

Progetto del Software AA 2021

Marcello Missiroli, Paolo Ciancarini

2° semestre AA 2021

1 Introduzione

Vogliamo costruire uno strumento di raccolta e analisi di informazioni diffuse sul territorio e geolocalizzabili, quali ad esempio i tweet di Twitter. Queste informazioni possono riferirsi ad una persona, ad un luogo, ad un evento.

Le persone che vivono un'emergenza e la comunicano attraverso i social media quali Facebook, Instagram, Twitter, eccetera possono essere considerate dei sensori umani sul territorio e i messaggi che si scambiano generano, se aggregati in tempo reale, una serie di informazioni utili proprio per gestire l'emergenza stessa.

In una situazione di emergenza civile, come subito dopo un terremoto distruttivo, i tweet costituiscono una delle poche fonti immediate ed economiche di informazione dal campo, utilizzabile per ricostruire una situazione operativa¹.

Ci sono tante altre situazioni interessanti e non emergenziali, ad esempio: la condivisione di un evento come un concerto o una partita di calcio, con le foto. Oppure i visitatori di un ristorante che in uno stesso periodo di tempo pubblicano tweet sul ristorante stesso. In questo caso i tweet possono essere "ricomposti" per offrire un punto di vista speciale sull'evento stesso.

Un'altra situazione tipo è quella di un gruppo di persone che si sposta in una città, visitandola e condividendo commenti e foto via tweet: si può cercare di ricostruire i movimenti delle persone del gruppo a partire dai tweet che producono, piazzando inoltre le foto su una mappa.

2 Descrizione del problema

L'obiettivo del progetto è quello di creare un applicazione in grado di raccogliere i tweet e rilevare eventi macroscopici, locali o basati su particolari parole chiave esaminando i tweet stessi, di uno storico (cioè da un repository) o raccolti in stream in tempo reale. L'applicazione dovrà permettere di visualizzare, consultare i tweet con certi hashtag e - sotto certe condizioni- attivare una procedura di allarme. Ad esempio, il sistema dovrà essere in grado di:

- raccogliere i tweet (scoprirete che certe raccolte di tweet sono a pagamento: ovviamente vogliamo usare solo tweet gratis, quindi imparate a raccogliervi voi)
- classificare i tweet geolocalizzati
- classificare i tweet pubblicati in una data area geografica
- classificare i tweet contenenti un certo punto di interesse, ad esempio #SanSiro
- classificare i tweet con una certa parola chiave, ad esempio #viaggiodiclasse5B
- classificare i tweet geolocalizzati di una persona specifica, per seguirne gli spostamenti

I tweet così raccolti e classificati potranno essere analizzati in vari modi, principalmente in modalità grafica. In particolare si richiede di aggregare i tweet in una *dashboard* (visualizzatore) interattiva, possibilmente mostrando viste coordinate che presentino diversi dettagli dei dati (es. le posizioni su una mappa, una *term cloud* (nuvola di parole: vedere Fig.2 e 3) dei termini usati, un diagramma a barre con il numero di tweet nell'unità di tempo, ecc.). È un requisito fondamentale mostrare la posizione dei dati raccolti su una mappa (es. su Google Maps) ovviamente nel caso tale informazione sia presente nel tweet.

¹blog.twitter.com/en_sea/topics/insights/2018/5-Tips-for-using-Twitter-during-emergencies-and-natural-disaster.html



Figura 3: Nuvola aggregante dei tweet alle ore 5.03. Fonte: <http://www.datajournalism.it/terremoto-emilia-2012-i-tweet-che-contano/>

3 Un processo agile / Scrum-like

I gruppi del progetto dovrebbero avere sei membri. Tollerati gruppi con un membro in più o in meno.

Uno dei membri deve assumere il ruolo di Scrum Master che dovrà aiutare il team a rimuovere gli ostacoli che si presenteranno, incluso le relazioni con il gli stakeholder-Manegement e - in generale - gestire il processo; un altro ricorporà il ruolo di Product Owner, che fungerà da collegamento con il cliente (ruolo che ricopre il docente Missiroli). Gli altri membri sono developer (sviluppatori). Come detto,

Il processo si svolge su almeno tre sprint di due settimane ciascuno. Sono previste inoltre una fase preliminare e una fase conclusiva (*release sprint*), che conclude lo sviluppo.

Fase preliminare Addestramento a Scrum del gruppo mediante uso del gioco Scrumble (vedere link alla fine di questo documento), raccolta degli indicatori di valutazione del team da parte dello Scrum Master.
Progettazione preliminare: architettura, studio di fattibilità, ecc.

Fase di esecuzione Il team prosegue il progetto con un certo numero di sprint della durata di due settimane. Al termine di ciascuno sprint il team dovrà preparare una demo dello stato di avanzamento del prodotto (Sprint review - se possibile si terrà in diretta con il cliente), una riflessione sul processo (possibilmente usando le carte Essence) e una sessione di *backlog grooming* (cioè una revisione del backlog).

Fase conclusiva Preparazione del rapporto finale, inclusa autovalutazione (verranno date indicazioni sul rapporto finale e sull'autovalutazione entro la fine di aprile). Consegna del software e del rapporto finale entro maggio. Si suggerisce la preparazione di un breve video (max 3 min.) di presentazione del prodotto sviluppato, che mostri anche un'analisi "interessante" eseguita con il prodotto stesso. Voto di gruppo basato su demo, qualità del rapporto finale e analisi del processo; ad esso verrà applicato un differenziale basato sul contributo individuale e la relazione finale.

Allo scopo di mostrare la flessibilità dell'approccio agile, ciascun team decide e inizia lo sviluppo con un insieme di user story che potrà essere integrato in corso d'opera con altre, richieste dal docente/cliente.

Esempi (lista non esaustiva): aggiungere un elemento di filtro/selezione dei tweet, come la possibilità di selezionare una finestra temporale, in modo che l'utente del sistema possa selezionare i tweet solo di un determinato periodo; aggiungere la possibilità di cliccare su una delle parole della tag cloud in modo da filtrare i dati sulla base di tale parola chiave e di vedere quindi la mappa aggiornata di conseguenza; aggiungere altre sintesi grafiche di data analytics, quale un istogramma che mostri la frequenza con cui è comparsa una parola chiave nella finestra temporale prescelta.

3.1 Strumenti

L'uso dell'ambiente di sviluppo CAS [1] è obbligatorio; il nostro scopo è di raccogliere dati di progetto grazie alle funzionalità presenti nell'ambiente CAS, che è fatto da componenti open source. L'ambiente contiene i seguenti elementi:

1. Taiga (OBBLIGATORIO)
2. Gitlab (OBBLIGATORIO)
3. SonarQube (OBBLIGATORIO)
4. Mattermost (simile a Slack, ma FLOSS)
5. Bugzilla
6. Jenkins
7. Innometrics logger server, per raccogliere dati di sviluppo direttamente dai client

L'ambiente CAS è disponibile in più versioni:

- server di informatica: <http://pccabri2.mat.unimo.it/>
- si possono anche usare i singoli strumenti installandoli su un computer privato del gruppo: vedere <http://aminsep.disi.unibo.it/cas-project/> e poi <http://aminsep.disi.unibo.it/cas-project/source/1.0.1/>
- sul cloud, ma attenzione ai costi.

Vista la scarsa disponibilità del server di istituto, l'idea di avere versioni ridondanti non è da scartare.

Le carte Essence non fanno parte di CAS, si trovano al link di cui sotto.

4 Link

Twitter <https://developer.twitter.com/en/docs>

Tweet-Tracker <https://github.com/Zacomo/Tweet-Tracker>

Scrumble <http://scrumble.pyxis-tech.com>

Essence <http://semat.org/essence-user-guide>

Riferimenti bibliografici

- [1] P. Ciancarini, M. Missiroli, F. Poggi, and D. Russo. An open source environment for an agile development model. In *IFIP International Conference on Open Source Systems*, volume 582 of *Advances in Information and Communication Technology*, pages 148–162. Springer, 2020.