

Caccia al tweet: un approccio per la geolocalizzazione di un utente sulla base dei suoi tweet

Valerio Gregori

Email: val.gregori2@stud.uniroma3.it

Mattia Iodice

Email: mat.iodice1@stud.uniroma3.it

Alessandro Oddi

Email: ale.odd1@stud.uniroma3.it

*Nella relazione viene presentata l'implementazione di un framework per la geolocalizzazione di un utente a partire dal corpus dei suoi tweet. L'obiettivo prefissato era appunto l'implementazione del processo di geolocalizzazione, basato sull'articolo *You are where you tweet* di Cheng, Caverlee e Lee, prendendo spunto dal lavoro dello studente F. Tanzi. Il documento è strutturato in sezioni, ciascuna focalizzata su un aspetto diverso del lavoro svolto. In particolare, dopo una breve descrizione delle caratteristiche fondamentali degli strumenti utilizzati, vengono descritti gli approcci seguiti nella fase di implementazione, insieme alla giustificazione delle scelte architetturali e alle motivazioni che hanno spinto alla reimplementazione totale del tool. Nella sezione finale vengono quindi presentati i risultati ottenuti, fornendo un confronto diretto con le metriche riportate nell'articolo precedentemente citato.*

1 Dataset di riferimento

La prima parte del lavoro svolto ha riguardato lo studio dell'articolo *You are where you tweet* insieme all'analisi dei dati in esso utilizzati. Vengono quindi presentati i punti salienti emersi dall'indagine.

I dati presi in analisi nello studio condotto fanno riferimento a un dataset relativo a un insieme di tweet estrapolati e suddivisi in training set e test set. In particolare, le caratteristiche fondamentali sono le seguenti:

1. il processo di estrazione dei tweet è avvenuto tra il Settembre del 2009 al Gennaio del 2010
2. il training set contiene 115,886 utenti di Twitter e 3,844,612 aggiornamenti da parte degli utenti stessi.
3. ciascuna località degli utenti è automaticamente etichettata negli USA con livello di dettaglio relativo alla città.
4. il test set contiene 5,136 utenti di Twitter e 5,156,047 tweets
5. tutte le localizzazioni degli utenti sono ottenute dalla posizione dei loro telefoni e sono espresse nella forma *latitudine, longitudine*.

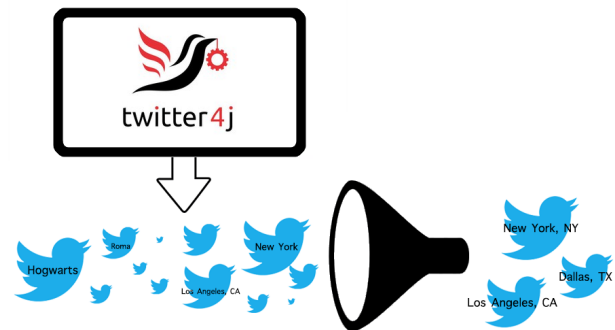


Fig. 1. Architettura del modulo per il retrieve di tweet dallo stream. Vengono scartate città non USA e vengono associate agli stati alle città non correttamente formattate.

Considerate le dimensioni piuttosto contenute dei dati in questione, in un primo momento abbiamo definito un modulo per arricchire (locupletare) il dataset. La scelta è stata guidata inoltre dall'esigenza di rispondere in modo efficiente ed efficace all'evoluzione della lingua. Infatti, le formule di espressione linguistiche, in contesti estremamente dinamici come quelli dei social network, sono soggette a continui cambiamenti: i.e. slang, neologismi, trend sociali ecc.

Il modulo implementato è essenzialmente un filtro che sfrutta lo stream offerto da **Twitter4j** effettuando una cernita tra tweet di interesse e non.

Per tweet di interesse si fa riferimento alla classe di aggiornamenti provenienti da utenti *ben geolocalizzati*: l'attributo *position* dell'utente deve comparire nella forma *città, stato* e la coppia in questione deve essere collocabile all'interno del territorio statunitense. Inoltre per far fronte all'eccessivo numero di utenti che dichiarano la città senza fornire lo Stato di appartenenza si è deciso di seguire un approccio gazetteer per l'individuazione dello Stato.

Considerando l'ingente tasso di omonimia fra i nomi delle città americane, data una città senza uno stato, si associa ad essa lo Stato relativo alla città più popolosa con quel nome. Mediante l'utilizzo di questo approccio vengono scartati fino

all' 80% di tweet che non sono geotaggati.

2 Problematiche iniziali

L'obiettivo definito all'inizio del progetto era improntato all' *intention mining* degli utenti, diversificandoli sulla base della loro localizzazione.

Per il task di localizzazione si voleva utilizzare il tool realizzato dallo studente Tanzi e quindi si é proceduto con un'attenta analisi del codice e delle rispettive funzionalità. Già dalla prima esecuzione non é stato possibile ottenere l'output sperato.

In particolar modo, é emerso che il dataset in input presentava tweet malformattati che non rispettavano la struttura necessaria per l'esecuzione del tool.

Escludendo i tweet in questione e testando il tool su un sottoinsieme del dataset iniziale é stato quindi possibile effettuare una stima del tempo d'esecuzione del processo di *parsing* sull'intero dataset. Il tempo richiesto stimato ammontava a circa una settimana a causa dell'utilizzo di strumenti non strettamente necessari, come spiegato nella Sezione successiva.

Risolti quindi i principali problemi legati ai tempi di computazione, si é proceduto con l'esecuzione del processo di geolocalizzazione. Anche qui, purtroppo, i costi d'esecuzione non sono stati soddisfacenti: l'implementazione aveva un costo pari a $O(n^3)$, con tempi di esecuzione stimati di circa 6 giorni.

Alla luce delle problematiche riscontrate si é quindi deciso di rinunciare al task legato all'*intention mining* degli utenti e piuttosto di focalizzarsi su un'implementazione funzionante e performante del processo di geolocalizzazione.

3 Descrizione dei flussi di esecuzione

Con l'intento di ridefinire l'intero processo di geolocalizzazione e la relativa valutazione, sono stati definiti tre macro-processi:

1. Parsing
2. Calcolo del Tf-Idf
3. Validazione

Nel paper *You Are Where You Tweet* il fulcro del processo di geolocalizzazione dell'utente é l'utilizzo di un indice **Tf-Idf** sulle parole utilizzate all'interno dei tweet analizzati. Un requisito fondamentale per la costruzione di un tf-idf di qualità risiede nella qualità del training set. A tale fine, é stato creato un package dedicato al raffinamento e alla pulizia dei dati.

3.1 Parsing

La prima fondamentale differenza con le implementazioni proposte dallo studente Tanzi risiede nell'utilizzo dello *Jazzy Spell Checker*. Essenzialmente si tratta di un modulo che corregge le parole in input non scritte correttamente associandole a termini simili.

Ad esempio il termine *boook* verrebbe associato al corrispettivo *book*.

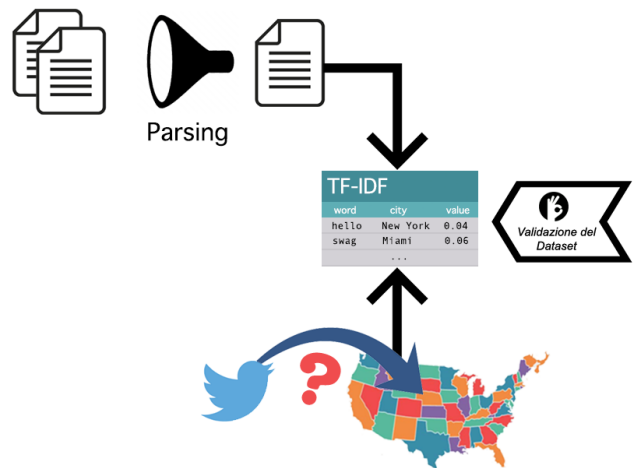


Fig. 2. Macro-processi nel flusso di geolocalizzazione.

Questo processo introduce un overhead considerevole, dovendo effettuare un confronto tra ciascuna delle 9M di parole nel dataset con 350K record nel dizionario. Oltre al notevole costo computazionale si tratta di un processo che esclude termini che potrebbero essere potenzialmente rilevanti nell'intero processo di geolocalizzazione. Questo é dovuto principalmente alla presenza di termini dello slang che non fanno parte del dizionario: si é quindi deciso di non utilizzare tale modulo.

Il processo di parsing si articola in 3 fasi differenti:

1. Pulizia del dataset
2. Analisi e raffinamento dei termini
3. Composizione dei record

La fase di pulizia del dataset si occupa di scartare tutti quei tweet che non sono nel formato richiesto, ovvero *id utente*, *id tweet*, *testo*.

Il numero di tweet che non risultavano formattati correttamente e quindi scartati dal primo filtro é di circa **100K**.

La fase successiva, quella dell'analisi e del raffinamento dei termini, ha invece lo scopo di eliminare il contenuto che non é di interesse nel processo di geolocalizzazione.

In particolare, vengono scartati:

1. Indirizzi email
2. Siti
3. Hashtag
4. Tag
5. Parole contenenti caratteri non ASCII
6. Caratteri singoli
7. Tweet duplicati

Alla fine di questa fase vengono scartati ben 170K tweet. I rimanenti subiscono un'ulteriore trasformazione consistente in un join con le coppie *utente-città*.

Come accennato in precedenza, nel dataset non tutti gli utenti sono correttamente geotaggati.

Il problema più frequente consiste nel fatto che alcune città sono sprovviste della sigla dello Stato: per ovviare a questo problema si é adottata la soluzione di assegnare alle città lo

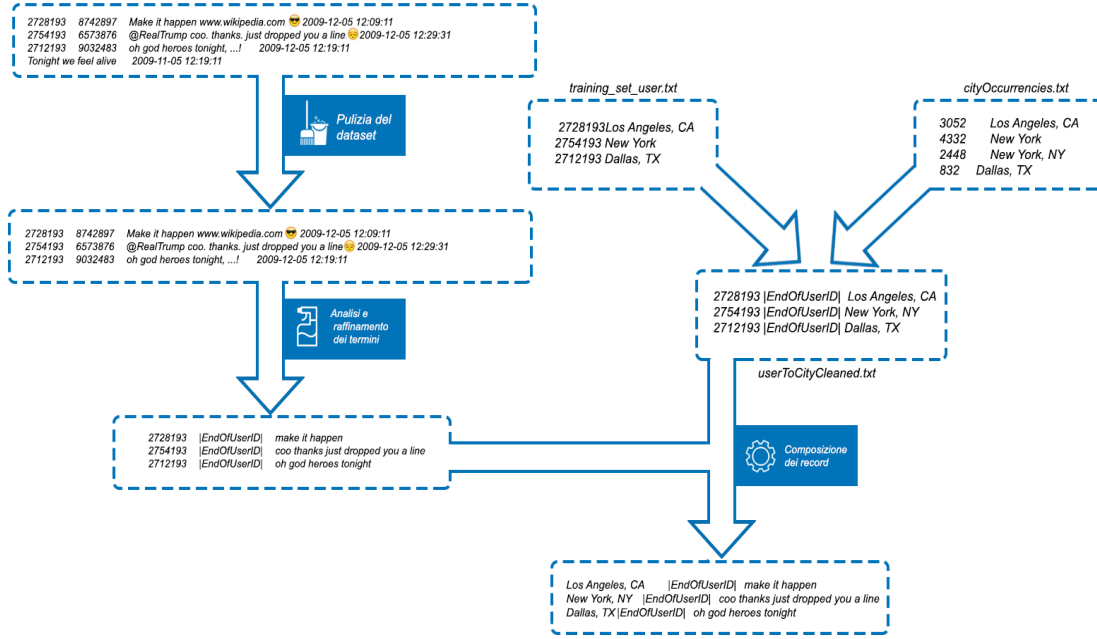


Fig. 3. Nella figura vengono mostrate le dipendenze individuate tramite la libreria di Stanford. Gli nmod rilevanti, cerchiati in verde, sono quelli che presentano diversi dependent. nmod:of viene quindi scartato in quanto presenta una sola dependent.

stato corrispondente alla città omonima e più popolosa. Anche in caso di città malformattate, come ad esempio *New York, New York*, il secondo elemento viene rimosso in favore del relativo Stato.

3.2 TfIdf

Il processo di geolocalizzazione avviene sulla base del calcolo del *Tf-Idf* delle parole all'interno dei tweet. Intuitivamente parole che sono utilizzate molto in una città e poco nelle altre sono preziosi indicatori. La formula utilizzata differisce leggermente da quella presente nel paper, ed è la seguente:

$$tf_{ij} = \frac{f_{ij}}{\max_k \{f_{kj}\}}$$

$$idf_i = \log_a \left(\frac{N}{d_{f_i}} \right)$$

In un primo momento nel calcolo del *tf-idf* sono state prese in considerazione tutti i termini presenti nei tweet, implicando un tempo di esecuzione di circa 4h. Dopo un'attenta analisi è emerso che molti dei termini presenti, circa il 90%, si presentavano meno di 50 volte all'interno dell'intero dataset. Escludendo le parole in questione e senza avere impatti negativi sull'accuratezza è stato possibile abbassare il tempo di esecuzione a 15min.

Table 1. Esempio dei record utilizzati nel processo di validazione

Word	City Coordinates	Tf-Idf
love	44.9800000, -93.2636111	0,03
peace	47.8211111, -122.3138889	0,01
morales	25.7738889, -80.1938889	0,027

3.3 Validazione

Il modulo di geolocalizzazione è stato validato sul dataset di test utilizzato in *You Are Where You Tweet*. I dati in input sono stati processati come illustrato in Sez. 3.1.

Un aspetto rilevante di questa fase è il calcolo delle distanze tra coordinate relative a città diverse. Nell'implementazione dello studente Tanzi era previsto un uso massiccio dell'API *Google Geocoder* per determinare le coordinate corrispondenti a una città e poi calcolare le distanze. L'utilizzo dell'API è però limitato ad un massimo di 2500 richieste giornaliere e per questo motivo un approccio basato su continue interrogazioni è destinato a interrompersi. L'implementazione corrente sfrutta invece una mappa esistente di tipo *città, coordinate*, inoltrando richieste all'API soltanto nel caso in cui il record d'interesse non è presente. Tale implementazione sostiene anche l'utilizzo dello streaming per il retrieve di nuovi dati.

Per il calcolo della distanza tra due coordinate abbiamo fatto uso della seguente formula:

$$dist = \arccos(\sin(lat_1) \cdot \sin(lat_2) + \cos(lat_1) \cdot \cos(lat_2) \cdot \cos(long_2 - long_1))$$

4 Risultati