esaustivo a spiegare il traffico *Twitter* generato dal rispettivo programma TV.

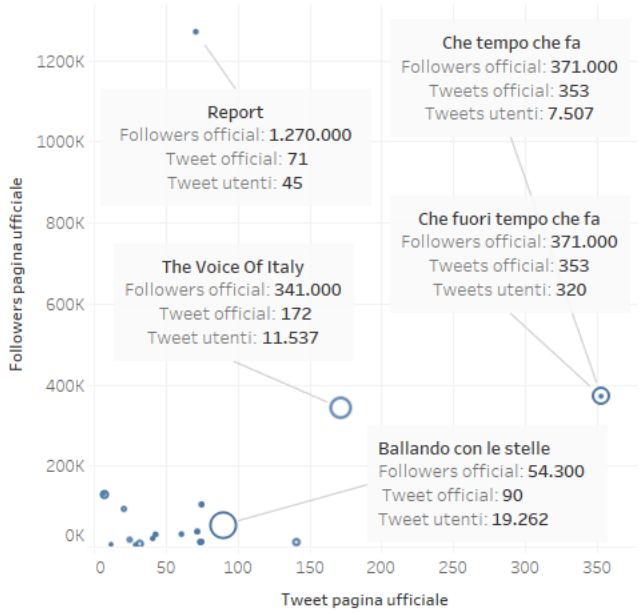

Fig. 6. Relazione tra pagine ufficiali e tweet dell'evento

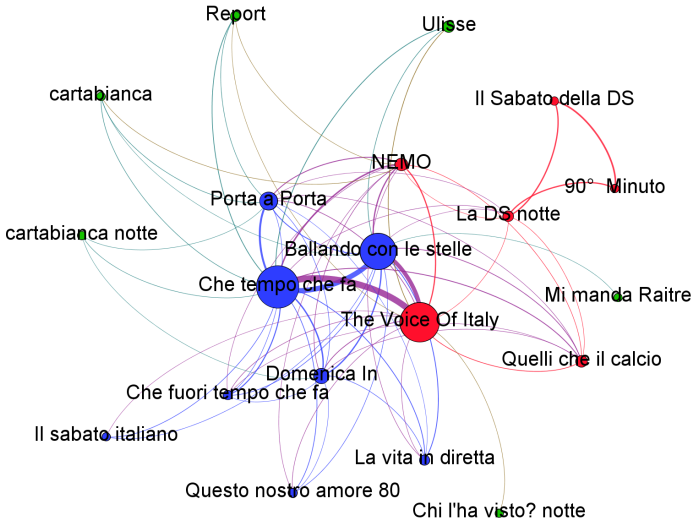

Fig. 7. Grafo eventi

Sulla base del conteggio degli utenti che hanno twittato su più eventi, tramite il software *Gephi*[6], è stato generato il grafico in Figura 7 che mostra la relazione fra eventi trasmessi sui differenti canali. Ogni nodo rappresenta una trasmissione televisiva ed è tanto più grande quanti più tweet ha generato. Il colore dei nodi corrisponde al canale: *rai1* in blu, *rai2* in rosso, *rai3* in verde. Gli archi sono tanto più spessi quanto più sono numerosi gli utenti che hanno twittato su entrambi gli eventi. Inoltre i nodi sono disposti in modo da posizionare vicini fra loro due eventi fortemente coti. Si può notare come *Ballando con le stelle*, *Che tempo che fa* e *The voice of Italy* sono gli eventi che hanno generato più tweet e sono fortemente interconnessi. Si denota un forte legame interno fra i programmi di *rai1* e i programmi di *rai2*. In particolare si osserva una forte connessione fra i tre programmi calcistici trasmessi su *rai2*: *Novantesimo minuto*, *La domenica sportiva* e *La domenica sportiva - notte* Diversamente da quando accade per gli altri canali, i programmi di *rai3* non sono fra loro connessi.

Conclusioni

Lo sviluppo di questo progetto ha portato alla realizzazione di un'architettura *Lambda* tramite l'utilizzo di *Kafka* che consente di memorizzare in un database *MongoDB* il flusso di tweet che hanno come oggetto un determinato programma TV. La caratteristica più complessa e particolare di questa architettura è il modo in cui, ricevendo un file csv in input, riesca a gestire in streaming i tweet di più eventi sequenziali, richiedendo quindi all'utente interessato ad utilizzarla di eseguire una volta sola i file necessari ad avviare producer e consumer batch. Inoltre, avviando il consumer stream, è possibile avere una panoramica live degli hashtag più utilizzati durante un determinato periodo di tempo. Il sistema di storage utilizzato permette anche una semplice e rapida analisi dei dati (sia in ambiente virtuale sia in locale). Da tale analisi si evince che i programmi TV *Rai* che generano più traffico su *Twitter* sono principalmente i talent show e i talk show. I primi infatti richiedono spesso una partecipazione attiva degli spettatori che possono dunque esprimere il proprio voto su *Twitter*, mentre i secondi trattano argomenti che generalmente suscitano interesse nell'utente medio del social network in questione [7]. Si è notato inoltre che il comportamento delle pagine ufficiali di un determinato programma non incide particolarmente sulla popolarità di questo se espressa in termini di tweet. Infine possiamo dedurre che i legami più significativi corrispondono agli eventi *Che tempo che fa*, *Ballando con le stelle* e *The Voice of Italy*, in quanto gli utenti che hanno twittato in uno di questi tre programmi, generalmente lo hanno fatto anche negli altri due. Si osservano altri legami minori tra programmi dello stesso genere, come ad esempio quello sportivo.

References

[1] Diffusione, uso, insidie dei social network: i dati dell'Osservatorio Giovani dell'Istituto Toniolo, 19 gennaio 2017, http://www.rapportogiovani.it/
[2] http:/webscraper.io/
[3] https://apps.twitter.com/
[4] https://www.mongodb.com/
[5] https://www.tableau.com/
[6] https://gephi.org/
[7] Il vero problema di Twitter, 11 febbraio 2014, https://www.ilpost.it/2014/02/11/twitter-mainstream/