UNIVERSITÀ DI PISA

Corso di Laurea in Informatica Umanistica

RELAZIONE

Analisi dei post sui social network per l'individuazione di social event

Candidato: *Alessio Vaccai*

Relatore: *Maurizio Tesconi*

Correlatore: *Mirko Tavosanis*

Correlatore: *Fabio Del Vigna*

Anno Accademico 2014-2015

Indice

1. Introduzione

Un concerto, una fiera, l'apertura di un nuovo negozio, una manifestazione, una degustazione di vini... Questi sono tipici esempi di eventi che, attraendo un sufficiente numero di persone, creano quello che viene detto *social event*. Sono eventi, quindi, che portano le persone ad aggregarsi, stare insieme e condividere esperienze.

Con l'avvento degli smartphone e dei social network online, servizi che permettono agli utenti di condividere contenuti testuali, immagini e video con altre persone e di creare reti sociali, è nata la tendenza a pubblicare su internet foto e commenti di ciò che attrae la nostra attenzione o che riteniamo degno di nota. La quantità di questi dati è incrementata molto velocemente negli ultimi anni e continua tuttora a crescere, creando una collezione di dati eterogenea e in continua evoluzione. Spesso ciò che spinge gli utenti a pubblicare contenuti online (detti *post* in gergo) è la volontà di condividere informazioni ritenute interessanti con altre persone.

Anche i social event sono oggetto di condivisioni e pubblicazioni, che spesso avvengono proprio mentre questi eventi si stanno svolgendo. Questa tendenza può portare ad ipotizzare che, in presenza di social event, i post inviati dagli utenti si concentreranno nelle zone in cui tali eventi si manifestano. Sulla base di tale supposizione, se fossimo in grado di conoscere i luoghi in cui i post si raggruppano, riusciremmo ad ottenere una mappa dei luoghi d'interesse utile sia all'individuazione istantanea di social event, sia all'indagine delle potenzialità attrattive di una località.

Il progetto oggetto di questa relazione si propone di verificare questa ipotesi creando un'applicazione web in grado di estrapolare una base di dati interpretabili dai post di Instagram, utili allo studio del comportamento degli utenti e all'indagine delle tendenze e delle fonti di interesse delle zone analizzate.

2. Stato dell'arte

Ad oggi, sono state sviluppate varie applicazioni mirate all'individuazione in tempo reale di eventi. Allo stesso modo, il flusso di informazioni inviate dagli utenti sui servizi social è stato oggetto di studi da parte della comunità scientifica.

Uno di questi in particolare, dal titolo *Modeling People and Places with Internet Photo Collections* (Carnall, Snavely, 2012), riscontrava che "i dati ottenuti da Flickr[1] possono essere utilizzati per studiare il comportamento dei fotografi, poiché ogni foto è una osservazione di ciò che un particolare utente sta facendo in un particolare istante. Per esempio, studiare le sequenze di foto geolocalizzate e fornite di timestamp può servire a tracciare i percorsi seguiti dalle persone. [...] Questo dimostra come i dati ottenuti da sistemi online di social-sharing possano essere utilizzati per indagare su questioni sociologiche ad un livello finora irraggiungibile. "

Figura 1. Percorsi individuati mediante l'analisi delle sequenze dei post degli utenti nell'isola di Manhattan. Si può notare come la struttura a griglia delle strade sia chiaramente visibile, così come i percorsi turistici più popolari come la camminata lungo il ponte di Brooklyn e i traghetti che partono dalla parte inferiore dell'isola.

Anche il problema dell'identificazione dei social event attraverso l'interpretazione dei social media è stato affrontato in precedenza. Una possibile soluzione è stata discussa da Manchon-Vizuete et al, dove gli autori hanno sfruttato tutti i metadati forniti insieme alle foto per individuare i social event da loro rappresentati, dimostrando che, combinando adeguatamente talie meta-informazioni, è possibile ottenere risultati molto accurati rispetto all'utilizzo del solo timestamp.

[1] Flickr è un sito web e applicazione per smartphone per la condivisione online di fotografie.

Anche Tamura et al hanno proposto una soluzione creando un algoritmo per individuare post inerenti allo stesso argomento. Tale algoritmo si basa sulla correlazione dei post in base ai termini presenti nelle loro didascalie e sull'individuazione di *burst*[2].

L'utilizzo dei post come indice di gradimento verso un certo prodotto è stato studiato con esiti positivi da Asur e Huberman nel loro studio intitolato *Predicting the future with social media* (2010). In questo esperimento venivano analizzati i post inerenti all'uscita di una serie di film tra novembre 2009 e febbraio 2010, indagando sulla frequenza con cui questi venivano pubblicati e sulle opinioni espresse dagli autori.

I risultati hanno dimostrato che non solo i post riflettono il successo di un certo prodotto, ma possono anche essere utilizzati per ottenere delle previsioni realistiche sul suo gradimento.

Dal punto di vista applicativo invece, due esempi notevoli sono state le app Now e GonnaBe.

La prima si basava sui post ricavati da vari social network (Instagram, Facebook, Vine e Google+). Sfruttando algoritmi per l'elaborazione di immagini, insieme all'interpretazione dei contenuti ed alla loro composizione da parte di addetti, Now forniva agli utenti uno sguardo accurato su cosa stava succedendo vicino a loro.

[2] Con burst si intende il rapido aumento delle pubblicazioni di post in un ristretto intervallo temporale.

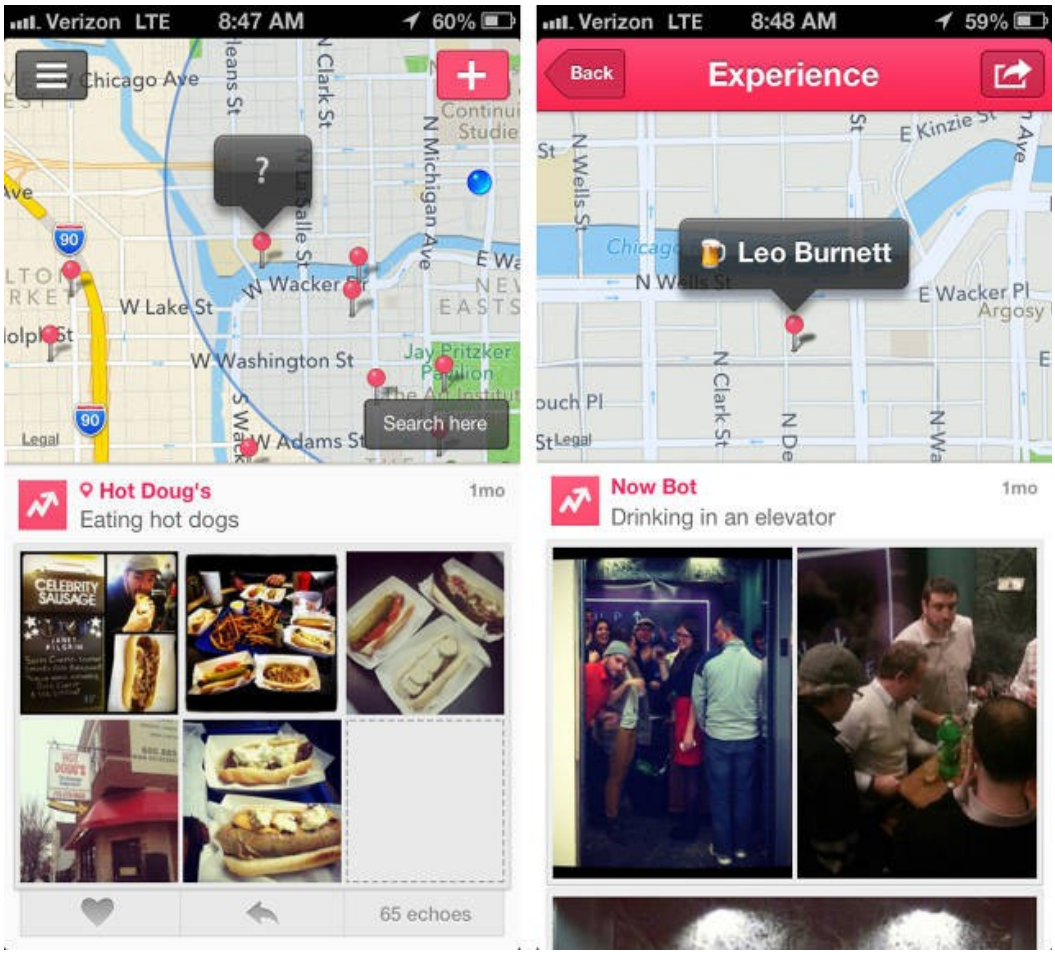

Figura 2. Schermate dell'applicazione Now

GonnaBe, invece, si proponeva come vera e propria "sfera di cristallo" in grado di prevedere gli eventi più interessanti e di tendenza in base ai dati forniti dagli utenti dei social network.

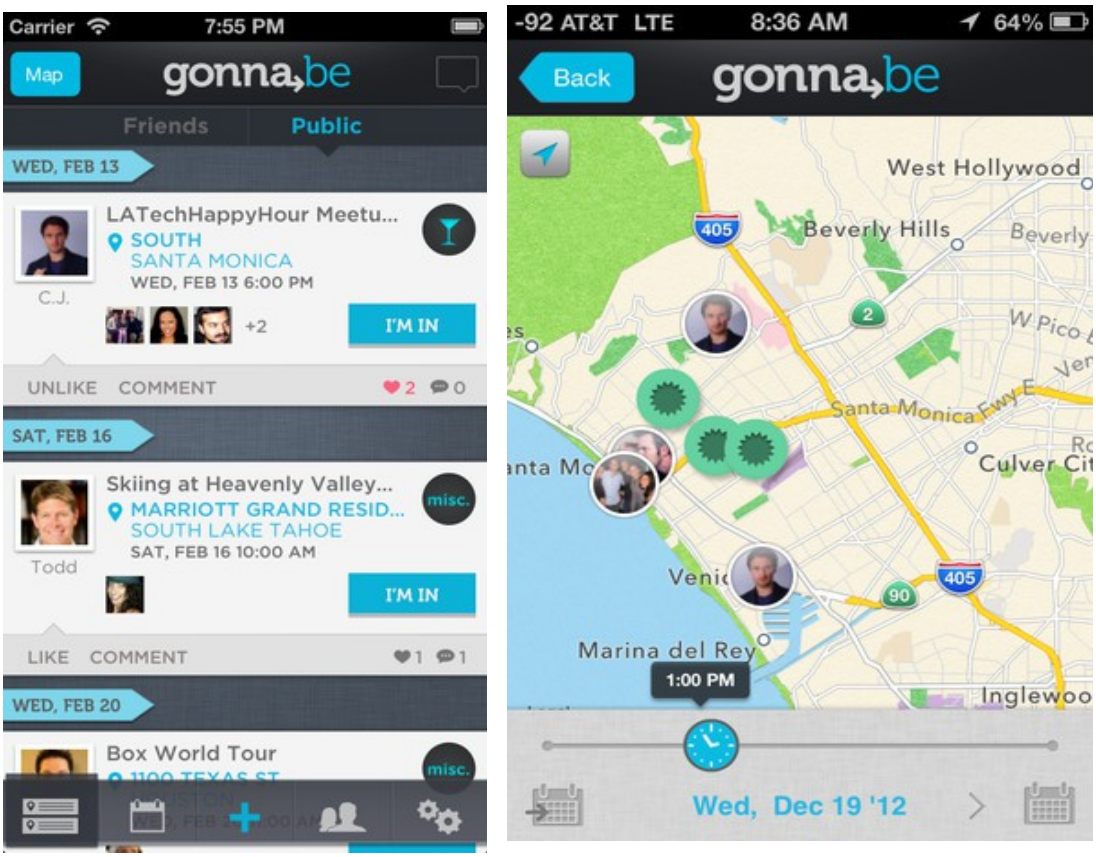

Figura 3. Schermata dell'applicazione GonnaBe

Era possibile infatti pubblicare direttamente sull'applicazione i nostri programmi futuri e leggere quelli inviati dagli altri, creando così una specie di "meteo degli eventi".

Entrambe queste applicazioni sono state dismesse.

3. Strumenti e servizi utilizzati

L'applicazione oggetto di questa relazione è stata realizzata utilizzando i principali linguaggi di sviluppo web in congiunzione con tre servizi online allo scopo di recuperare ed elaborare i post di Instagram.

L'intero codice è stato scritto utilizzando un semplice editor di testo.

3.1 Interfaccia e grafica

Per usufruire dell'applicazione è stata creata una pagina web utilizzando il linguaggio HTML[3] per definirne la struttura. A questa pagina stato applicato un foglio di stile CSS[4] per impostare i dettagli grafici, rendendo l'interfaccia meno scarna e più accattivante e leggibile. Per una migliore interazione con gli elementi della pagina, oltre ad una più rapida scrittura del codice stesso, è stata utilizzata la libreria jQuery[5] ed uno script in JavaScipt, detto DateTimePicker, per selezionare in maniera più pratica le coordinate temporali.

3.2 Algoritmi e chiamate ai servizi

Gli algoritmi di recupero, elaborazione e visualizzazione dei dati sono stati realizzati in JavaScript, un linguaggio di programmazione comunemente utilizzato per la realizzazione di applicazioni web lato client (ovvero, l'applicazione viene eseguita sul dispositivo utilizzato dall'utente per navigare la pagina).

Le richieste da inviare ai vari servizi vengono inoltrate attraverso chiamate AJAX[6], utilizzando la funzione della libreria jQuery `$.getJSON()`.

Tale funzione prende in input l'indirizzo web del servizio da interrogare ed una funzione, detta di *callback*, che verrà invocata all'arrivo della risposta da parte del server.

[3] L'HyperText Markup Language (HTML) è un linguaggio utilizzato per formattare e impaginare documenti ipertestuali sotto forma di pagine web.

[4] Il CSS (Cascading Style Sheet) è un linguaggio utilizzato per la formattazione delle pagine web.

[5] jQuery è una libreria di funzioni Javascript per le applicazioni web, che si propone come obiettivo quello di semplificare la manipolazione, la gestione degli eventi e l'animazione delle pagine HTML.

[6] L'Asynchronous Javascript And XML è una tecnica di sviluppo di applicazioni web interattive che si basa sullo scambio in background di informazioni tra il browser e il server, consentendo l'aggiornamento dinamico della pagina web senza l'esplicito ricaricamento da parte dell'utente.

3.3 Archiviazione delle informazioni

Per archiviare le informazioni dei post e poterle riutilizzare in futuro è stato utilizzato un database SQL[7]. Per interagire con esso è stato creato un file PHP[8] in cui è stato scritto il codice per poter salvare i dati dei post nelle tabelle del database o per recuperare tali informazioni e renderle disponibili all'elaborazione.

3.4 Instagram

Instagram è un'applicazione per smartphone gratuita e un social network fotografico. Permette di scattare fotografie e di condividerle istantaneamente anche su altri social media. Insieme alle foto l'utente può inserire una didascalia e delle tag, ovvero delle parole chiave che descrivono il contenuto della foto. Gli altri utenti dell'applicazione possono commentare le foto pubblicate, manifestare il loro apprezzamento attraverso i *like*[9], e seguire (*follow*) gli aggiornamenti dell'autore.
Instagram permette anche di manipolare le foto applicando dei filtri, offrendo una vasta gamma di effetti pronti all'uso attraverso una interfaccia semplice ed accessibile anche agli utenti meno esperti.
Inizialmente solo per telefoni con sistema operativo iOS, si è estesa anche su cellulari Android e più recentemente anche sul web con la versione desktop.
Nata nell'ottobre del 2010 e acquisita da Facebook nel 2012, ha riscosso un successo sempre maggiore, diventando una delle applicazioni più usate e tutt'oggi in voga.

[7] Lo Structured Query Language (SQL) è un linguaggio utilizzato per la creazione, la gestione e l'amministrazione del database.
[8] Il PHP (PHP: Hypertext Preprocessor) è un linguaggio di programmazione utilizzato principalmente per lo sviluppo di applicazioni lato server.
[9] In molti social network è presente il pulsante *like* che gli utenti possono utilizzare per esprimere il proprio apprezzamento verso un certo contenuto. Tipicamente, insieme al pulsante, è presente anche un numero che indica la quantità di *like* accumulati dal post.

3.5 Google Maps

Google Maps è un servizio offerto da Google che permette a titolo gratuito la fruizione di mappe e foto satellitari della quasi totalità del pianeta. Oltre a questo, Google Maps mette a disposizione dell'utente funzioni di geolocalizzazione e indicazioni stradali.
L'applicazione per la visualizzazione delle mappe Google Maps si trova preinstallata su tutti i dispositivi Android ed è disponibile anche per altri sistemi operativi sia per smartphone che per pc. Il suo ampio utilizzo la sta rendendo una valida alternativa ai navigatori satellitari[10].
Google ha reso disponibile questo servizio sia come applicazione mobile e desktop, sia come libreria JavaScript da poter inserire nel proprio sito web.

3.6 Language Detection API

La Language Detection API è un servizio web per l'individuazione della lingua di un testo[11]. In particolare, inviando un testo all'indirizzo della API mediante una chiamata POST[12], otteniamo una stringa in formato JSON[13] che contiene il codice ISO 639-1 (uno standard per la reappresentazione dei nomi delle lingue in due caratteri) della lingua identificata e il grado di probabilità dell'identificazione. Il risultato non sempre è affidabile, ma in generale, un testo più lungo sarà identificato con maggior accuratezza.
La Language Detection API può identificare fino a centosessanta linguaggi.

[10] Aranzulla.it, *Navigatore Android* - http://www.aranzulla.it/navigatore-android-41656.html

[11] https://detectlanguage.com/

[12] Il metodo POST è uno dei metodi con cui è possibile inviare una richiesta ad un server. Questo metodo viene utilizzato per inviare i dati al server in modo che vengano processati.

[13] Il JavaScript Object Notation (JSON) è un formato per lo scambio di dati tra applicazioni client-server.

3.7 Application Programming Interface (API)

Molti servizi web, tra cui quelli descritti sopra, mettono a disposizione degli sviluppatori delle interfacce per poter sfruttare i loro dati in altre applicazioni.
Queste interfacce vengono dette *Application Programming Interface* o *API*.
In questo progetto sono state utilizzate principalmente tre API: quelle di Google Maps per visualizzare una mappa del mondo navigabile ed interattiva; quelle di Instagram, per ottenere le foto pubblicate filtrate per timestamp e luogo, insieme alle loro tag ed ai commenti; la *Language Detection API*, un servizio che permette di conoscere la lingua con cui è scritto un testo.
Spesso chi mette a disposizione delle API ne limita il loro utilizzo gratuito e richiede dei prezzi in base al numero di richieste che si prevede di dover inviare. È appunto il caso di Instagram e di Language Detection API.

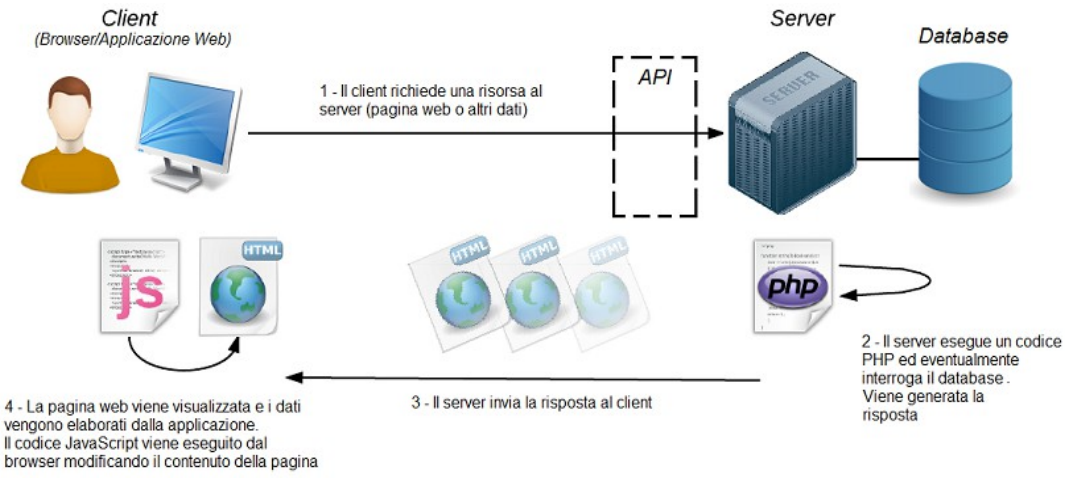

Figura 4. Schema dell'interazione tra client e server durante richiesta di una risorsa.

4. Raccolta dati

Tra i principali social network presenti in rete, Instagram è uno dei più
adatti ad essere utilizzato come fonte di dati, essendo concepito per permettere agli utenti di scattare foto e di pubblicarle direttamente online. L'utilizzo di tag e di

immagini, insieme alla geolocalizzazione dei post e alla loro collocazione temporale, permette la raccolta di informazioni utili allo scopo.

Per ottenere un database di post da utilizzare per un eventuale analisi futura è stato messo a punto un *crawler*, ovvero un software in grado di recuperare in automatico grandi quantità di informazioni da una rete o da un database (in questo caso da Instagram). Questa applicazione è stata creata separatamente da quella destinata all'utente finale ed ha l'unico scopo di reperire e salvare sul database grandi quantità di informazioni. Questo tipo di sofware viene spesso utilizzato per creare indici o collezioni di pagine web da utilizzare in locale, partendo da un insieme di URL[14] detto *seed URL* ed esplorandoli sistematicamente, aggiungendo ogni altro indirizzo incontrato nel processo alla lista di pagine da visitare. In questo progetto il crawler è stato utilizzato per ottenere i dati dei post post pubblicati in dieci città italiane (Bologna, Cagliari, Firenze, Milano, Napoli, Palermo, Rimini, Roma, Torino e Venezia) tra febbraio e marzo 2015. La scelta delle città è stata fatta in modo da coprire tutto lo Stivale ed in modo da poter valutare zone ricche di mete turistiche ed aree in cui è presente una maggior quantità di attività social.

Le informazioni raccolte dal crawler sono state poi salvate in un database in modo da renderle fruibili ed analizzabili.

Per ottenere questa collezione di post, è stata scritta un'applicazione che, una volta impostati i parametri latitudine e longitudine, porta automaticamente la data un mese indietro. Successivamente l'area da ricercare viene divisa in una griglia, in modo da permettere una migliore fruizione dei dati (cfr. Ottenere i dati da Instagram).

Ogni parte della griglia viene divisa temporalmente in tre fasce orarie: mattina (dalle 8.00 alle 13.59), pomeriggio (dalle 14.00 alle 19.59) e sera (dalle 20.00 alle 23.59) .

Questo processo viene ripetuto spostando progressivamente la finestra temporale fino ad arrivare alla data attuale.

Ogni post ottenuto viene quindi inviato al server, che lo inserisce nel database.

[14] L' Uniform Resource Locator (URL) è una stringa di caratteri che identifica univocamente una risorsa su internet.

Di ogni post vengono salvati:

- **id** - numero identificativo del post
- **id utente** - id dell'autore del post
- **latitudine** e **longitudine** - coordinate del luogo in cui è stato pubblicato il post
- **data** - data di pubblicazione
- **tag** - elenco delle tag del post
- **immagine** - URL dell'immagine pubblicata
- **fascia oraria** - mattina (dalle 08.00 alle 13.59), pomeriggio (dalle 14.00 alle 19.59) o sera (dalle 20.00 alle 23.59)
- **timestamp**[15]
- **giorno della settimana**
- **città**
- **nome luogo** - campo inserito dall'utente al momento della pubblicazione

Questi dati possono essere utilizzati per ottenere uno storico degli eventi o per effettuare analisi statistiche sulla tendenza all'uso di Instagram in una certa zona.

4.1 Realizzazione del crawler

Il crawler descritto nel paragrafo precedente è stato realizzato utilizzando un database SQL per l'archiviazione dei dati. Questi tipi di database vengono comunemente utilizzati per la loro efficenza nel gestire e manipolare grandi quantità di dati e sono molto diffusi tra le applicazioni web.
Per l'interrogazione della search API è stato scritto un codice JavaScript e una semplice interfaccia è stata implementata per rendere più pratico l'utilizzo del crawler.
Le richieste venivano inviate al server di Instagram attraverso chiamate AJAX e una volta ricevuti i dati di risposta, questi venivano rispediti ad una pagina PHP che salvava le informazioni sul server.

[15] Il timestamp, o marca temporale, è una sequenza di caratteri che rappresenta una data e/o un orario. In questo progetto, il timestamp equivale alla data di creazione del post espressa in secondi.

4.2 Analisi dei dati

Gli eventi solitamente comportano l'aggregazione di un certo numero di persone all'interno di aree geografiche limitate per tutta la durata di tale occorrenza. Queste aggregazioni sono identificabili se si considera il numero di post che vengono pubblicati in una certa area e in un limitato intervallo di tempo. Se quei post si concentrano in determinati luoghi, è probabile che gli autori di quei contenuti stiano partecipando ad un social event. Questo approccio però non tiene in considerazione la densità di popolazione media delle aree interessate. Aree ad alta intensità abitativa o di forte interesse turistico/economico presentano naturalmente una forte tendenza a formare concentrazioni di post, indipendentemente se vi siano o meno eventi nelle vicinanze. Secondo quanto riportato da Instagram[16], infatti, le dieci città con maggior numero di foto e condivisioni del 2013 sono state New York, Bangkok, Los Angeles, Londra, São Paulo, Mosca, Rio de Janeiro, San Diego, Las Vegas e San Francisco, tutte aree demograficamente prospere e ricche di attrazioni turistiche.
Dopo la fase di raccolta dei post descritta nel paragrafo precedente, è necessario ora indagare sui risultati che otteniamo dalla search API in tempo reale. Un confronto tra questi due dati, secondo le ipotesi scritte nell'introduzione, dovrebbe permettere la comprensione dei comportamenti degli utenti e delle dinamiche di aggregazione sociale all'interno delle aree in esame.

4.3 Ottenere i post da Instagram

Come già spiegato, i post di Instagram vengono recuperati attraverso un'apposita API detta *search API*. La search API offre un'interfaccia che permettere di sottomettere query[17] al sistema basate su posizione geografica e intervallo temporale di pubblicazione.
La API è disponibile al seguente indirizzo:

```
https://api.instagram.com/v1/media/search
```

[16] http://blog.instagram.com/post/69877035043/top-locations-2013
[17] Con query si intende l'interrogazione del database da parte di un utente.

La API permette di costruire query per il filtraggio dei dati, utilizzando i seguenti parametri:

- **Lat** - Latitudine del centro dell'area da ricercare
- **Lng** - Longitudine del centro dell'area da ricercare
- **Distance** - Lunghezza del lato dell'area da ricercare (in metri)
- **Min_timestamp** - Timestamp minimo da ricercare
- **Max_timestamp** - Timestamp massimo da ricercare
- **Count -** Numero di post da ottenere (massimo 100)

Ogni ricerca viene effettuata su porzioni di mappa di forma quadrata. Il lato di tale area è regolato dal parametro **distance**.
Per rendere più comodo l'utilizzo di questa API, è stata messa a punto questa interfaccia web:

Figura 5. Pannello di ricerca dell'applicazione

Il pulsante *Cerca Qui* imposterà automaticamente i parametri Lat e Lng sulle coordinate del centro della mappa (cfr. paragrafo successivo).

La ricerca restituisce una stringa in formato JSON in cui i post sono così rappresentati:

```
data:{
      attribution: null
      caption: Object
      comments: Object
      created_time: "1429827668"
      filter: "Normal"
      id: "969787226789948986_209015426"
      images: Object
      likes: Object
      link: "https://instagram.com/p/11YFg4hq46/"
      location: Object
      tags: Array[9]
      type: "image"
      user: Object
      users_in_photo: Array[0]
}
```

Tra questi dati, quelli che verranno utilizzati nell'applicazione sono:

- **caption** - didascalia della foto
- **comments** - commenti al post
- **created_time** - timestamp del momento in cui è stato creato il post
- **id** - codice identificativo
- **images** - link all'immagine
- **link** - link al post
- **location** - contiene coordinate ed altre informazioni sul luogo di pubblicazione
- **tags** - elenco delle tag

Come si è detto nei paragrafi precedenti, le API di Instagram impongono dei limiti al numero di richieste effettuabili gratuitamente. Tale limite corrisponde a cinquemila

richieste all'ora e restringe ad un massimo di cento il numero di post ottenibili per ognuna[18].

Per ottimizzare il numero di invocazioni della API concesso, l'algoritmo di ricerca è stato impostato in modo da dividere l'area da ricercare in parti più piccole e di inviare una richiesta per ogni parte. In questo modo si possono ottenere fino a cento post per ogni parte, incrementando nettamente il numero dei risultati, a scapito del numero di richieste effettuabili.

Questo limite tuttavia, sebbene sia più che accettabile su finestre temporali ristrette, diventa invalidante su ricerche per periodi più ampi. Infatti, sia che si ricerchino i post delle ultime quattro ore o che si ricerchino quelli dell'ultima settimana, otterremo sempre un massimo di cento post per richiesta.

Per ovviare a questo problema, si è scelto di aumentare il numero di richieste in base alla larghezza della finestra temporale, dividendola in parti più piccole qualora fosse necessario.

Instagram restituisce i post in ordine di tempo decrescente, dal più recente al meno recente. Può capitare quindi che, una volta raggiunta la soglia dei cento post, l'ultimo di questi sia ancora lontano dal limite inferiore della finestra temporale.

Per risolvere questo problema, l'algoritmo è stato impostato in modo da effettuare una nuova invocazione della API se il numero di post ricevuto equivale alla soglia massima (ovvero cento) e se la differenza tra il timestamp dell'ultimo post ricevuto e il parametro *min_timestamp* è maggiore di due minuti.

Il parametro *max_timestamp* della nuova chiamata verrà impostato con il valore del timestamp dell'ultimo post ricevuto. Questa pratica viene detta paginazione.

[18] https://instagram.com/developer/limits/

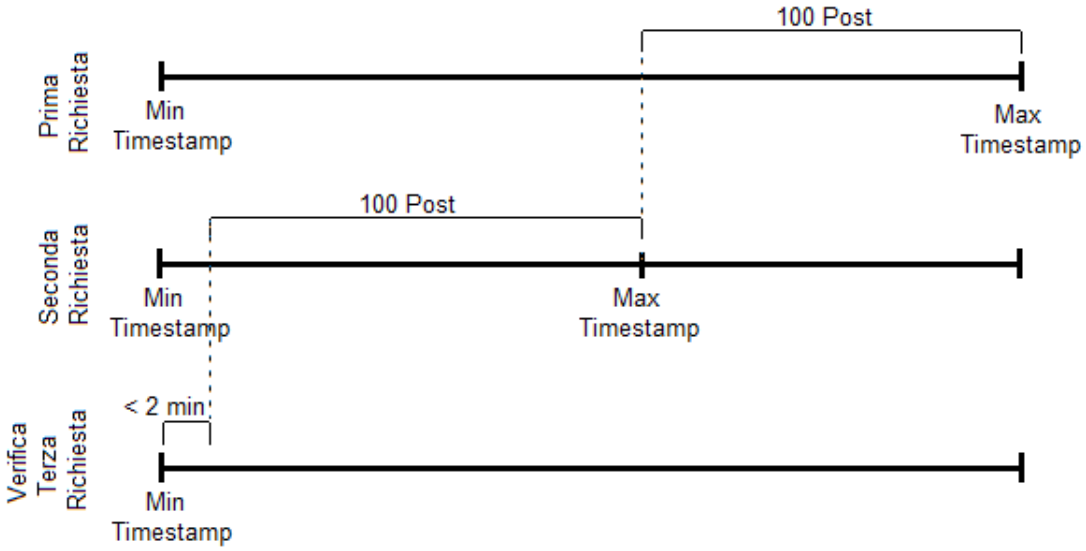

Figura 6. Schema di funzionamento della paginazione delle richieste.

Questo procedimento viene effettuato per ogni cella della griglia in modo che, per ognuna di esse, il numero di chiamate sia proporzionato alla densità di post al suo interno.

Utilizzando questo metodo è possibile ottenere tutti i post presenti nell'area soggetta alla ricerca che sono stati scattati nell'intervallo temporale impostato dall'utente in maniera efficiente e completa.

Riuscire a recuperare dati esaurienti è fondamentale per assicurare la migliore accuratezza possibile nell'individuazione dei punti di interesse.

4.4 L'individuazione della lingua mediante n-grammi

Il metodo utilizzato dalla Language Detection API per individuare la lingua di un testo è detto *categorizzazione mediante n-grammi*. Un n-gramma è un insieme di *n* caratteri adiacenti tra loro e appartenenti alla stessa parola.

Per la parola *ciao* avremo quindi (aggiungendo degli spazi per compensare la mancanza di caratteri)

- Bigrammi: _c, ci, ia, ao, o_
- Trigrammi: _ci, cia, iao, ao_, o__
- Tetragrammi: _cia, ciao, iao_, ao__, o___

e così via.

Ogni linguaggio umano possiede un insieme di parole che si ripetono più frequentemente di altre. Questo concetto viene comunemente espresso da quella che viene detta *Legge di Zipf*:

> La *n*-esima parola più comune di un testo scritto in un linguaggio umano occorre con una frequenza inversamente proporzionale a *n*.

Questa legge, in altri termini, stabilisce che la frequenza di una parola è inversamente proporzionale al *rango*. Ordinando le parole di un testo in base alla frequenza decrescente, il rango rappresenta la posizione occupata dalle parole in questo ordinamento. Quindi la parola più frequente avrà rango 1, la seconda avrà rango 2 e così via fino all'ultima, che avrà la frequenza più bassa.

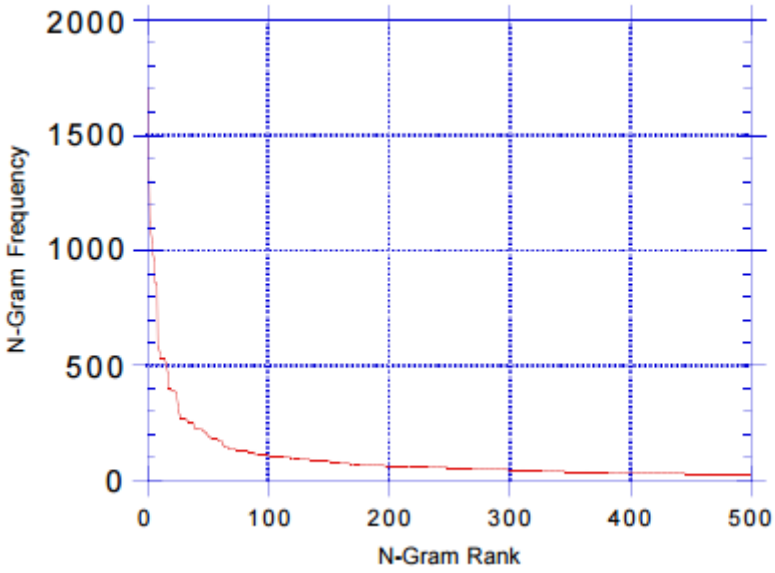

Figura 8. Grafico delle frequenze degli n-grammi in base al loro rango

Questo vale sia per le parole, che per gli n-grammi, come si nota dal grafico in figura 8, dove è rappresentata della distribuzione degli n-grammi in un testo tecnico in inglese.

Sull'asse delle ordinate sono riportate le frequenze degli n-grammi, mentre sulle ascisse si trova il rango degli n-grammi .

Ciò implica che testi diversi appartenenti allo stesso linguaggio avranno tendenzialmente la stessa distribuzione della frequenza degli n-grammi.
Questa osservazione permette quindi di stabilire una relazione tra gli n-grammi di un testo e il linguaggio con cui è scritto.
Il server della API ha in memoria le frequenze medie degli n-grammi di ogni lingua che riesce ad identificare, ottenute attraverso l'analisi di grandi collezioni di testi dette *corpora*.
L'applicazione estrapola quindi dal testo che riceve in input gli n-grammi e ne calcola la frequenza. Questi dati vengono poi confrontati con le frequenze di ogni lingua. Il linguaggio del testo sarà probabilmente quello che si avvicna di più alle frequenze inviate.
Il sistema ovviamente non è infallibile, ma più il testo da identificare è lungo, più il risultato sarà accurato.

4.5 Individuare la lingua di un post

Come già scritto sopra, la *Language Detection API* ci fornisce informazioni sulla lingua di un post. Una volta ottenuta la didascalia e i commenti possiamo quindi tentare di indovinare la lingua di chi ha pubblicato la foto.
Le tag in questo procedimento verranno utilizzate solo nel caso in cui didascalia e commenti non siano disponibili. È infatti facile[19] imbattersi in post che si presentano più o meno in questa forma:

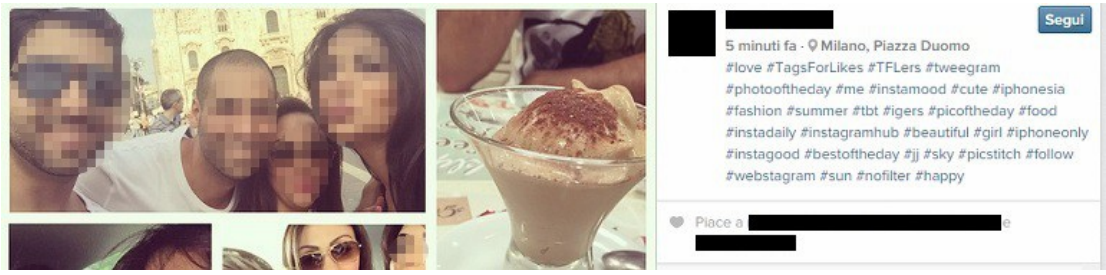

Figura 9. Uno dei frequenti post con tag non informative

[19] Non a caso le tag nella didascalia del post in figura compaiono quasi interamente nella lista delle cinquanta tag più utilizzate su Instagram (http://top-hashtags.com/instagram/).

Questa lista di tag viene spesso utilizzata per rendere il post più visibile e oltre ad essere priva di contenuto informativo, può portare ad una valutazione errata da parte della API, che potrebbe identificare erroneamente il post.

Per utilizzare la *Language Detection API* è sufficiente inviare una richiesta al seguente indirizzo:

```
http://ws.detectlanguage.com/0.2/detect
```

L'unico parametro da inviare è il testo da tradurre. Il risultato sarà uno script JSON in questa forma:

```
"data":{
  "detections":[
    {
      "isReliable":false,
      "confidence":0.45171339563862928,
      "language":"es"
    }
  ]
}
```

Dove il campo *language* è un'abbreviazione del nome della lingua identificata, *isReliable* indica se l'identificazione è affidabile sulla base del parametro *confidence*, che determina quanto l'identificazione è accurata in funzione della lunghezza del testo inviato.
Solo i risultati affidabili verranno presi in considerazione all'interno dell'applicazione.

4.6 Efficacia della Language Detection API

Come già detto, la Langiage Detection API non è infallibile e in certi casi potrebbe assegnare un linguaggio sbagliato ad un post scritto in un'altra lingua o non riuscire a capire di quale linguaggio si tratti.

Il fallimento della identificazione dipende da diversi fattori, quali refusi nella didascalia, la presenza di più lingue nello stesso testo oppure l'insufficienza di dati da elaborare.

Per testare l'efficienza della API è stato esaminato un campione di post presi da quattro città italiane (Firenze, Bologna, Milano, Roma) ed è stata confrontata manualmente la lingua del post con quella individuata dalla Language Detection. Su duecento post, centocinque sono stati identificati correttamente, cinquantasei sono stati classificati erroneamente e trentanove non contenevano informazioni sufficienti ad essere identificati.

		Lingua individuata										
		Italiano	Inglese	Spagnolo	Portoghese	olandese	ucraino	Cinese	Coreano	Russo	Turco	Non Classificabile
Lingua del post	Italiano	35	0	1	0	0	0	0	0	0	0	24
	Inglese	1	50	0	0	0	0	0	0	0	0	23
	Spagnolo	0	0	4	0	0	0	0	0	0	0	2
	Portoghese	0	0	0	3	0	0	0	0	0	0	1
	olandese	0	0	0	0	1	0	0	0	0	0	0
	ucraino	0	0	0	0	0	1	0	0	0	0	0
	Cinese	0	0	0	0	0	0	2	0	0	0	1
	Coreano	0	0	0	0	0	0	0	2	0	0	0
	Russo	0	0	0	0	0	0	0	0	5	0	2
	Turco	0	0	0	0	0	0	0	0	0	2	0
	Non Classificabile	1	0	0	0	0	0	0	0	0	0	39

Figura 10. Confusion matrix dell'elaborazione della Language Detection API su un campione di cento post

Se si considerano solo i post identificabili, la percentuale di successo si attesta tra il 65 e il 70%.

Per meglio valutare l'efficenza dell'identificatore si può fare ricorso a tre parametri comunemente utilizzati per stimare la qualità di un classificatore: *Precision, Recall* e *F-Measure*.

Il primo rappresenta l'accuratezza con cui sono stati classificati gli elementi di una certa classe (es. l'accuratezza con cui i post in italiano sono stati identificati come tali). Il secondo misura la capacità del classificatore nell'individuare una certa classe.
La F-measure calcola una media armonica tra Precision e Recall, dandoci una stima sommaria della qualità del classificatore per ogni classe.
Tutti e tre i valori saranno sempre compresi tra 0 e 1. In particolare, un valore di Precision pari a 1 indica che tutti gli elementi classificati come appartenenti ad una certa classe appartengono effettivamente ad essa. Al contrario, un valore di Recall pari a 1 indica che ogni elemento appartenente ad una classe è stato constrassegnato come appartenente ad essa.
Un classificatore con F-measure pari a 1, quindi, assegnerebbe sempre la lingua giusta ad ogni post.
Questi tre indici sono calcolati in base agli esiti ottenuti dal classificatore su un campione di dati. Per ogni classe (nel nostro caso, per ogni lingua) avremo quattro categorie di esiti:

- **True Positive (TP)**: La lingua del post viene correttamente classificata come quella corrente.
- **True Negative (TN):** La lingua del post viene correttamente classificata con un'altra lingua.
- **False Positive (FP):** La lingua del post viene erroneamente classificata con la lingua corrente.
- **False Negative (FN):** La lingua del post viene erroneamente classificata con un'altra lingua.

Per ogni lingua quindi vengono calcolati Precision, Recall ed F-Measure in questo modo :

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Recall} = \frac{tp}{tp + fn}$$

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Basandoci sui dati riportati nella confusion matrix in figura 10, otteniamo i seguenti valori.

	TP	TN	FP	FN	Precision	Recall	F-Measure
Italiano	35	138	2	25	0,95	0,58	0,72
Inglese	50	126	0	24	1,00	0,68	0,81
Spagnolo	4	193	1	2	0,80	0,67	0,73
Portoghese	3	196	0	1	1,00	0,75	0,86
olandese	1	199	0	0	1,00	1,00	1,00
ucraino	1	199	0	0	1,00	1,00	1,00
Cinese	2	197	0	1	1,00	0,67	0,80
Coreano	2	198	0	0	1,00	1,00	1,00
Russo	5	193	0	2	1,00	0,71	0,83
Turco	2	198	0	0	1,00	1,00	1,00
Non Classificabile	39	107	53	1	0,42	0,98	0,59

Figura 11. Tabella riassuntiva dei valori Precision, Recall ed F-Measure per ogni lingua.

Per ottenere una valutazione sommaria della Language Detection API è stata calcolcata una media pesata degli indici di ogni lingua. Il peso di ognuno equivale al numero di post appartenente ad ogni classe.
I valori medi ottenuti sono: Precision 0,86, Recall 0,72 e F-Measure 0,74.
Secondo quanto ottenuto, la classificazione risulta sufficientemente efficente e soprattutto figura una buona precisione nell'assegnamento della lingua. Un po' più basso il livello di riconoscimento delle lingue indicato dalla Recall, che comunque si aggira su livelli più che accettabili.

4.7 Visualizzare i post sulla mappa

Una volta ottenuti i post e individuata la loro lingua è necessario visualizzarli sulla mappa. A tale scopo verranno utilizzati i *marker*, degli oggetti forniti da Google Maps per marcare i punti sulla mappa.
Visualizzare tutti i marker contemporaneamente, tuttavia, compromette la buona visualizzazione dei risultati, creando ammassi di icone sovrapposte.

Una soluzione molto pratica a questo problema è quello di raggruppare i marker più vicini tra loro sotto un'unica icona rappresentativa. Risolvere questo problema, tuttavia, non permette soltanto una migliore visualizzazione dei risultati, ma anche di stabilire quali siano le zone con maggior numero di post all'interno dell'area ricercata, fornendoci quindi gli indizi necessari a verificare la nostra ipotesi.

5. Ricerca dei punti d'interesse

Secondo quanto detto nell'introduzione, i post si concentrano in corrispondenza dei luoghi di attrazione di una particolare zona.
Risolvere il problema della visualizzazione dei post, quindi, è ciò che ci permette di individuare i punti d'interesse di un determinato luogo. Il compito dell'applicazione sarà ora quello di aggregare i post più vicini tra loro in modo da poter trovare tali punti con facilità. Problemi di questo tipo, in cui gli oggetti analizzati devono essere divisi in categorie ed in cui tali categorie devono essere scoperte durante il processo, vengono detti problemi di *clustering*.

5.1 Cos'è il Clustering

Con clustering si intende il processo di partizionamento di un insieme di oggetti in sottoinsiemi. Ogni sottoinsieme è detto **cluster** ed ogni oggetto al suo interno è simile agli altri oggetti che appartengono ad esso, ma diverso da quelli appartenenti ad altri cluster.
Il clustering è stato ampiamente usato in diversi ambiti, come il riconoscimento delle immagini, ricerca web, biologia, sicurezza.
Esistono diversi metodi per realizzare questo processo, e si dividono principalmente in quattro famiglie: metodo partizionale, metodo gerarchico, metodo basato sulla densità e metodo a griglia.
Il clustering partizionale prevede di dividere lo spazio dei punti in zone e di assegnare ogni punto alla zona più vicina. Questo metodo viene realizzato attraverso una procedura iterativa, che parte da un partizionamento iniziale arbitrario e lo migliora ad ogni iterazione. Un esempio classico di questo approccio è l'algoritmo K-Means.

Il clustering gerarchico invece rappresenta l'insieme dei punti come un albero le cui foglie sono i post e la radice è il cluster che li comprende tutti. Esistono due tipi di approcci per questa classe di clustering: *divisivo* e *agglomerativo*. Il primo parte dalla radice e la scompone in parti sempre più piccole fino ad arrivare ad ottenere cluster di un solo elemento. Il secondo parte dalle foglie, prese separatamente, e le unisce di volta in volta fino ad ottenere un unico nodo che rappresenta l'intera popolazione.
Il metodo basato sulla densità prevede di estendere un cluster fintanto che il numero di oggetti intorno ad esso non supera una certa soglia.
Il clustering a griglia divide lo spazio dei dati in parti uguali ed analizza la densità di ognuna di esse per poi individuare i cluster.
In questo progetto è stato utilizzato inizialmente un algoritmo di clustering partizionale (K-Means) per poi passare ad uno di tipo gerarchico.

5.2 Come funziona K-Means

K-Means è un algoritmo di clustering partizionale, ovvero, preso in input un insieme di punti, lo suddivide in *k* sottoinsiemi ed assegna ogni elemento al sottoinsieme appropriato.
Le sue applicazioni variano dal clustering all'approssimazione di una distribuzione, alla verifica dell'indipendeza di un set di variabili.
Il numero *k* deve essere scelto a priori ed è fondamentale per ottenere un buon risultato dall'algoritmo (cfr. paragrafo successivo).
Il criterio di assegnazione, in questo caso, è la distanza tra il punto in esame e la posizione del cluster, detta centroide. Il centroide è calcolato come la media delle coordinate dei punti che fanno parte del cluster.
L'algoritmo segue questa procedura:

1. Seleziona una partizione iniziale con *k* cluster.
2. Crea una nuova partizione assegnando ogni elemento al centroide più vicino.
3. Ricalcola i centroidi dei cluster ottenuti.
4. Se i centroidi sono diversi da quelli della partizione precedente, torna al punto 2, altrimenti termina l'esecuzione

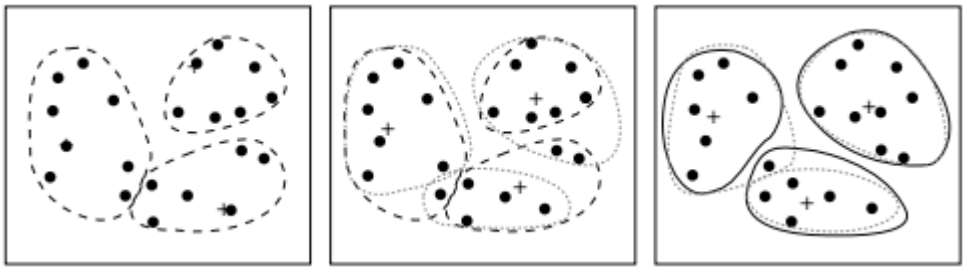

Figura 12. Assegnamenti effettuati da K-Means durante tre iterazioni

L'algoritmo è sufficientemente veloce su piccole quantità di dati (i post in tempo reale vengono elaborati quasi istantaneamente) mentre occorre un po' più di tempo quando i post si aggirano sulle decine di migliaia (qualche minuto).
La sua complessità è $O(n^{kd+1} log\ n)$, dove *d* è il numero di dimensioni dello spazio su cui vengono confrontati gli *n* oggetti.

5.3 Lo svantaggio di K-Means, il parametro *k*

K-Means richiede l'inizializzazione dell'algoritmo con un numero *k* che identifica il numero di centroidi, e quindi di cluster. Tale numero impatta notevolmente sul risultato dell'algoritmo, pertanto la scelta di *k* deve essere ponderata adeguatamente. Riuscire a predire con precisione questo numero diventa ora questione focale del problema.
Esistono diversi metodi: il più semplice, detto *Rule of Thumb* pone

$$k \approx \sqrt{n/2}$$

dove *n* è il numero degli elementi dell'insieme.
Questo metodo è molto approssimativo e quando i dati sono molti e molto vicini tra loro, k potrebbe essere sproporzionato rispetto all'effettivo numero richiesto.

Un altro metodo, detto *Elbow Method*, consiste nell'incrementare progressivamente il valore di k, in modo da diminuire un parametro detto *sum of squared errors* (SSE), ovvero la somma delle distanze al quadrato di ogni punto dal centroide del cluster di appartenenza. Questo parametro indica la discrepanza del modello ottenuto dai dati effettivi. Più piccola è la SSE, migliore sarà l'adattamento del modello ai dati, e quindi le conclusioni che se ne trarranno saranno più precise.

$$SSE = \sum_{i=1}^{k} \sum_{x \in c_i} dist(x, c_i)^2$$

In questa formula, k è il numero di cluster stabilito a priori, i è un indice che parte da 1 e arriva a k, x è un elemento dell'insieme di dati e c_i è l'i-esimo dei k cluster.
Man mano che k aumenta, l'errore diminuisce dapprima sensibilmente, poi sempre meno fino a quando la diminuzione dell'errore non è inferiore ad un parametro ε, scelto in fase di realizzazione del progetto. Quando ciò accade, significa che il modello che abbiamo ottenuto non può essere migliorato ulteriormente e che quindi abbiamo trovato il k migliore.
Anche la scelta di ε, quindi, deve essere effettuata attentamente testando il codice su valori diversi.

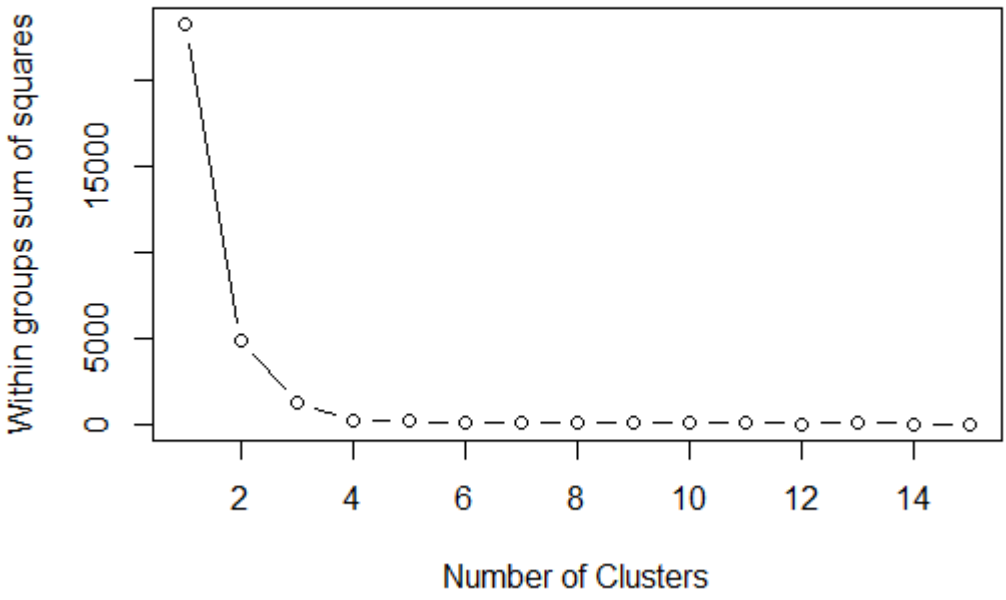

Figura 13. Grafico che mostra la diminuzione dell'errore all'aumentare di k. *Il valore migliore, in questo caso, risulta essere 4*

In questo progetto viene prima calcolato *k* con la *Rule of Thumb*, poi viene applicato l'*Elbow Method* in modo da verificare che il numero trovato sia quello giusto.
Dopo alcune prove, K-Means si è rivelato inadatto allo scopo in quanto la clusterizzazione risultava troppo approssimativa, mostrando solo cluster di grandi dimensioni senza possibilità di vederne i dettagli in modo accurato.
Altra significativa limitazione di K-Means sta nel fatto che questo algoritmo può costruire solo cluster di forma sferica o ellittica. Se un gruppo di persone pubblicasse delle foto lungo una strada, si creerebbe un cluster dalla forma molto allungata, che l'algoritmo non sarebbe in grado di identificare.
Per questi motivi si è scelto di passare ad un clustering di tipo gerarchico.

5.4 Clustering gerarchico

Un altro approccio al clustering è quello gerarchico, che prevede di organizzare i cluster secondo, appunto, una gerarchia. Questo metodo si divide in due classi: metodo agglomerativo e metodo divisivo. Nel primo, detto anche *bottom-up*, tutti i punti sono separati e vengono considerati come singoli cluster, detti *foglie*. Al primo

passo, i due punti più vicini vengono uniti per formare un cluster di due oggetti (mentre gli altri restano separati). Nei passi successivi verrano uniti i cluster più vicini, ma per fare questo sarà necessario definire come misuare la distanza tra cluster comprendenti più oggetti. Tale criterio è detto *linkage* ed i più comuni sono così definiti:

- **Single linkage**: la distanza tra due cluster è la più piccola distanza tra un oggetto del primo cluster ed uno del secondo.
- **Complete linkage**: la distanza tra due cluster è la distanza più grande tra un oggetto del primo cluster ed un oggetto del secondo.
- **Average linkage**: la distanza tra due cluster è definita come la media delle distanze tra gli oggetti appartenenti al cluster.

In questo progetto è stato utilizzato il criterio Average Linkage in quanto tende ad unire tra loro i cluster più coesi e non presenta la tendenza a creare cluster "a catena", come nel caso del Single Linkage, o di forma sferica e compatta, come nel Complete Linkage.
L'algoritmo continua ad unire i cluster fino ad ottenerne uno solo, detto *radice*, che comprende tutti gli elementi dell'inisieme iniziale.
Il metodo divisivo, detto anche *top-down*, si comporta in maniera inversa, unendo al primo passo tutti gli elementi dell'insieme in un unico cluster e dividendo, ad ogni iterazione, ogni cluster in parti più piccole fino a quando ognuno di essi non è composto da un solo elemento.

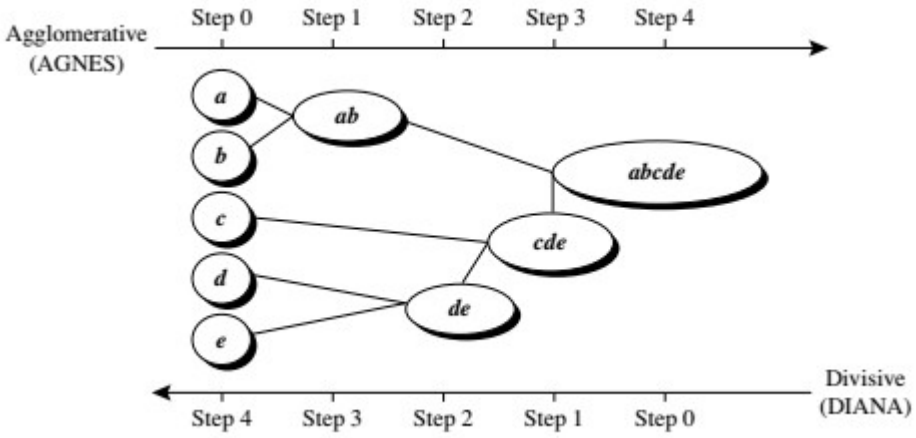

Figura 14. Algoritmi agglomerativo (AGNES) e divisivo (DIANA) a confronto

In questo progetto è stato utilizzato un algoritmo di tipo *bottom-up*, che ci ha permesso di ottenere risultati più accurati e non richiede la conoscenza a priori del numero di cluster da formare.

La complessità di questo tipo di clustering è $O(n^3)$, che lo rende troppo lento per grandi quantità di dati. Altro svantaggio è l'impossibilità di riassegnare i punti ai cluster, se non attraverso la riesecuzione dell'algoritmo. Tuttavia i risultati sono molto più pratici e completi poichè permettono di ottenere un clustering con diverse risoluzioni e di dividere i cluster in sotto-cluster per analizzarne meglio la struttura.
Questi sono i passaggi dell'algoritmo di tipo agglomerativo:

Sia $X=\{x_1,x_2, x_3, \ldots, x_n\}$ l'insieme dei punti da clusterizzare.

1. Creare una tabella delle adiacenze, in cui sono riportate le distanze di ogni punto dagli altri.
2. Trovare la coppia di punti con distanza minima (*r*,*s*).
3. Fondere i due punti in un unico cluster.
4. Aggiornare la tabella delle distanze, eliminando la riga e la colonna relative ai punti *r* ed *s* ed aggiungere una riga ed una colonna relativa al nuovo cluster, detto (*r*,*s*). La distanza tra (*r*,*s*) ed un altro cluster *k* è la minima distanza tra *r* e *k* o tra *s* e *k*. Formalmente: d[(k), (r,s)] = min (d[(k),(r)], d[(k),(s)]).
5. Se tutti i punti sono stati inclusi nello stesso cluster, terminare l'esecuzione. Altrimenti ripartire dal punto 2.

Al termine dell'algoritmo, ciò che otteniamo è una struttura ad albero in cui le foglie equivalgono ai singoli post, mentre la radice equivale al cluster che li comprende tutti. Tale struttura viene detta *dendrogramma*.

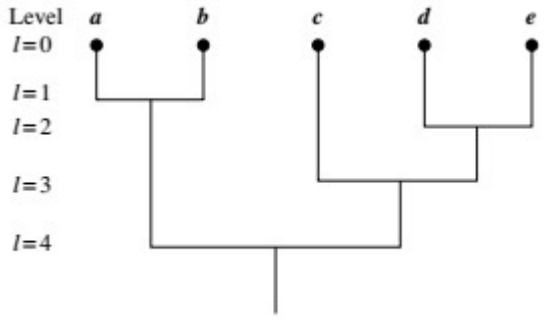

Figura 15. Il dendrogramma ottenuto dal clustering gerarchico

Per realizzare questo tipo di clustering è stata utilizzata la libreria JavaScript Clusterfck che permette di scegliere la metrica e il linkage più adatta alle esigenze dell'applicazione.
Questo tipo di clustering ha permesso di migliorare la visualizzazione dei risultati, consentendo di impostare una vista più dettagliata in base allo zoom della mappa.
In particolare, aumentando lo zoom, i cluster si dividono nei loro sottogruppi, permettendo una vista più dettagliata della distribuzione dei post. Diminuendolo, i cluster si aggregano, dandoci la possibilità di visualizzare le zone in cui si concentra l'attività social.
Ad ogni livello di zoom è associata una distanza intercluster (in metri) così calcolata:

$$m = 12 \times 2^{Z - maxZ}$$

Dove *Z* è lo zoom attuale, mentre *maxZ* è il massimo grado di zoom fornito da Google Maps. Ogni volta che lo zoom cambia, viene invocata una funzione che ricerca il livello del dendrogramma in cui i cluster distano al massimo *m* e lo visualizza sulla mappa.
Il risultato di questo algoritmo è una scomposizione dei cluster nelle loro parti più piccole all'aumentare dello zoom, ed una aggregazione in cluster più grandi al diminuire.

6. Risultati

Per usufruire dell'applicazione è stata messa a punto una interfaccia web interattiva, dove è possibile selezionare la finestra temporale in cui effettuare la ricerca e visualizzarne gli esiti sulla mappa di Google Maps.

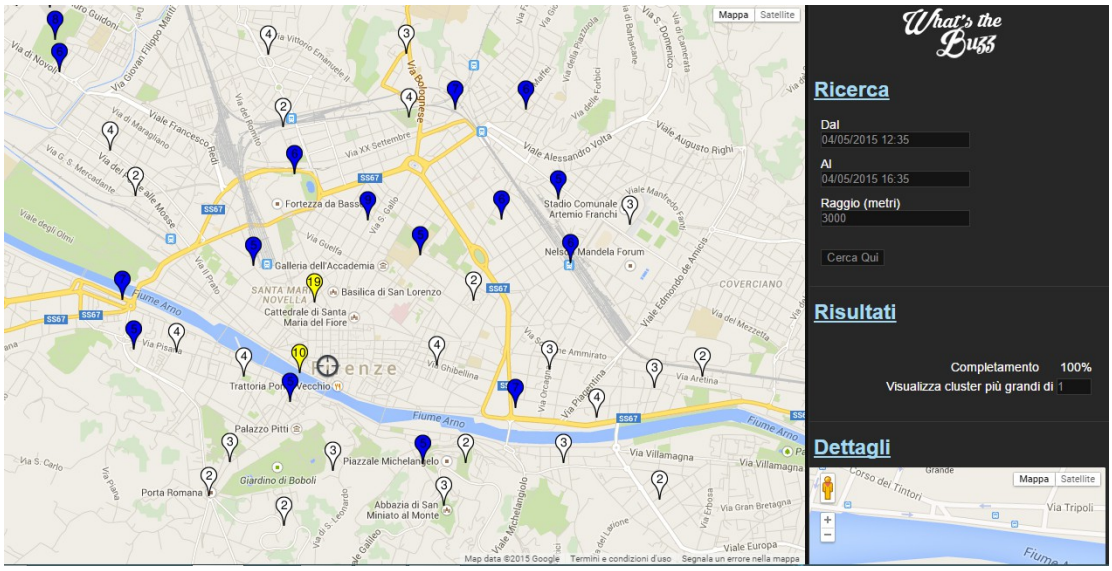

Figura 15. Interfaccia dell'applicazione

Il pannello di destra è diviso in tre sezioni: ricerca, risultati e dettagli.
Nella prima è possibile impostare i parametri sull'area e sulla finestra temporale da ricercare ed avviare quindi la procedura mediante il pulsante *Cerca Qui*. La latitudine e la longitudine saranno quelle del centro della mappa.
Cliccando sul pulsante verranno impostati automaticamente i parametri di ricerca e, mediante un codice JavaScript, verranno inviate le richieste calcolate dal programma al server di Instagram mediante delle chiamate AJAX. I dati ricevuti verranno poi processati dall'algoritmo di clustering, fornendoci il modello ad albero illustrato nei capitoli precedenti. In base allo zoom della mappa viene calcolato il livello del dendrogramma da visualizzare e per ogni cluster di quel livello viene creato un marker, che viene poi posizionato sulla mappa.
Ogni cluster è identificato da quattro variabili:

- **Canonical** - Elemento rappresentativo del cluster (scelto casualmente tra quelli appartenenti ad esso)
- **Left** - Figlio sinistro del cluster che, unito a quello destro, ha creato il cluster corrente.
- **Right** - Figlio destro del cluster che, unito al sinistro, ha creato il cluster corrente.
- **Size** - Numero di elementi che fanno parte del cluster.

La sezione Risultati ci mostra l'avanzamento dell'elaborazione e ci permette di filtrare i cluster che vediamo sulla mappa in base alla loro dimensione. Questa pratica risulta molto utile per ridurre il "rumore" prodotto dal naturale utilizzo di Instagram, che crea piccoli cluster in zone dove non si trovano elementi attrattivi.

Cliccando su un cluster, si espande la sezione Dettagli, che mostra i post appartenenti ad esso ed il loro numero (rank), la frequenza delle lingue individuate, la lista delle immagini e delle tag, ordinate per frequenza.

L'icona dei post indica la lingua individuata dalla Language Detection API (nel caso in cui non si abbiano informazioni sulla lingua, viene visualizzata l'icona di Instagram).

Figura 16. Pannello dei dettagli. Nella mappa le icone dei marker indicano la lingua individuata dalla Language Detection API

Ai marker vengono assegnati dei colori in base alla grandezza del cluster: bianco se inferiore a cinque, blu se compreso tra cinque e dieci, giallo se compreso tra dieci e venti, rosso se compreso tra venti e cinquanta e nero se maggiore di cinquanta.

Figura 17. I cinque colori dei marker in base al numero di post al loro interno

I risultati variano molto in base alla zona soggetta alla ricerca e alla larghezza della finestra temporale. Tuttavia si può notare come i cluster più grandi tendano a posizionarsi in corrispondenza delle zone più interessanti. Questo fatto diventa ancora più evidente quando vengono nascosti dalla visuale i cluster più piccoli e che spesso generano soltanto rumore.

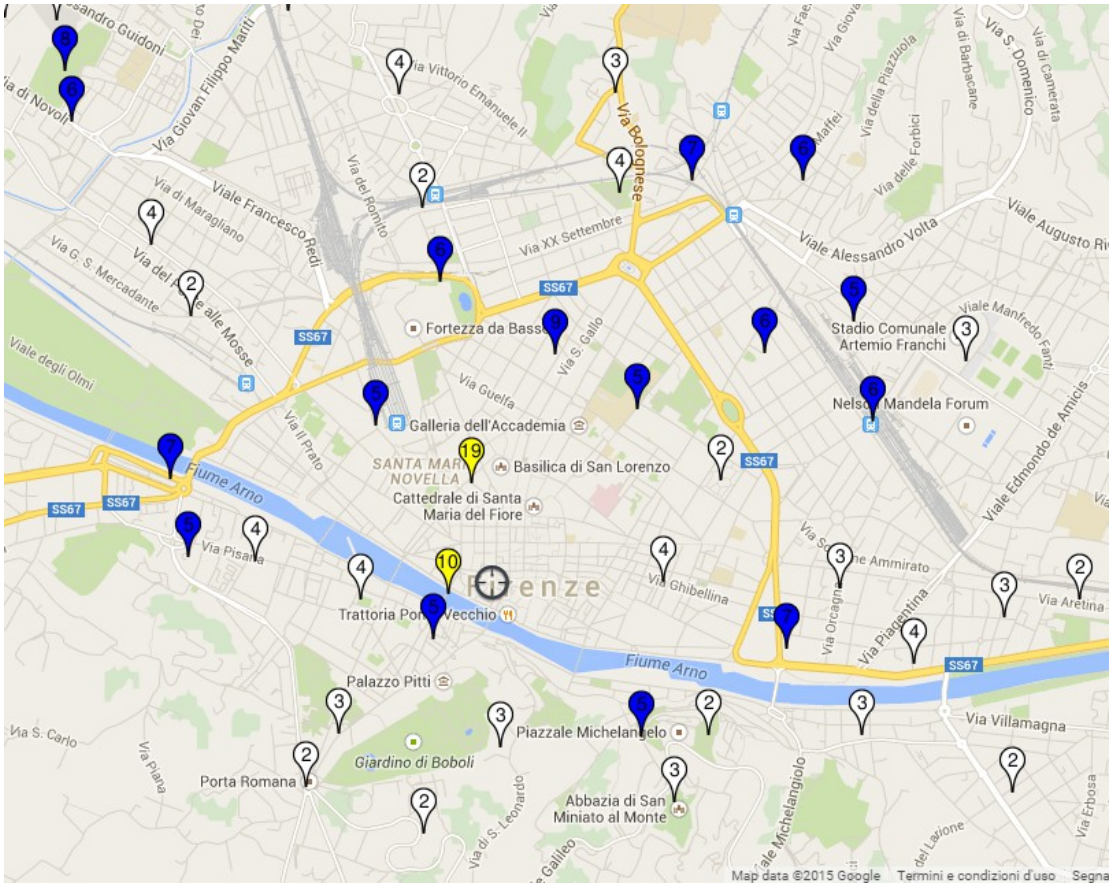

Figura 18. Un pomeriggio a Firenze

L’individuazione della lingua può fornire informazioni interessanti sui cluster più grandi, rivelando in che quantità i post appartenenti ad essi sono stati pubblicati dalla gente del posto o da turisti. Inoltre può fornire informazioni sulle nazionalità che più frequentano determinati luoghi, permettendo così di migliorare la proposta dei servizi turistici.

6.1 Implicazioni e possibilità d’uso

Dal punto di vista sociale ed economico, questa applicazione si presenta come strumento in grado di favorire la valorizzazione del territorio, così come mezzo d’indagine per la rilevazione di nuovi dati utili allo studio del comportamento della popolazione di un luogo.
Le informazioni inviate dagli utenti possono essere utilizzate anche nell’ambito della sicurezza, in quanto possono informare le altre persone nel caso si verifichi una calamità o un incidente in una determinata zona.
I dati ricavati sul linguaggio parlato dagli utenti possono permettere una migliore pianificazione delle attrattive turistiche e culturali, rivelando quali sono le nazionalità che sono più attratte da determinati luoghi o i percorsi privilegiati dai visitatori.

7. Conclusioni

Ciò che è stato ottenuto è un’applicazione in grado di dirci sommariamente quali sono le tendenze degli utenti di Instagram all’interno di una certa area. Si può notare infatti come i cluster tendano ad aggregarsi nelle vicinanze di mete turistiche o di luoghi d’interesse.

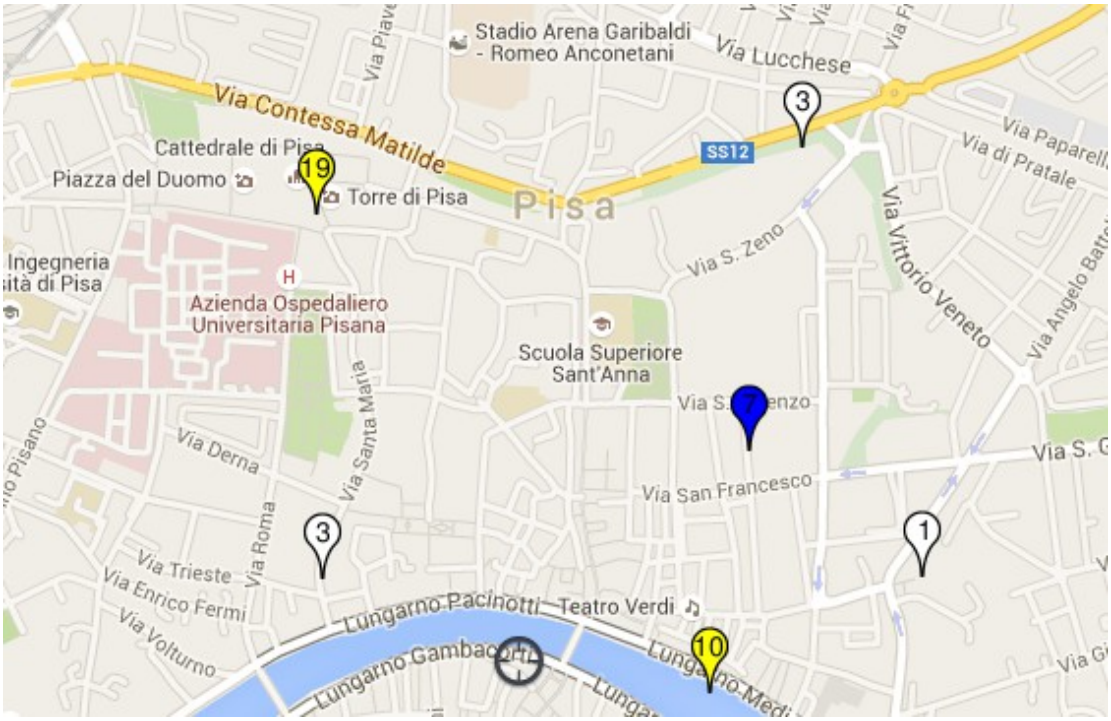

Figura 19. Mattinata a Pisa, i post si concentrano principalmente sulla Torre e sul lungarno

Come già anticipato nell'introduzione, però, in aree con alta densità di popolazione possono andarsi a creare dei cluster anche in assenza di social event.
Questi tipi di cluster sono caratterizzati da una scarsa coesione dei post, come si può notare nella figura sottostante.

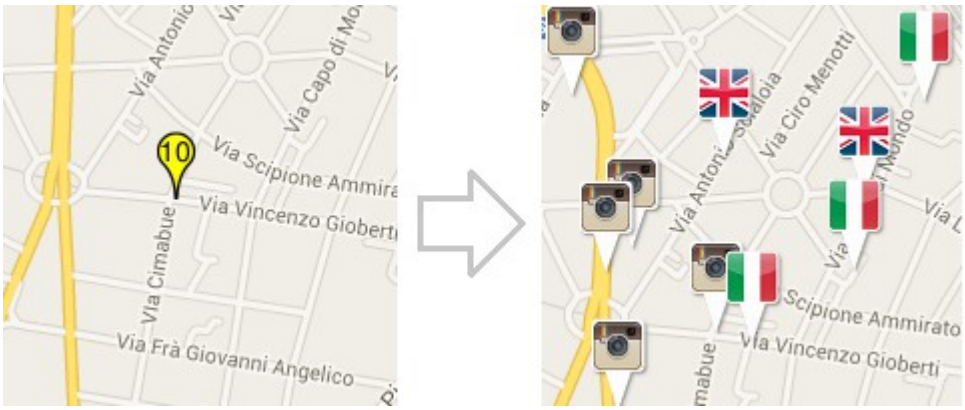

Figura 20. Diffusione dei post appartenenti ad un cluster casuale

Anche le foto tendono a fornire pochi indizi, mostrandoci immagini incoerenti tra loro, postate molto probabilmente in situazioni diverse.

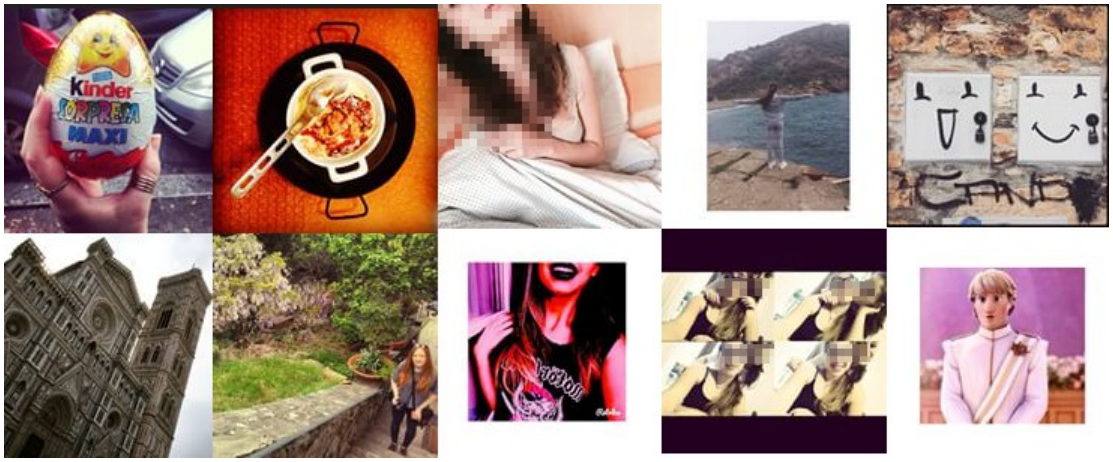

Figura 21. Immagini dei post appartenenti al cluster della figura precedente

Analizzando la lista delle tag, si può notare come neanche in questo campo i post rivelino informazioni interessanti.

Tag in questo Cluster

picoftheday(3) instamood(2) love(2) tagsforlikesapp(2)
tagsforlikes(2) fashion(2) beautiful(2) instagood(2) smile(2)
me(2) unviaggiodaricordare(1) wiw(1) (1) whatiwore(1)
mylook(1) ootd(1) whatiworetoday(1) style(1)
currentlywearing(1) instastyle(1) outfitoftheday(1)
lookoftheday(1) todayimwearing(1) kinder(1) maxi(1)
outfitpost(1) instafashion(1) clothes(1) amaretto(1)
bonodamorire(1) whatitalyis(1) santamariadelfiore(1) instapic(1)
firenze(1) instatravel(1) campanile(1) tuscan(1) ootdshare(1)
missingflorence(1) florence(1) wiwt(1) italy(1)
underthetuscansky(1) iphone6(1) duomo(1) travel(1)
iphoneonly(1) fashionista(1) ilbelpaese(1) toscanadaamare(1)
italia(1) igersitalia(1) ilpaesedellemeraviglie(1) lookbook(1)

Figura 22. Frequenza delle tag del cluster casuale.

Questi cluster in genere avranno dimensioni minori rispetto a quelli che effettivamente si trovano nelle vicinanze di punti d'interesse e quindi l'utilizzo del filtro sulla grandezza dei cluster da visualizzare può essere d'aiuto per l'eliminazione del rumore provocato da questi agglomerati di post.

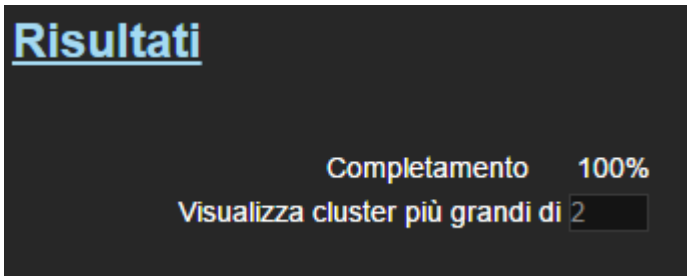

Figura 23. Interfaccia del filtro con cui è possibile rimuovere dalla visualizzazione i cluster più piccoli di una certa soglia.

Talvolta è possibile "smascherare" questi cluster casuali semplicemente zoomando su di essi, rivelando come i post che li compongono siano in realtà discretamente separati.

In uno spazio temporale più ampio, questa applicazione permette di individuare usi, costumi e abitudini dei luoghi oggetto di indagine e di osservare lo sviluppo delle capacità attrattive di eventi ricorrenti.
Le due schermate sottostanti rappresentano la diversa distribuzione dei post nel centro di Torino in due fasi della giornata. Si può notare come la concentrazione dei cluster cambi in base ai comportamenti della popolazione presente in quell'area.

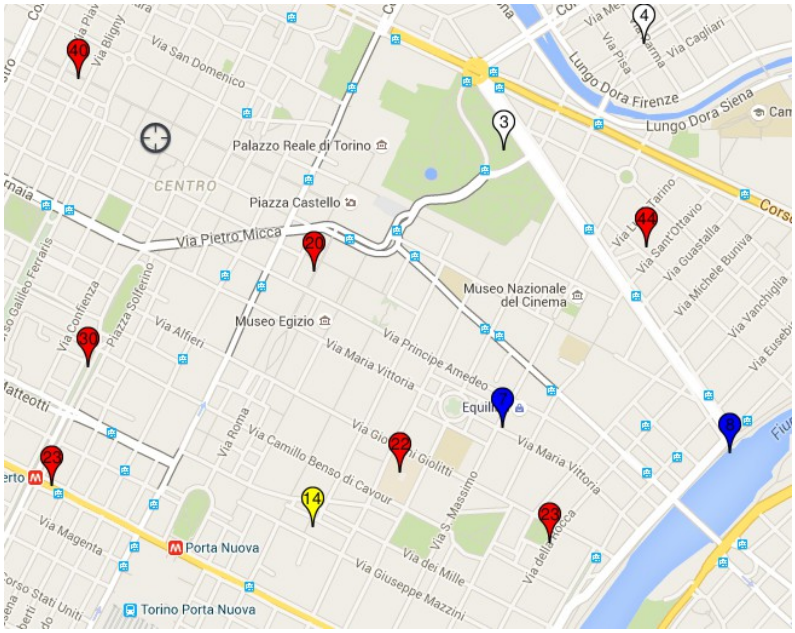

Figura 24. Post del centro di Torino dalle 15:00 alle 19:00 del 21/03/2015

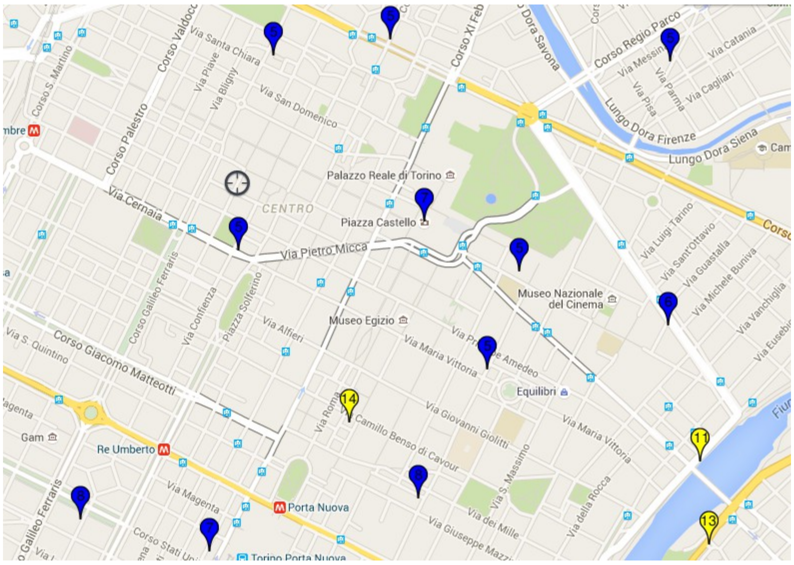

Figura 25. Post del centro di Torino dalle 21.00 alle 01:00 della notte tra il 21 e il 22/03/2015

Grazie a questa sua caratteristica, questo servizio può essere utilizzato anche come strumento di pianificazione delle attività turistiche e culturali, sfruttando anche le informazioni sulla lingua ottenute durante l'analisi. Generalmente, sui luoghi di interesse turistico si accumuleranno post in lingua straniera, a differenza delle zone meno conosciute e frequentate dagli abitanti del posto.

Capire quali siano i movimenti privilegiati dai turisti può aiutare la valorizzazione del territorio, promuovendo una migliore scelta strategica delle attività di pubblicizzazione delle attrattive turistiche.

Seguendo i cluster di dimensioni più grandi, infatti, è possibile intuire quali siano i percorsi più battuti dai visitatori ed utilizzare queste informazioni per promuovere al meglio le varie attrattive che una città può proporre. Tali scelte sono spesso significative dal punto di vista economico e sono fondamentali per sviluppo del territorio.

Analizzando invece i dati linguistici ottenuti con la Language Detection API è possible ricavare informazioni interessanti riguardo ai flussi turistici provenienti da altre nazioni. Nel grafico sottostante è riportata la frequenza delle lingue individuate dall'applicazione tra aprile e maggio 2015 nell'area della Torre di Pisa (nel grafico sono stati omessi i 2090 post non identificati dalla Language Detection).

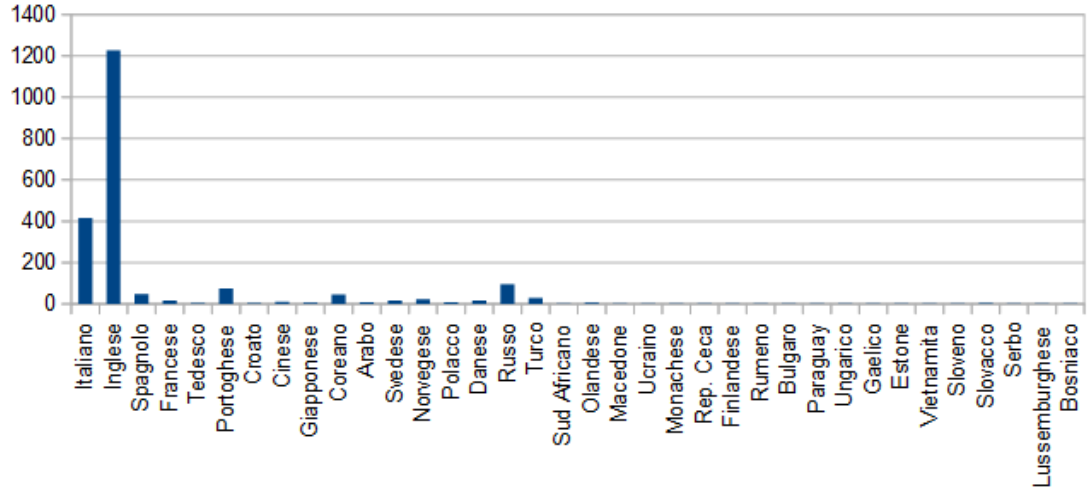

Figura 26. Frequenza delle lingue nell'area della Torre di Pisa tra Aprile e Maggio 2015

Salta subito all'occhio la discreta predominanza dell'inglese su tutte le altre lingue, persino sull'italiano, probabilmente a causa della sua grande diffusione e largo uso quotidiano in tutto il mondo. Seconda lingua in classifica è, prevdibilmente, l'italiano, la lingua ufficiale del posto, parlata sia dai residenti che dai turisti della stessa nazione. Tra le altre lingue, quelle che spiccano significativamente sono il portoghese, il russo, il coreano e lo spagnolo. Questo dato può farci intuire che c'è stato un incremento di interesse da parte dei parlanti di questi quattro idiomi nei confronti della città di Pisa e può essere sfruttato per creare un'offerta turistica mirata. Risulta utile riscontrare questi risultati con altre città, per esempio con i post ottenuti nello stesso periodo su Montecatini Terme (i 1033 post non identificati dalla API non sono stati inseriti nel grafico).

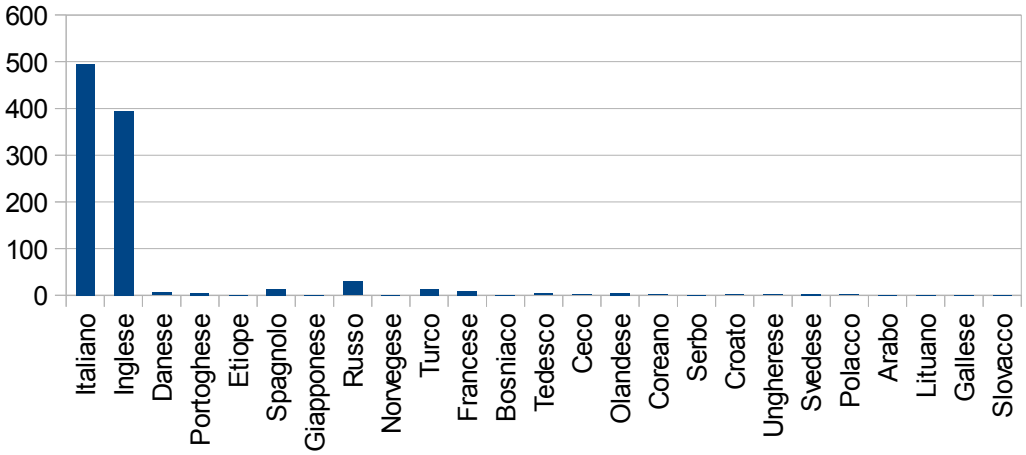

Figura 27. Frequenza delle lingue individuate a Montecatini tra aprile e maggio 2015

La prima differenza che si nota è la predominanza dell'italiano sull'inglese, al contrario di quanto avveniva per la Torre di Pisa. Inoltre, anche le lingue presenti cambiano leggermente, mostrando sempre una prevalenza del russo, ma stavolta accompagnato da turco e spagnolo.
Questi indizi possono far intuire come i flussi turistici di una città sia differente da quello di un'altra e in che maniera.
Tutte le informazioni descritte finora possono essere utilizzate per comprendere in che misura una certa area suscita interesse, verso quali utenti e in che modo, fornendo così uno strumento d'indagine utile in vari ambiti: turismo, pubblicità, marketing, sicurezza.

8. Future Works

I risultati ottenuti da questa applicazione possono essere migliorati estendendo le potenzialità del servizio. Aggiungere altri social network tra le fonti di dati può aiutare ad individuare con maggior precisione i social event e ad integrare le informazioni di ogni cluster. Riuscire a capire il contenuto dei post analizzando le immagini con metodi di riconoscimento degli oggetti così come lo studio della frequenza delle tag e di un loro clustering può essere un primo approccio per una interpretazione più approfondita delle concentrazioni di post.

I dati salvati nel database possono essere utilizzati come base statistica con cui confrontare i dati ottenuti dall'applicazione o per ricavare uno storico degli eventi passati. In particolare, possono essere utilizzati per capire se si verificano eventi anche in zone che notoriamente hanno già un'alta attrattiva.
Consentire il filtraggio dei dati in modo da eliminare i punti di interesse che hanno la naturale tendenza ad aggregare persone (come monumenti, musei), utilizzando servizi online per l'archiviazione di punti d'interesse (es. dbpedia, openpois), può aiutare l'utente a riscontrare se si stanno verificando eventi interessanti estemporanei.
A tale scopo, uno studio (Manchon-Vizuete, Gris-Sarabia, Giro-i-Nieto, 2013) ha dimostrato come utilizzare altri metadati (come le tag, il timestamp e l'id dell'utente) nella misura della distanza tra post, oltre alla geolocalizzazione, possa aumentare il livello di precisione con cui questi eventi vengono individuati. Questo approccio tuttavia presuppone l'utilizzo di un set di dati con cui addestrare l'algoritmo e dare il giusto peso ad ogni dato considerato nella metrica, che potrebbe essere ricavato dal database di cui abbiamo parlato all'inizio di questa relazione.
Disporre di questo dataset ed utilizzare una metrica basata su più metadati può aiutare ad individuare social event in tempo reale e a raffinare i risultati ottenuti dall'applicazione.
Queste sono solo alcune delle innumerevoli soluzioni realizzabili e con l'avanzare delle tecnologie sarà possibile rendere l'applicazione molto più efficace o utilizzarla come base per un livello di astrazione superiore.

9. Bibliografia

Jiawei, Han, Micheline Kamber, Jian Pei, 2012, *Data Mining, concepts and techniques*, Waltham, Morgan Kaufmann.

Cavnar, William B., John M. Trenkle, *N-gram-based text categorization*.

Kaufman, Leonard, Peter J. Rousseew, 2005, *Finding groups in data, an introduction to clustering analysis*, John Wiley & Sons Inc., Hoboken, New Jersey.

J. MacQueen, *Some Methods for Calssifications and Analysis of Multivariate Observations*, in *Proceedings of the Fifth Barkeley Symposium on Mathematical Statistics and Probability, volume I: Theory of Statistics*, a cura di Lucien Marie Le Cam, Jerzy Neyman.

Shkapenyuk V., Suel Torster, *Design and implementation of a high-performance distributed Web crawler*

Anil K. Jain *Data clustering: 50 years beyond K-means* in *Pattern Recognition Letters*, Volume 31 numero 8, giugno, 2010, Elsevier Science Inc. New York, NY, USA

Manchon-Vizuete Daniel, Irene Gris-Sarabia, Xavier Giro-i-Nieto, *Photo clustering of social events by extending PhotoTOC to a rich context.* 2013

Tamura Shingo, Keiichi Tamura, Hajime Kitakami, Kaishi Hirahara, *Clustering-based burst detection algorithm for web-image document stream on social media*, 2012

Qiu Judy, Bingjing Zhang, *Mammoth data in the cloud: Clustering social images*

Asur Sitaram, Bernardo A. Huberman, *Predicting the future with social media.* 2010

Sokolova Marina, Guy Lapalme, *A systematic analysis of performance measures for classification tasks.* 2009

10. Sitografia

Statistica Linguistica

http://www.treccani.it/enciclopedia/statistica-linguistica_(Enciclopedia-Italiana)/

Hierarchical Clustering Algorithm
https://sites.google.com/site/dataclusteringalgorithms/hierarchical-clustering-algorithm

9.2 Hierarchical Clustering
http://sfb649.wiwi.hu-berlin.de/fedc_homepage/xplore/tutorials/xaghtmlnode53.html

Single-Link, Complete-Link & Average-Link Clustering
http://nlp.stanford.edu/IR-book/completelink.html

Web Development 101: Top Web Development Languages in 2014
https://www.upwork.com/blog/2014/03/web-development-101-top-web-development-languages-2014/

11. Elenco Figure