PhD Dissertation

International Doctorate School in Information and Communication Technologies

DISI - University of Trento

Human Activity Analytics Based on Mobility and Social Media Data

Pavlos Paraskevopoulos

Advisor:
Prof. Themis Palpanas
Université Paris-Descartes, France
Università degli Studi di Trento, Italy

Thesis Committee:
Prof. Barbara Catania, *Università degli Studi di Genova, Italy*
Prof. Alessandro Moschiti, *Università degli Studi di Trento, Italy*
Prof. Myra Spiliopoulou, *Universität Magdeburg, Germany*
Prof. Athena Vakali, *Aristotle University, Greece*

April 2017

Abstract

The development of social networks such as Twitter, Facebook and Google+ allow users to share their beliefs, feelings, or observations with their circles of friends. Based on these data, a range of applications and techniques has been developed, targeting to provide a better quality of life to the users. Nevertheless, the quality of results of the geolocation-aware applications is significantly restricted due to the tiny percentage of the social media data that is geotagged (2% for Twitter). Hence, increasing this percentage is an important and challenging problem. Moreover, information extracted from social media data can be complemented by the analysis of mobile phone usage data, in order to provide further insights on human activity patterns.

In this thesis, we present a novel method for analyzing and geolocalizing non-geotagged Twitter posts. The proposed method is the first to do so at the fine-grain of city neighborhoods, while being both effective and time efficient. Our method is based on the extraction of representative keywords for each candidate location,as well as the analysis of the tweet volume time series. We also describe a system built on top of our method, which geolocalizes tweets and allows users to visually examine the results and their evolution over time. Our system allows the user to get a better idea of how the activity of a particular location changes, which the most important keywords are, as well as to geolocalize individual tweets of interest. Moreover, we study the activity and mobility characteristics of the users that post geotagged tweets and compared the mobility of users who attended the event with a random set of users. Interestingly, the results of this analysis indicate that a very small number of users (i.e., less than 35 users in this study) is able to represent the mobility patterns present in the entire dataset.

Finally, we study the call activity and mobility patterns, clustering the observed behaviors that exhibited similar characteristics, and characterizing the anomalous behaviors. We analyzed a Call Detail Record (CDR) dataset, containing (aggregated) information on the calls among mobile phones. Employing density-based algorithms and statistical analysis, we developed a framework that identifies abnormal locations, as well abnormal time intervals. The results of this work can be used for early identification of exceptional situations, monitoring the effects of important events in urban and transportation planning, and others.

Keywords[mobile devices, social networks, geolocalization, tweets, events, abnormal activity, call detail records]

Acknowledgements (Write your acknowledgments here)

Pavlos Paraskevopoulos

Contents

List of Tables

List of Figures

Chapter 1

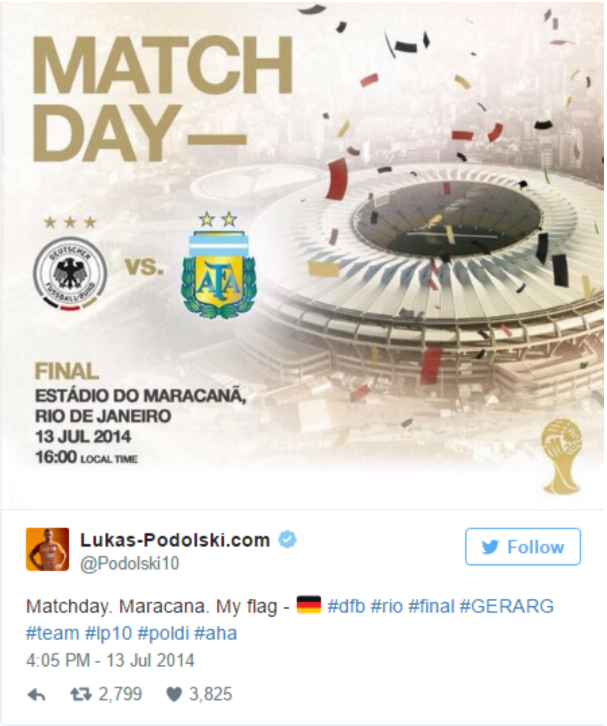
Introduction

It's the time you spent on your rose that makes your rose so important...People have forgotten this truth, but you mustn't forget it. You become responsible forever for what you've tamed. You're responsible for your rose.

Antoine de Saint-Exupery, The Little Prince

Events that happen around us affect our lives to different degrees. The effects of an event on a community vary depending on the type of the event and its dynamics. For example, traffic jams affect the way we move, football matches and concerts may affect the normal pace of life in the area of the venue for a short period of time, while earthquakes and diseases are unpredicted events, which could cause significant problems that have to be addressed fast. Many entities, public and private, are interested in analyzing the effects of such events, in order to better understand and react to them, and lead to a better quality of life. For example, the identification of lack of clean water at a place would lead the water providers to take special care for resolving the problem. Even though this would be a manual, labour-intensive, and time-consuming process in the past (e.g., consider the 1854 cholera outbreak in London [36]), this is no longer the case.

People tend to share their experiences, especially those affecting their lives (or feelings). The several social networks, such as Twitter [3], Facebook [1] and Google+ [2], that have emerged during the last decade, give users the opportunity to express themselves and report details about their everyday social activities. The combination of this behavior with the widespread use of mobile smart-phones and tablets has allowed users to report their activities in real time, adding reports from several different locations (not just from their homes, or workplaces). Consequently, we now have access to datasets containing detailed information of social activities. Furthermore, the usage of the mobile devices for accessing Internet, sending SMS or calling, generates Call Detail Records (CDRs) con-



Lukas-Podolski.com
@Podolski10

Matchday. Maracana. My flag - 🇩🇪 #dfb #rio #final #GERARG #team #lp10 #poldi #aha

4:05 PM - 13 Jul 2014

2,799 3,825

Figure 1.1: Example of Tweet that includes Text Content, an Image, the Username, and Timestamp

taining the approximate location of the user, which is recorded by the mobile network providers.

1.1 Geolocalized Posts on Social Media

Although all the data generated on social networks are important, the data generated from mobile devices are more valuable due to the fact that they can describe the events in real time, also providing the exact location.

Twitter[1] is one of the most famous social networks, counting more than 313M monthly active users, 82% of which are on mobile devices. The posts generated from Twitter are called "tweets" and they contain raw text, hashtags and the time they were posted, while the user has the option to include the location. Also, the sharing of photos and links is possible, while other users can retweet or favorite the post. An example of a tweet is presented in Figure 1.1 which is a tweet posted by Podolski, a player of the German national team, at the matchday of the final of the World Cup 2014.

[1] https://about.twitter.com/company

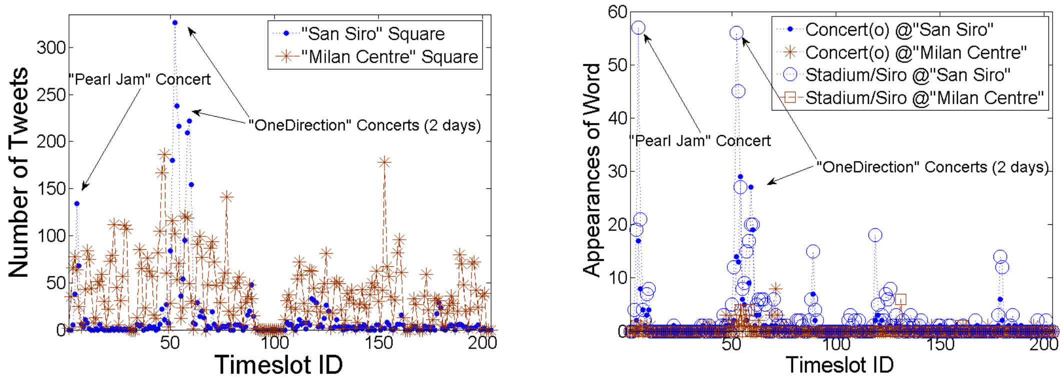
Posts like the one presented in Figure1.1 contain important information that can be used for the better and more detailed understanding of social activities. To that effect, several studies [70], including applications [8; 9; 20; 27; 45; 62; 71; 73; 79] and techniques [30; 43; 55; 69; 72; 74] have been developed that analyze datasets created through the use of social networks, in order to provide benefits to end users, businesses, civil authorities and scientists alike [57].

Several of these applications, depend on the knowledge of the user location at the time of the posting. For example, this knowledge is necessary for applications that target to characterize an urban landscape, or to optimize urban planning [27], to identify and report natural disasters, such as earthquakes [20; 62], and to monitor and track mobility and traffic [9]. Such applications, which represent an increasingly wide range of domains, are restricted to the use of geotagged data[2], that is, posts in social networks containing the geographic coordinates of the user at the time of posting.

Evidently, the availability of geotagged data, determines not only the possibility to use such applications, but also their quality-performance characteristics: the more geotagged data posts are available, the better the quality of the results will be (more accurately: the higher the probability for being able to produce better quality results). Nevertheless, the availability of geotagged data is rather limited. In Twitter, which is the focus of our study, the number of geotagged tweets is a mere 1.5-3% of the total number of tweets [31; 39; 49]. As a result, the amount of useful data for these applications to analyze is small, which in turn limits the utility of the applications. Even if we considered this subset of geotagged tweets as representative, "there is a tendency for geotaggers to be slightly older than non-geotaggers" [65], which may lead to non-representative, or skewed results.

In this thesis, we address this problem by describing a method for geolocalising tweets that are non-geotagged. Even though previous works have recognized the importance and have studied this problem [16; 37] (for a comprehensive discussion of this problem refer to [31]), their goal was to produce a coarse-grained estimate of the location of a set of non-geotagged tweets (e.g., those originating from a single user). The algorithms they propose operate at the level of postal zipcodes, cities, and geographical areas larger than cities. In contrast, we study this problem at a much finer granularity, providing location estimates for *individual* tweets first at the level of cities, and then at the level of *city neighborhoods*. More precisely, we focused on the identification of the location, where the location belonged to a set of candidate locations. This solution exploits the similarities in the content between an individual tweet and a set of geotagged tweets, as well as their time-evolution characteristics. We first determine the city, and then the neighborhood in the city, by building content-based models and analyzing the volume of posts over time,

[2] For the rest of this thesis, we will use the terms *geotagged* and *geolocalised* interchangeably.



Number of Tweets

Appearances of Words

Figure 1.2: Data generated from different neighborhoods (i.e., squares with side 1000 meters) in Milan (Italy), for time intervals of 4 hours, between June 20 and July 23, 2014.

independently for each one of these two levels. Using this set up, we are able to effectively predict the location of a post from the Twitter stream, when the only input we have is the actual content of the post and its timestamp.

In addition, we study the specific problem of geolocalising tweets deriving from targeted locations of interest, that is, neighborhoods of a particular cultural, social, or touristic importance (e.g., the Vatican in Rome). Our experiments show that we can reuse our technique for this case, as well, by adjusting its operation to this context, where a small number of popular keywords mentioned in the posts characterize the location.

Figure 1.2(a) depicts the number of tweets posted from the neighborhood in which the "*SanSiroStadium*" is located, and from a neighborhood located in the center of the Milan (Italy), while Figure 1.2(b) shows the number of appearances of the keywords *concert* (in English and Italian) and *stadium*/*siro* in these neighborhoods. As these graphs show, the "San Siro" geolocation exhibits an unusually high activity during the time intervals that coincide with the concerts that took place in this stadium. Furthermore, during these concerts, the words *concert*(*o*) and *stadium*/*siro* originate from the "San Siro" geolocation much more frequently than a random geolocation in the city.

There are two main challenges that emerge when the granularity level becomes fine: first, to maintain high accuracy despite the wider range of possible locations available to the prediction algorithm; and second, to achieve high time performance despite the increased size of the search space of the algorithm. The framework we describe for the fine-grained geolocalisation of non-geotagged tweets is based on the careful evaluation of the similarities in the content between a new, non-geotagged tweet and a training set of geotagged tweets. The solutions we propose for this similarity evaluation make use of efficient-to-compute information retrieval and statistical measures, namely, Tf-Idf among the tweet contents, and correlation among the time series representing the volume of tweets in different candidate locations. Moreover, we propose an alternative method, based on machine learning, for performing the tweet classification task, namely, Logistic Regression. The advantages of these measures are that they can effectively capture the most significant pieces of information needed to solve the problem, and that they have low time complexity.

Recognizing the need of the supervision by the user in cases of crisis events, a third challenge that emerged was to allow the user to choose whichever tweets fits his case. In order to address this issue, we built an interactive system, which is based on our geolocalization algorithms.

The focus of the system is still on the *fine-grained* location prediction: we wish to estimate the location of a post at the level of a city neighborhood and operates in both streaming and off-line manner.

Our system, provides interactive visualizations that include heatmaps for the depiction of the volume of (geotagged and geolocalized) tweets, and allows the user to zoom at different levels of granularity, ranging from a country, down to a city neighborhood. At the city neighborhood level, the user can also visualize the keywords that characterize that neighborhood. Finally, TweeLoc provides visualizations that illustrate in a comprehensive manner the changes in the volume of posts over time, for each neighborhood in a city (at a short time scale), as well as for an individual neighborhood (over long time intervals).

1.2 Characterizing User Behavior Patterns Based on The Calling Activity

Call Detail Records (CDRs) are created by the user of the mobile phone network whenever the user operates a mobile device. CRDs could be call records or SMS records, containing the location that initiated the call or sent the SMS and its destination. For the record of the (approximate) location of the mobile device, the providers use their cell-towers that distribute their signal.

Starting with the assumption that important events affect the behavioral patterns of a significant number of people in such a way that these changes are reflected in their use of the mobile telephony (CDRs), this thesis aims to develop methods for enabling the extraction, analysis, and evaluation of quantitative and qualitative information about the calling behavior patterns of users. We focus on the characterization of normal behavior patterns, the identification of exceptional, or divergent behaviors and the characterization of such behaviors (e.g., offering explanations for these behaviors). Examples of the situations we are interested in are national and religious holidays, as well as major events of local interest (e.g., sports events).

In this thesis, we study the call activity, classify the observed behaviors that exhibit similar characteristics, and we analyze and characterize the anomalous behaviors. The results of our work can be used for early identification of exceptional situations, monitoring the effects of important events in large areas, urban and transportation planning, and others.

1.3 Summary of Contributions

In this thesis, we concentrate on the problem on the lack of geotagged information and the information we can get from the usage of geotagged information. Our contributions can be broken down along 3 axes.

Geolocalization of tweets at a fine grain:

1. We describe and define the problem of fine-grained geolocalisation of non-geotagged tweets, which aims to operate on individual tweets, at the level of city-neighborhoods. We argue that the efficient solution of this problem will enable a multitude of applications that require detailed location information.

2. We propose a framework for the solution of the above problem, which is based on the content similarities of tweets, as well as their time-evolution characteristics. The solution we describe is general, and essentially parameter free.

3. We introduce a two-stage process: we first determine the city, and then the neighborhood in the city, by building content-based models and analyzing the volume of posts over time, independently for each one of these two levels. Using this set up, we are able to effectively predict the location of a post form the Twitter stream, when the only input we have is the actual content of the post and its timestamp.

4. we study the specific problem of geolocalising tweets deriving from targeted locations of interest, that is, neighborhoods of a particular cultural, social, or touristic importance (e.g., the Vatican in Rome). Our experiments show that we can reuse our technique for this case, as well, by adjusting its operation to this context, where a small number of popular keywords mentioned in the posts characterize the location.

5. we perform a detailed experimental evaluation of our approach, using real data from Twitter. The results demonstrate the efficiency and effectiveness of the proposed approach when compared to various alternatives.

6. Finally, we present the visualizations of the prototype dashboard application we have developed, which can help end-users and large-scale event organizers to better plan and manage their activities. These interactive visualizations include heatmaps for the volume of (geotagged and geolocalised) tweets, where the user can zoom at different levels of granularity, ranging from a country, down to a city neighborhood, for which the user can also explore the relevant keywords. Furthermore, we provide visualizations that illustrate in a comprehensive manner the changes in the volume of posts at different locations over time.

Social media activity and mobility analysis based on geotagged tweets:

1. We examine the difference of the activity patterns of people who attend an important event such as a concert, as opposed to the general user population.

2. We investigate the minimum possible user sample that represents the most important mobility and activity patterns of users.

3. The results indicate that user presence in special events or locations (such as an important touristic attraction, or a major concert) is often related to the activity patterns of the user.

Human behavior characterization based on mobile data:

We study the call activity and mobility patterns, classify the observed behaviors that exhibit similar characteristics, and we analyze and characterize the anomalous behaviors. The results of our work can be used for early identification of exceptional situations, monitoring the effects of important events in large areas, urban and transportation planning, and others.

1.3.1 Publications Produced

The work presented in this thesis has appeared in the following papers.

1. Paraskevopoulos P.et al. "Identification and characterization of human behavior patterns from mobile phone data." Proc. of NetMob (2013).

2. Paraskevopoulos P. and Palpanas T. "Fine-grained geolocalisation of non-geotagged tweets." Advances in Social Networks Analysis and Mining (ASONAM), 2015.

3. Paraskevopoulos P.and Palpanas T. "Where has this tweet come from? Fast and fine-grained geolocalization of non-geotagged tweets." Social Network Analysis and Mining (SNAM) Journal, 2016.

4. Paraskevopoulos P., Pellegrini G., and Palpanas T. “When a tweet finds its place: fine-grained tweet geolocalisation.” International workshop on data science for social good (SoGood - ECML PKDD), 2016.

5. Paraskevopoulos P., Pellegrini G., and Palpanas T. “TweeLoc: A System for Geolocalizing Tweets at Fine-Grain” (under submission)

6. Paraskevopoulos P.and Palpanas T.“Logistic Regression and Fine-Grain Geolocalization” (under submission)

7. Paraskevopoulos P.and Palpanas T. “What do Geotagged Tweets Reveal about the Users?” (under submission)

Chapter 2

Related Work

In order to develop our methods, we are going to use CDRs and data deriving from social media, while we combine a variety of different fields such as Statistical Analysis, Time Series Correlation and Analysis and Natural Language Processing.

In this chapter, we initially present studies that use social media posts, either targeting to the development of applications that can offer a better quality of life, or to the increase of the number of the geotagged posts. Afterwards, we present studies that have used CDRs in order to achieve their targets. CDRs can be treated in two ways, either as points or as time-series. We briefly analyze previous studies that have treated CDRs either as points or as time-series, extracting interesting conclusions for people behavior.

2.1 Geolocalisation of Social Media Posts

Several works have been developed for analyzing geotagged datasets created through the use of social networks. Target of these works is to provide benefits to end users, businesses, civil authorities and scientists alike [57].

Crowd mobility: Balduini et al. [9] studied the movement of people by analyzing geotagged tweets. The authors analyzed tweets originating from London, and more precisely close to the Olympic stadium during the Olympic games. The results show that they could identify and track the movement of the crowd, especially during the opening ceremony. In [53] the authors target to identify traffic anomalies using data from Weibo (i.e. Chinese social network) which they combine with road network data. The Dub-STAR system presented in [21] achieves a fusion between traditional city data sources and real-time social media updates in order to surface and highlight the underlying causes of current traffic conditions.

Event Identification: Some studies focus on the extraction of local events by analyzing the text in the tweets [23]. A recent study describes an approach of how to use social media data (including Twitter), in order to better understand and manage city-scale events, part of which involves the extraction of location information for tweets [8].

Abdelhaq et al. [4] use both geotagged and non-geotagged tweets for identifying keywords that best describe events. Then they keep only the geotagged tweets in order to extract the local events. Twitter posts have also been studied in order to identify the location of earthquakes [62], or fine-grained details on user activities (such as drinking alcohol) [33].

The SaferCity system [10] detects and characterizes incidents related to public safety, based on information deriving from social media. This system aims to help law enforcement entities have obtain a more detailed picture on activities happening in a city, even those that may not be officially reported.

Points of Interest: The identification of POIs with temporal awareness is the focus of a recent study [40]. The authors are analyzing tweets posted by Singaporean users, while using Foursquare check-ins referred in the tweets. Another study proposed a framework that can automatically recognize POIs by correlating geotagged tweets with geotagged data deriving from Flickr [75]. The goal is to identify places, such as restaurants and hotels, that are not already part of databases such as LinkedGeoData, Geonames, Google Places, or Foursquare.

Human Mobility: Third party information have been combined with social media data in order to identify human mobility. The combination of data from Foursquare and the logs of executed applications on a smart-phone, has been used in order to predict the next location of the users [44]. The authors of [18] examine the movement of the users combining CDRs and social media posts, while in [32] the authors target to understand urban human activity and mobility patterns using large-scale location-based data from online social media.

All the previously described studies achieve their target using tweets that are already geotagged, or tweets that mention the venue and/or event of interest, for a predefined set of venues and events. On the other hand, our thesis present a framework that is able to estimate the location of tweets regardless of the use of hashtags, or any other entity reference in the content, leading to a general solution that is effective even in cases where of tweets referring to an unforeseen event (e.g., an accident), or tweets that do not explicitly mention the venue.

Extracting User and Tweet Locations: The problem of using tweets in order to identify the location of a user, or the place that an event took place has been studied in the past. The "who, where, what, when" attributes extracted from a user's profile can be used to create spatio-temporal profiles of users, and ultimately lead to identification of mobility patterns [78]. Cheng et al. [17] create location profiles based on idiomatic keywords and unique phrases mentioned in the tweets of users who have declared those locations as their origins. The similarity between user profiles and location profiles has also been used in [16]. In this approach, they create user profiles for the active users, and extract the keywords that are characteristic of specific locations (i.e., they usually appear in some location, and not in the rest of locations). For the extraction of these keywords they initially assign weights using the Geometric-Localness (GL) method, and then prune them using a predefined keyword-weight threshold. This leads to a set of representative keywords for each location, which allows the algorithm to compute the probability that a given user comes from that location. A recent study evaluates the GL method, and compares it to other methods that solve the same problem. The experimental evaluation shows that the GL method achieves the best results [31].

Two studies that target to geotag tweets are presented in [24] and [54]. These two methods create chains of words that represent a location by using Latent Dirichlet Allocation (LDA) [11]. The latter study takes in addition into consideration the location a user has recorded as their home location. A study that predicts both a user's location and the place a unique tweet was generated from is presented in [37]. In this study, the authors construct language models by using Bayesian inversion, achieving good results for the country and state level identification tasks. Finally, [64] presented a method for identifying the geolocation of photos by using the textual annotations of these photos.

Even though some of these studies are closely related to our work (e.g., [16; 37], which we further discuss in chapter 3), we observe that they operate at a very different time and space scale. The profiles they create involve the tweets generated over a long period of time (up to several months), and the location that has to be estimated is the location of origin of the user, rather than the location from where a particular tweet was posted. Moreover, the space granularity used in these studies ranges from postal zipcodes to areas larger than a city. On the contrary, in our work we predict the location of individual tweets, at the level of city neighborhoods.

Two studies that target to geolocalize unique tweets are presented in [35] and [63]. The first method trains a model using past messages associated to locations, by extracting keywords that are connected to this location. In the later study the authors develop a multi-indicator approach that combines information from the user's profile and the tweets's message for estimating both the location of a unique tweet and a user's residence location. The main difference to our approach is that these methods rely on users that post many tweets in a time interval t, or on data from the user's profile. In contrast, we target to geotag tweets even from users that have never posted before, or do not provide any profile data (such as their home location).

A recent survey presents methods relevant to location inference [5].

With our work, we advance the state-of-the-art with the development of a novel method for analyzing and geolocalizing non-geotagged Twitter posts. The proposed method is the first to do so at the fine-grain of city neighborhoods. The geolocalization of an increased number of posts, could allow us to get a more detailed picture of an event's impact, while identifying actionable insights hidden in the non-geotagged posts.

2.2 Analyzing Mobile Phone Usage Data

Calls placed from mobile phone devices are traced in logs which can serve as an indication to understand personal and social behaviors. Researchers in the areas of behavioral and social science are interested in examining mobile phone data to characterize and to understand real-life phenomena [6; 19; 26; 38; 66], including individual traits, as well as human mobility patterns [7; 14], communication and interaction patterns[14; 52; 61]. Furthermore, studies that target to predict the semantic of a place [34; 48; 80] have been developed, while others target to predict the next location of a user [25; 29; 76] or to identify the demographics of a location[12; 47; 51].

Dynamics of call activity: Candia et al. [15] proposes an approach to understand the dynamics of individual calling activities, which could carry implications on social networks. The author analyzed calling activities of different groups of users; (some people rarely used a mobile phone, others used it more often). The cumulative distribution of consecutive calls made by each user is measured within each group and the result explains that the subsequent time of consecutive calls is useful to discover some characteristic values for the behaviors. For example, peaks occur near noon and late evening. The fraction of active traveling population and average distance of travel are almost stable during the day. This approach can be applied for detecting anomalous events.

Moreover, a number of interesting approaches propose to analyze mobile phone data to understand personal movement patterns, in particular individual tracking and monitoring [13; 60; 67], and behavioral routines [22].

Human mobility: Furletti et al. [28] extract user profiles from mobile phone data. The authors analyze moving human behaviors which correspond to specific human profiles (such as commuter, resident, in-transit, tourist), inferred by profile assumptions. A classification technique based on neural networks, called self organizing map, is used to classify users by similar profiles that have temporal constraints based on their temporal distributions. The result shows that the percentage of residents was compatible with the customer statistics provided by the Telecom operator, and short-ranged temporal profiles like commuter and in-transit are significantly different and distinguishable from the profiles with a larger extent like resident. The authors tested their approach on a case study in the city of Pisa (Italy). The data consists of around 7.8 million call records during the period of one month. They identified a peak which was caused by the reporting of an earthquake news. The authors highlighted the necessity to align temporal call distributions with a series of high level observations concerning events and other contextual information coming from different data-sources, in order to have more specific interpretation of the phenomena.

Phithakkitnukoon et al. [59] analyze the correlation of geographic areas and human activity patterns (i.e., sequence of daily activities). pYsearch (Python APIs for Y! search services) is used in order to extract the Points of Interest (POIs) from a map. The POIs are annotated with activities like eating, recreational, shopping and entertainment. The Bayes theorem is then used to classify the areas into a crisp distribution map of activities. Identifying the work location as a frequent stop during the day from the trajectories of individuals, it derives the mobility choices of users towards daily activity patterns. The stop extractions are done in the same way as in [14]. The study shows that the people who have same work profiles have strongly similar daily activity patterns. But this similarity is reduced when the distance of work profile location of the people are increased. Due to the limitation of heterogeneity of activities in this paper, the result includes some strange behaviors, like shopping during the night in the shopping area, which cannot be explained by the ground truth of activities.

Anomaly detection: Candia et al. [15] propose a simple approach to detect exceptional situations on the basis of anomalies from the call patterns in a certain region. The approach partitions the area using Voronoi regions centered on the cell-towers, and computes the call pattern in the “normal situation”. These patterns are compared with the actual data and anomalies are detected with the use of the percolation method. At the work presented in [50], Naboulsi et al. achieve to successfully identify unexpected traffic profiles by classifying and characterizing the mobile traffic profiles.

Mobility patterns: In [14] the author analyze the mobility traces of groups of users with the objective of extracting standard mobility patterns for people in special events. In particular this work presents an analysis of anonymized traces from the Boston metropolitan area during a number of selected events that happened in the city. They indeed demonstrate that people who live close to an event are preferentially interested in those events.

Another interesting work on CDRs, based on the probabilistic models, is the one presented by Tanahasi et al. in [68]. In this study the authors propose a framework that analyzes CDRs, creating trajectories that are used to divide the map into representative areas, according to the movement of the users. This study achieves a very interesting result that divides the region of New York into suburbs by using only CDRs. Furthermore, it extracts representative life-patterns using Naive Bayes Models, while it tries to get the semantic of a place by using the measurement of the Entropy. On the other hand, studies such as [41] classify trajectories by using Markov Chains. In this study Lima et al. are exploiting the cellular data for monitoring and predicting the spread of epidemics in Ivory Coast. The idea of this study is based on the probability that a user has to follow a predefined path. The main reason that forces the two last studies to use different probabilistic model is the structure of the data that they use. The trajectories that are constructed in [41] are continuous while those in [68] derive from data that are not continuous, having time-gaps and making it impossible to be manipulated as data-series.

Social response to events: The social response to events, and behavior changes in particular, have been studied by J.P.Bagrow et al. [7]. The authors explored societal response to external perturbations like bombing, plane crash, earthquake, blackout, concert, and festival, in order to identify real-time changes in communication and mobility patterns. The results show that from a quantitative aspect, behavioral changes under extreme conditions are radically increased right after the emergency events occur and they have long term impacts.

Crowd mobility: Calabrese et al. [14] characterize the relationship between events and its attendees, more specifically of their home area. The consecutive calls are measured in the same manner as in [15], in order to determine the stop duration of the trajectories. Given an event, for each cell-tower of the grid, the count of people who are attending that event, and whose home location does not fall inside that cell-tower, describes the attendance of events in geo-space. Most of the people attending one type of event are most probably not attending other types of event and people who live close to an event are preferentially attracted by it. As a consequence, the approach could partly predict starting locations of people who are coming to the future events. This could be useful to determine anomalies and additional travel demands for the capacity planning considering the type of an event. Conversely, knowing event interests of people helps to detect the event. But estimating the actual number of attendees and validating the models is still an open problem due to the presence of noise in the ground truth data. So, it derives to other issues like refining mobility patterns belonging to the events which occurred in the similar region at a closer time, and distinguishing home locations for people who live in the same location where events are organized.

We advance the state-of-the-art with the development of three easy-to-apply methods for identifying patterns and outliers in the behavior of mobile phone users, based on the analysis of the data recorded by cell-towers. Our methods can be used to identify important events, such as festivals and public holidays.

Chapter 3

Fine-Grained Geolocalization of Non-Geotagged Tweets

3.1 Introduction

The rise in the use of social networks in the recent years has resulted in an abundance of information on different aspects of everyday social activities that is available online, with one of the most prominent and timely source of such information being Twitter. This has resulted in a proliferation of tools and applications that can help end-users and large-scale event organizers to better plan and manage their activities. In this process of analysis of the information originating from social networks, an important aspect is that of the geographic coordinates, i.e., geolocalisation, of the relevant information, which is necessary for several applications (e.g., on trending venues, traffic jams, etc.). Unfortunately, only a very small percentage of the twitter posts are geotagged, which significantly restricts the applicability and utility of such applications. In this thesis, we address this problem by proposing a framework for geolocating tweets that are not geotagged. Our solution is general, and estimates the location from which a post was generated by exploiting the similarities in the content between this post and a set of geotagged tweets, as well as their time-evolution characteristics. Contrary to previous approaches, our framework aims at providing accurate geolocation estimates at fine grain (i.e., within a city). The experimental evaluation with real data demonstrates the efficiency and effectiveness of our approach.

3.2 Problem Formulation

The problem we want to solve in this work is the estimation of the geographic location of individual, non-geotagged posts in social networks.

Problem 1: Given a set of geotagged posts $P^{l_1}_{t_j}, ..., P^{l_i}_{t_j}$, $t_1 \leq t_j \leq t_2$, where l_i is the location the post was generated from and t_j is the time interval during which the post was generated at, and a non-geotagged post Q_{t_q}, $t_1 \leq t_q \leq t_2$, we wish to identify the location l from which Q was generated.

The timestamps t_1 and t_2 represent the start and end times, respectively, of the time interval we are interested in.

In the context of this work, we concentrate on fine-grained location predictions: we wish to estimate the location of a post at the level of a city neighborhood (which is usually much smaller than a postal zipcode). Furthermore, we focus on twitter posts, whose particular characteristics are the very small size (i.e., up to 140 characters long), and the heavy use of abbreviations and jargon language.

3.3 Proposed Approach

In this section, we describe our solution to the problem of fine-grained geolocalisation of non-geotagged tweets.

We provide a high level description of our approach in Algorithm 1. Our method is based on the creation of vectors describing the Twitter activity in terms of important keywords for each geolocation we have data from, and for the period of time we are interested in. The geolocations correspond to fine-grained spatial regions (in our study, they are squares with side length of 1000 meters). The time intervals correspond to brief

Algorithm 1 Tweet Geotagging Algorithm

INPUT: A training set of timestamped and geotagged tweets, a timestamped query-tweet (Q_t) that is not geotagged.

OUTPUT: The most eligible candidate location.

```
for all $i \in$ {candidate geolocations: Geolocs} do        ▷ process training dataset, for all locations
    for all $t \in$ {time intervals} do                      ▷ and for all time intervals
        $Doc_{i_t} \leftarrow$ all tweets in location $i$ at time interval $t$
        $kwVector_{i_t} \leftarrow$ create vector of $Doc_{i_t}$ keywords and their weights
$kwVector_{Q_t} \leftarrow$ create vector of $Q_t$ keywords and their weights      ▷ process non-geotagged tweet $Q_t$
$location \leftarrow argmax_{i \in Geolocs}$ {similarity between $kwVector_{i_t}$ and $kwVector_{Q_t}$}  ▷ identify location of tweet $Q_t$
return $location$
```

time segments, during which posts on the same, or related topics may be observed (in our study, they are 4 hour intervals). The vectors represent the weights of each keyword, and are stored in $kwVector$ for each geolocation and time interval. There are several ways to compute these weights: we consider the number of appearances of a keyword in a given geolocation, and the significance of a keyword, measured using Tf-Idf, for a given geolocation and the entire dataset.

In order to identify the geolocation for a non-geotagged tweet, Q, we compute the similarity between the vector of Q and the vector of each candidate geolocation. When calculating this similarity, we can additionally take into account the correlation between the local and the global activity time series, i.e., the evolution over time of the number of tweets in a given geolocation and all the geolocations, respectively. Furthermore, the usage of the slope of the time-series, allows us to filter out candidate locations whose activity gets decreased. Finally, the algorithm returns the geolocation with the highest similarity value.

In the following sections, we elaborate on the methods discussed above.

3.3.1 Grouping the Posts and Extracting Important Keywords

We start by processing the training set of geotagged posts. We group these posts according to the geolocation that they were generated from, and the time interval they belong to. After this grouping step, we calculate the concordance of the keywords in each group: the dictionary containing the number of appearances of each keyword in a geolocation. At the end, we have for each geolocation and time interval a vector of the important keywords, along with the corresponding weights. We call the algorithm that uses this method for generating the keyword vectors *TG (Tweet Geotagging)*.

We observe that concordance is a simple measure that only accounts for the frequencies of keywords, but fails to take into account their relative significance. Therefore, we also employ the Tf-Idf model: $df_{keyword} = \log(\frac{n}{k})$, where n is the number of documents, k is the number of documents that keyword appears in, and $tfidf_{i,keyword} = \frac{count}{l} * df_{keyword}$, where l is the total number of keywords in document i. Using Tf-Idf, we can calculate the significance of each keyword in our training dataset (according to the former equation above), and set the weight for a keyword in some geolocation, depending on the number of its appearances at this geolocation (according to the latter equation). This method leads to high weights for the keywords that appear at a small number of geolocations. As a final step, we sort the keywords according to their weight and prune the keywords with low weights, and therefore, only keep the significant keywords for each geolocation, which correspond to the keywords that best characterize the activity of the given geolocation at a particular time interval. We call the algorithm that uses this method for generating the keyword vectors *TG-TI (Tweet Geotagging Tf-Idf).*

In order to create the keyword vector for the non-geotagged tweet, Q, that we wish to geolocalise, we follow the same process as before, using the *idf* (i.e., the number of the locations the word appears in) as extracted from our training phase.

3.3.2 Similarity Calculation and Best Match Extraction

Our next target is to calculate the similarity between the keyword vector of Q and the keyword vector of each one of the candidate geolocations.

We follow the steps presented in Algorithm 2. The magnitude, mag is the Euclidean Norm, computed over all the keywords that appear in the vector. We calculate the magnitude of the Q vector, mag_{Q_t}, and of each one of the candidate geolocations i, mag_{i_t}, for a given time interval t. We denote with $kwVector[j]$ the weight of the j-th term of the vector. The similarity is computed using the formula shown in line 5 (over all the keywords that appear in both the vector Q and the vector of the geolocation i). The algorithm stores in a list the similarity values for each candidate geolocation which afterwards returns. We can then normalize these values over the sum of all similarities, giving us the probability that each candidate geolocation produced Q (Algorithm 3). Transforming these values into a probability distribution gives us more flexibility: for example, as we discuss next, we can readily combine this similarity measure with similarities computed using other methods. Furthermore, we can use the probability values in order to produce geolocation predictions only in the cases where we are confident (i.e., these probabilities are high). At the end, the algorithm returns the geolocation(s) with the highest probability(ies).

In our approach, this similarity calculation can happen in two phases (using for both the same general method presented above), either by examining only the neighborhood level, or combining the city level and the neighborhood level. In case of the combination, we first determine the similarities between the Corse-Grain Locations (CGL) and the Q. Having extracted the similarities with the CGLs, we proceed to the next stage and check all the Fine-Grain Location (FGL, i.e., square with side 1000 meters) within that city

Algorithm 2 Similarity Calculation

```
1: procedure VectorSim(vectors for $Q_t$ and candidate geolocations)
2:     $mag_{Q_t} \leftarrow \sqrt{\sum_{\forall j \in kwVector_{Q_t}} kwVector_{Q_t}[j]^2}$
3:     for all $i \in Geolocs$ do
4:         $mag_{i_t} \leftarrow \sqrt{\sum_{\forall j \in kwVector_{i_t}} kwVector_{i_t}[j]^2}$
5:         $Sim_{i_t,Q_t} \leftarrow \frac{\sum_j kwVector_{i_t}[j]*kwVector_{Q_t}[j]}{mag_{i_t}*mag_{Q_t}}, \forall j \in kwVector_{Q_t} \cap kwVector_{i_t}$
6: return $Sim_{Q_t}$
```

that are subsets of an eligible CGL (once again using the $VectorSim$ function). We then combine the FGL probability with the CGL probability, multiplying them and getting a unique value for each eligible FGL. At the end, the algorithm returns the geolocation with the highest total probability (refer to Algorithm 4).

3.3.3 Similarity Based on Correlation of Activity Time Series

The similarity measure that we discussed earlier is based entirely on the contents of the relevant posts, but ignores other useful characteristics of the data. In what follows, we describe a method that exploits the time-evolution behavior in order to derive an additional similarity measure.

This method is based on the activity time series, which record the number of posts generated by a given geolocation over time. We call these series *local activity* time series. We also compute the *global activity* time series, where we record the sum of the number of posts for all geolocations over time. The similarity is then expressed as the correlation value between the local activity of a candidate geolocation with the global activity. The intuition is that posts about an important event will significantly change the local activity and influence in the same way the global activity.

This process is shown in Algorithm 5. Since we are only interested in similar behavior between local and global activity, we only keep the positive correlations. More specifically, we construct the global activity time series, Gts_i, for a CGL (e.g., a city i), as well as the local activity time series, Lts_j, for all the FGL within the coarse-grain one (e.g., the neighborhoods j inside the city i). The correlation we compute between these time series is the Pearson's correlation. Finally, the algorithm returns the correlation of each candidate geolocation, added 1 in order to be shifted to the range [1,2].

We can then combine this method with the TG algorithm, by multiplying the two similarity measures (based on concordance and correlation), to obtain the *TG-C (Tweet Geotagging with activity Correlation)* algorithm. When we do the same with the TG-TI algorithm, we get the *TG-TI-C (Tweet Geotagging with Tf-Idf and activity Correlation)* algorithm.

Algorithm 3 Probability Calculation

```
procedure ProbCalc(similarities between $Q_t$ and candidate geolocations ($Geolocs$))
    for all $i \in Geolocs$ do                ▷ Get the probability distribution
        $Prob_{i_t,Q_t} \leftarrow \frac{Sim_{i_t,Q_t}}{\sum Sim_{Q_t}}$
    SortDescending $Prob_{i_t,Q_t}$
return $Geolocs$ and their $Prob_{i_t,Q_t}$
```

3.3.4 Filtering out Candidate Locations Based on Linear Regression

We note that the method described at the previous paragraph can sometimes produce undesirable results. The reason is that this method employs the correlation measure irrespective of the trend exhibited by the local and global activities. For example, these activities can be positively correlated, but have a negative trend (i.e., activity is diminishing). Evidently, in such cases the correlation does not help, and should not be taken into account.

We now describe a modified technique that addresses this problem. More specifically, we consider a location as a candidate location only if both the local and the global activity increase. As we demonstrate later, this modification on the usage of the correlation measure leads to a significantly better result.

This modified correlation-based technique is shown in Algorithm 6.

Initially, we construct the global activity time series, Gts_i, for a coarse-grain geolocation (e.g., a city g), as well as the local activity time series, Lts_{CL}, for all the fine-grain geolocations within the coarse-grain one (e.g., the neighborhoods j inside the city i). Since we are only interested in similar behavior between local and global activities only in the case where we have an increasing trend, we use the linear regression line in order to test this trend. In particular, we use the λ parameter of the equation representing the linear regression line, $y = \lambda * x + b$ (refer to line 4). If λ is positive, we assume that the time-series has a positive slope (lines $5 - 6$ and $12 - 13$). In this process, we use smaller sliding sub-windows of size $n/2$ (lines 3 and 10), sliding them across the original window. As a result, we have a sub-window that slides $n/2$ times on the original n-timeslot win-

Algorithm 4 Two-Step Similarity

```
procedure TWO-STEP SIMILARITY($similarity_{CGL}, similarity_{FGL}$)
    for all $j \in CandCGL$ do
        for all $FGL_i \in j$ do
            $TwoLevelSimilarity_{FGL_i} \leftarrow similarity_j * similarity_{FGL_i}$
    $location \leftarrow argmax_{i \in FGLs}\{ProbCalc(TwoLevelSimilarity)$        ▷ identify location of tweet $Q_t$
return $location$
```

Algorithm 5 Activity Correlation

```
procedure CORRELATIONSIM(global $Gts_i$ and local $Lts_j$ activity time series)
    for all $j \in i$ do
        $corr_{i_t,j_t} \leftarrow \frac{\Sigma(Gts_{i_t} - G\bar{t}s_i)(Lts_{j_t} - L\bar{t}s_j)}{\sqrt{\Sigma(Gts_{i_t} - G\bar{t}s_i)^2 \Sigma(Lts_{j_t} - L\bar{t}s_j)^2}}$
        if $corr_{i_t,j_t} \geq 0$ then
            $correlations_{i_t,j_t} \leftarrow corr_{i_t,j_t} + 1$
return $correlations$
```

dow, counting the number of the slides that result into positive linear regressions for both time-series describing the local and the global activity.

After having calculated all the λ for each candidate locations, we calculate the Pearson correlation between the time-series describing the local and the global activity, and add 1 to this value, in order to shift the range of values between [0,2] (line 14). This has the desirable effect that we avoid negative similarities (that would result from negative correlations). Note that candidate locations that correspond to positive correlation receive a bonus (they get multiplied by a number in the range (1,2]), while those that correspond to a negative correlation get penalized (they get multiplied by a number in [0,1)). Finally, we set a threshold th_{LR}, and we check if the number of the sliding windows for each location that have positive λ is greater than th_{LR} (line 7 and 16).

If the number of the sliding sub-windows that have positive λ exceeds th_{LR}, then this location is considered as a candidate location, and we assign to the location its correlation and the value $True$ for exceeding the threshold (line 17), otherwise we assign to the location its correlation and the value $False$ (line 19). Finally, the algorithm returns the final set of Candidate Locations, CL, which includes for each location its correlation value and the attribute that indicates if the location exceeds the th_{LR} threshold (line 22).

We can then combine this method with the TG algorithm, by multiplying the two similarity measures (concordance similarity and correlation), to obtain the *TG-CLR (Tweet Geotagging with activity Correlation with Linear Regression)*. When we do the same with the TG-TI algorithm, we get the *TG-TI-CLR (Tweet Geotagging with Tf-Idf and activity Correlation with Linear regression)* algorithm. If the candidate location that has the greatest similarity with the non-geotagged tweet Q does not exceed the th_{LR} threshold (i.e., it has been assigned the value $False$), then we do not match Q to any location.

3.3.5 Sliding Windows

We observe that previous methods use all past data in order to build their models. Methods such as [37] and [16] start building their models taking into consideration all available data. However, this may lead to situations where some local events may be mishandled. For example, consider the case, where a concert takes place in a city, followed by a second concert the following day. Then, a model that is based on all the data (and in the absence of specific and detailed keywords) is likely to assign the tweets relevant to the second concert to the location of the first concert, for which more data are available.

In order to avoid similar problems, we can use a tumbling window model [56]. Although this helps to address the problem mentioned above, tumbling windows may still mis-assign tweets that are generated at the beginning, or at the end of the window, and are connected to an event that is outside the window period.

Algorithm 6 Activity Correlation with Linear Regression

```
procedure CorrelationSim(global $Gts$ and local $Lts_{CL}$ activity time series, threshold $th_{LR}$, Candidate Locations $CL$, Window time intervals $[t_1, t_2]$)
    $counter_g \leftarrow 0$
    for all $subWindow_i \in Window$ do    ▷ For how many subWindows Global time-series have positive $\lambda$
        $\lambda_g \leftarrow \frac{\Sigma((x-\bar{x})(y-\bar{y}))}{\Sigma(x-\bar{x})^2\Sigma(y-\bar{y})^2}$
        if $\lambda_g \geq 0$ then
            $counter_g \leftarrow counter_g + 1$
    if $counter_g > th_{LR}$ then    ▷ If Global time-series have at least $th_{LR}$ subWindows with positive $\lambda$
        for all $loc \in CL$ do    ▷ check Local time-series of all Candidate Locations ($loc$)
            $counter_{loc} \leftarrow 0$
            for all $subWindow_i \in Window$ do
                $\lambda_{loc_t} \leftarrow \frac{\Sigma((x-\bar{x})(y-\bar{y}))}{\Sigma(x-\bar{x})^2\Sigma(y-\bar{y})^2}$
                if $\lambda_{loc_t} \geq 0$ then
                    $counter_{loc} \leftarrow counter_{loc} + 1$
            $corr_{g,loc} \leftarrow \frac{\Sigma_{t=t_1}^{t2}(Gts_{g_t} - \bar{Gts}_g)(Lts_{loc_t} - \bar{Lts}_{loc})}{\sqrt{\Sigma_{t=t_1}^{t2}(Gts_{g_t} - \bar{Gts}_g)^2\Sigma_{t=t_1}^{t2}(Lts_{loc_t} - \bar{Lts}_{loc})^2}} + 1$    ▷ Calculate the correlation between
                                                                    ▷ the time-series of the $loc$ and the global time-series
            if $counter_{loc} > th_{LR}$ then    ▷ Check if $loc$ exceeds the $th_{LR}$
                Assign to loc its correlation value, and True (for exceeding the $th_{LR}$ threshold)
            else
                Assign to loc its correlation value, and False (for not exceeding the $th_{LR}$ threshold)
    else
        Assign to all loc in $CL$ the values 0 (for the correlation) and False
return $CL$
```

A better idea is to use sliding windows, which we exploit in this work. In this case, a particular timeslot can be part of n-1 windows, where n is the length of the window. If a timeslot is at the beginning of an event (the latest in the window), the new timeslots to be inserted later are going to be more relevant. As a result, the timeslot is going to be in n-1 windows, the majority of which will be relevant.

Using the sliding window idea, we can now take advantage of the already extracted models of each location and incrementally update them for every slide, reducing dramatically the time needed for the contraction of the keyword vectors. In order to achieve this, we do not recalculate the concordance of each word for each location across the window. Instead, we extract the concordance across the window only for the first model created, and for every slide we update the concordances of each word by subtracting the concordance of the words in the data removed and adding those in the data added to our dataset. We can see the steps of the incremental update of the vectors in Algorithm 7.

Furthermore, due to the incremental update that we achieve at our concordance

$kwVectors$, we prove that our method can be applied in streaming manner. Unfortunately, the incremental update is not straight applicable on the Tf-Idf $kwVectors$ but still Tf-Idf methods get advantage on the incremental update of the concordances.

3.3.6 Dynamic Threshold Extraction

As we have already mentioned, the set up of this method allows us to identify non-geotagged tweets that are coming from the full stream of a social network. As a result, posts irrelevant to our candidate locations could still share stopwords, leading to a (small) similarity to some location. In order to filter out these cases, we use thresholds on the similarity, both for the CGLs and the FGLs.

The distribution of the keywords among the candidate locations is different depending on the time intervals we check. Therefore, the significant keywords are going to have different weights for each time interval. For example, during the night we have a few posts, leading to the creation of small dictionaries, where matching one of the stopwords in these dictionaries would lead to high similarity between the $Q{-}tweet$ and the candidate location. In this case, the threshold should be set high. This is not true when we consider the dictionaries created during the day.

In order to automatically set a dynamic threshold, we use a small training dataset (in our case 1 day), keeping the similarities between each $Q - tweet$ and the location that it corresponds to. We initiate our threshold by setting it to 0. Then, we identify the tweets that are correctly matched to a location, and we record their similarity. At the end, we compute the mean of all the similarities, giving us the threshold extracted from the first day and for the specific time intervals. In order to set up the threshold for these time intervals for the following day, we use the mean of the thresholds used in all previous days. As a result, the threshold for a given day and time interval is computed as the mean of the threshold means of the previous days for the same time interval.

Following the procedure described above, we dynamically update the threshold: the

Algorithm 7 Incremental Update of kwVector

```
procedure UPDATE OF KWVECTOR(all $kwVectors_t$,geotagged tweets from location $i$ for time intervals $t-1$ and $t+1$)
    for all $kwVector_{i_t} \in \{kwVectors_t\}$ do
        for all $word \in \{kwVector_{i_t}\}$ do
            $conc_{i_{t-1}} \leftarrow$ concordance in $i$ at $t-1$
            $conc_{i_{t+1}} \leftarrow$ concordance in $i$ at $t+1$
            $conc_i \leftarrow conc_{i_t} - conc_{i_{t-1}} + conc_{i_{t+1}}$
return $kwVectors_t$
```

thresholds are data driven, and the method is parameter-free.

3.3.7 Logistic Regression

After creating our models that use the Tf-Idf method in order to extract the most representative keywords of each location, we wanted to examine other types of methods that could probably help us to increase the number of the geolocalized posts. Due to this, we created two models that rely on the logistic regression model.

Although the merging of the geotagged tweets of each location into a single document seemed ideal in the case of the Tf-Idf, this was not the case when using the logistic regression model. This is due to the fact that for logistic regression, each single document (in our case each tweet) is a unique representative of the class (in our case each location). As a result, the number of the representatives is important and embedded in the logic of the algorithm.

Taking into consideration the logic of the method and the logic of our methods previously described, we created two models that are based on logistic regression:

1. the first model shares the logic that our previously described methods follow, merging the tweets of each location, creating a Single big Document per Location (SDpL),

2. the second model considers each tweet as a unique representative of a location (class), resulting into a model with Many Documents per Location (MDpL).

Having set up the documents of each location, we extract the vocabulary of our search space and we create a vector per location, representing the appearance of each word in the location. Finally, we train our model using these vectors and we classify the $Q - Tweet$ based on our model.

3.4 Experimental Evaluation

3.4.1 Evaluating the basic Algorithms

At the first part of our evaluation, we experimentally evaluate the four algorithms we described in Sections 3.3.1, 3.3.3, namely, TG, TG-TI, TG-C, and TG-TI-C.

Experimental Setup. We run the experiments at a machine that has OS Ubuntu 14.0.4 LTS, "4GB RAM" and processor "Intel Core i3 CPU M370 @2.40GHz x 4". For the implementation of our methods and the reimplementation of the QL, KL and GL we used Python 2.7.

Dataset. For the evaluation of our approach, we use a dataset containing English and Italian geotagged posts from Twitter, generated in Italy between June 20 and July 23,

2014. In particular, we have data from 6 of the largest Italian cities, namely, Rome, Milan, Naples, Bologna, Venice and Turin. The granularity of the neighborhood level we use is a square with side of 1000 meters. The time intervals we use have a duration of 4 hours (which can effectively capture an important event, as well as the start and the aftermath of this event), but we also keep detailed aggregated information for every 15min interval. The total number of tweets is 543.295 (219.681 originated from Rome, 137.622 from Milan, 60.065 from Naples, 49.434 from Bologna, 46.982 from Turin, and 29.511 from Venice).

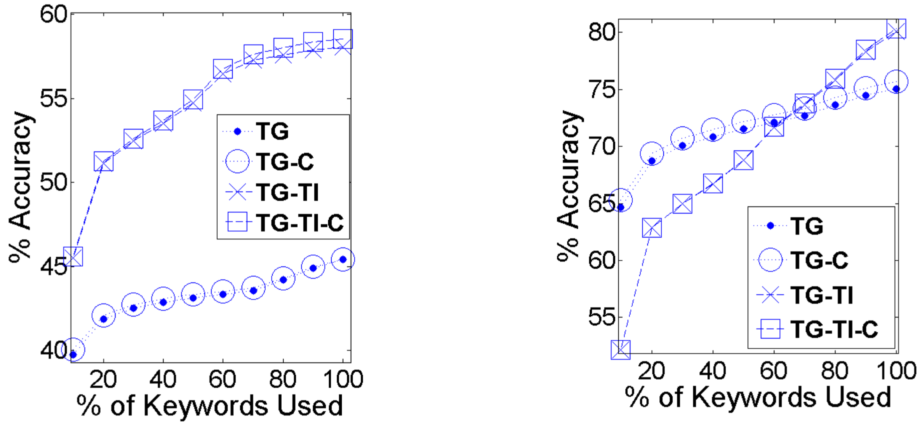
Algorithms. We experimentally evaluate the four one-level algorithms we described in Section 3.3, namely, TG, TG-TI, TG-C, and TG-TI-C, getting either the city or the neighborhood. As baselines, we implemented the QL and KL methods [37], which aim to solve a similar problem. We experimented with several values for the μ parameters used in these methods, and verified that $\mu = 10000$ gave the best results in our setting, as well. Furthermore, we implemented the GL method [16], considering each unique tweet as a unique user (which resulted in user profiles with only a few keywords).

Evaluation Measures. We study the time performance, as well as the effectiveness of each approach using the precision and recall measures: $Precicion = \frac{cgTweets}{gTweets}$ and $Recall = \frac{cgTweets}{aTweets}$, where $cTweets$ is the number of the correctly geolocalised tweets, $gTweets$ is the number of tweets we geolocalised, and $aTweets$ is the number of all tweets in the test set. In the cases where we predict the geolocation for all the tweets in the test set, the above precision and recall measures coincide, and we use the term *accuracy* instead. We also report the balanced F1 measure, $F1 = 2 * \frac{Precicion*Recall}{Precicion+Recall}$. Following previous work [37], we report the results when we consider the top-1 (@Top1), top-3 (@Top3), and top-5 (@Top5; only for neigborhood level) predicted geolocations, as well as the results when considering as correct the prediction of the exact geolocation (@0-Step), or of any geolocation at distance 1 (@1-Step; exact and its eight immediate neighbors), or 2 (@2-Step; exact and its 24 closest neighbors) from the exact. In all our experiments, we randomly divided the dataset in 80%training and 20% testing, repeated each experiment 30 times, and reported the mean values in the results.

City-Level Results

We start our analysis by running our method on city-level. We extract the English and Italian tweets from the 6 cities, removing the duplicated posts in order to avoid spam. We record the activity every 15 minutes, and we consider time intervals of 4 hour, leading to 181 timeslots (due to technical problems some of the timeslots were empty,and we do not consider those in our analysis).

In this case we extracted the similarities between the test tweets and the 6 cities, and



(a) @Top1

(b) @Top3

Figure 3.1: Accuracy for city level when using TG, TG-C, TG-TI, and TG-TI-C (@0-Step).

we also evaluated our approach using the correlation of the activity time series: we use the Pearson's correlation between the activity time-series of the 6 cities and the activity time series of Italy. The results (@Top1 and @0-Step) are presented in Figure 3.1a. As we can see in this plot, the accuracy for the city level is increased compared to the accuracy before the correlation. More precisely, we get the maximum number of matches in all four cases when we keep 100% of the keywords. The accuracy of TG and TG-C is almost identical, at 45%. For TG-TI we get 58% accuracy, while when using TG-TI-C we get 59% (though, our t-test analysis revealed that this difference is not statistically significant). After further analyzing the results of those two algorithms, we found that for 134 timeslots TG-TI-C has better accuracy, for 4 timeslots TG-TI and TG-TI-C have the same accuracy, and for the rest 43 timeslots TG-TI has better results. We note that the accuracy of an algorithm based on random choice was 17%.

After evaluating the algorithms using the most similar candidate geolocation (@Top1), we also evaluated them using the 3 most similar candidates (@Top3). As we can see in Figure 3.1b, when using only a small percentage of the keywords we get better results with the TG and TG-C algorithms. In contrast, when we use more than 70% of the keywords, the Tf-Idf based algorithms, TG-TI and TG-TI-C, result in better accuracy. The accuracy is increasing when the percentage of the keywords used increases.

Neighbourhood-Level Results

At this subsection we present the evaluation we did for our approach at the neighborhood level. Every time we run the algorithm, we get the similarity both before and after achieving correlation between the total number of tweets from Milan and the number of tweets from every square.

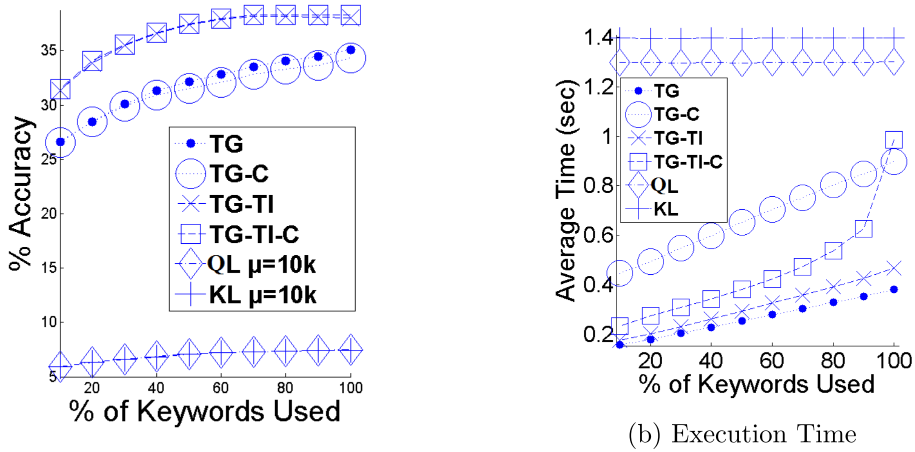
In Figure 3.2a, we present the mean accuracy that our algorithms have among all timeslots, depending on the percentage of the keywords used while taking into consideration only the first answer. After analyzing the results, we come to the conclusion that the best mean accuracy is 38% and is achieved by using 80% of the keywords and when using the TG-TI-C algorithm. The second best algorithm is TG-TI. In this case, the best accuracy is achieved when using 80% of the keywords and is almost 38%. The best accuracy achieved by TG is 35%, while TG-C reached 34% (both achieved when using 100% of the keywords). In order to make fair the random choice, we were not choosing between all the 400 squares but only between those which had data at the train datasets. The mean accuracy of the random algorithm was 2%.

The maximum accuracy we got for one timeslot is 74% and we got it using the TG-TI algorithm. Nevertheless, the best results tend to be when we use TG-TI-C. As happens at city level, the accuracy tends to get increased after the use of the correlation between city and square activity when using Tf-Idf, but on the contrary to the city level, it gets decreased when adding the correlation parameter to the method that uses as weights the raw number of the appearances of each word. This may be caused due to the fact that people post for the same topic even if they are at neighboring squares. For example, during the concert identified at city-level, we do not have great accuracy because people were tweeting even on their way there, causing neighboring squares to have the same topic.

After evaluating our algorithms, we compared them to the QL and KL baselines, using the same spatial and temporal granularities as those we used for our algorithms. These results are depicted in Figure 3.2a. Surprisingly, we found that our results are up to 31% better. This is due to the spatial and temporal setting-up that we use. The authors of [37] originally use much bigger spatial granularity, while the temporal granularity of the two datasets that they use is 4 weeks and 3 months respectively. Moreover, probably due to our granularity, the results between QL and KL are almost the same.

Furthermore, we run experiments with the GL method, whose accuracy was in the best case around 4-5%. We believe that this is due to the differences in the problem definition and focus of this method, which is geared towards spatio-temporal granularities that are much larger than the ones we consider in our work.

In order to check the trade-offs when using a percentage of the keywords and to compare the execution times needed for our algorithms and the state-of-the-art used, we measured the mean execution time needed per timeslot for training the models and answering the query-tweets. The results are depicted in Figure 3.2b. As we can see in the graph, the best time is achieved by TG. The reason is that it does not spend time for



(a) Accuracy

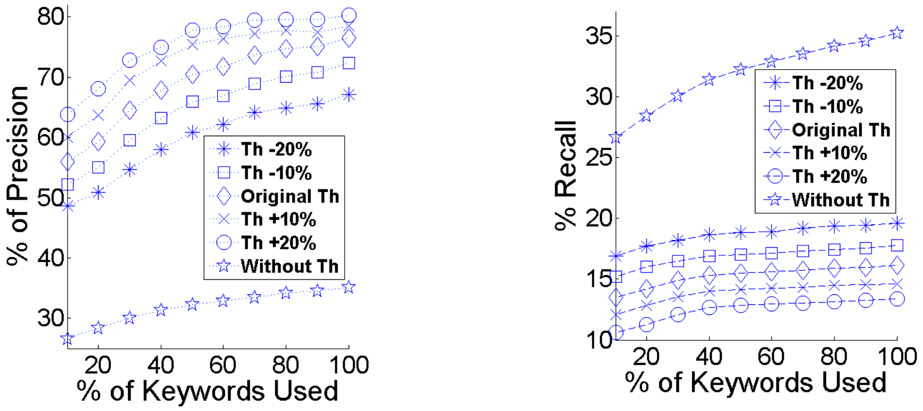
(b) Execution Time

Figure 3.2: Trade-off Between Execution Time and Accuracy for Neighbourhood Level (@Top1 and @0-Step).

calculating neither the new keyword weights, nor the correlations. The worst execution time of our 4 methods is achieved by the method TG-C. This is due to the fact that although we prune the keyword-space, we do not remove stopwords or common words that appear in many squares. As a result, the new similarities have to be recalculated for all those candidate squares that have the common or stopwords, while by using Tf-Idf we eliminate many of the candidates.

After having evaluated our methods and compared them with the state-of-the-art when answering to all the queries, we evaluated our methods when using a dynamically defined similarity threshold. The threshold we have chosen to use are automatically calculated by the results of the 4-hour timeslots of the previous days. As a result, we have 6 user-free dynamic thresholds that are calculated by taking the mean of the mean similarities of the previous respective timeslots. By introducing the thresholds in our methods, we answer to less query-tweets, reducing the recall but increasing the precision up to 100%. In Figure 3.3, we present the precisions and the recalls after the introduction of the thresholds for the method TG-C, while in Figure 3.4 we present the precision and recall for TG-TI-C. We run experiments by using no threshold, the exact dynamic threshold, and the exact threshold +-10% and +-20%. Furthermore, in order to evaluate the results, we use the balanced F1-measure. The results of the F1 measure for the two methods presented before is depicted in Figure 3.5.

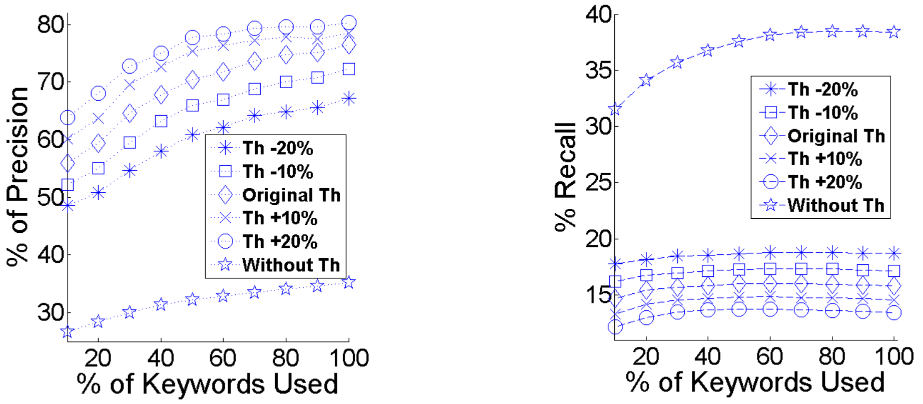
After evaluating our methods using the first most similar answer, we analyzed the results when taking under consideration the first 3 and the first 5 most similar candidates. In Figures 3.6a and 3.6b, we can see depicted the mean accuracies when using 10-100% of



(a) Precision

(b) Recall

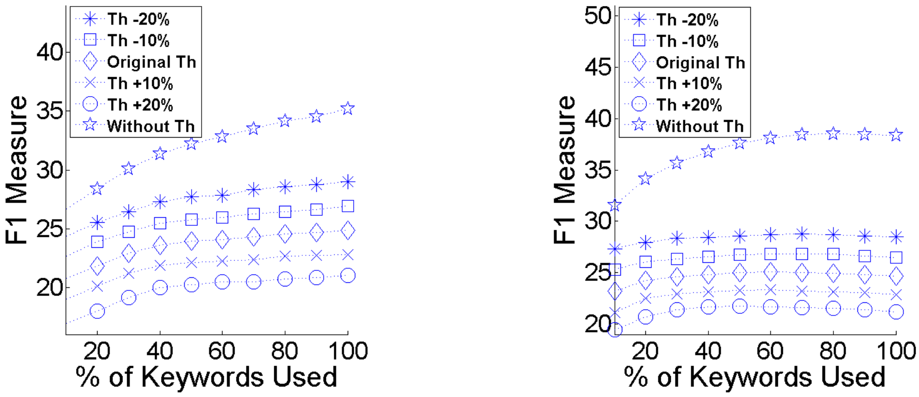
Figure 3.3: Precision and recall on Neighbourhood Level for TG when using dynamic thresholds (Th) (@Top1 and @0-Step).



(a) Precision

(b) Recall

Figure 3.4: Precision and recall on Neighbourhood Level for TG-TI-C when using dynamic thresholds (Th) (@Top1 and @0-Step).



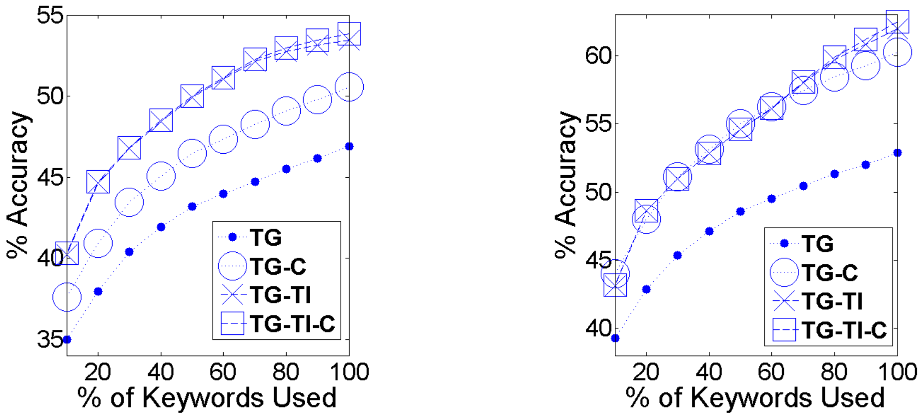
(a) TG

(b) TG-TI-C

Figure 3.5: F1 measure for Neighbourhood Level without and with threshold (Th) (@Top1 and @0-Step).

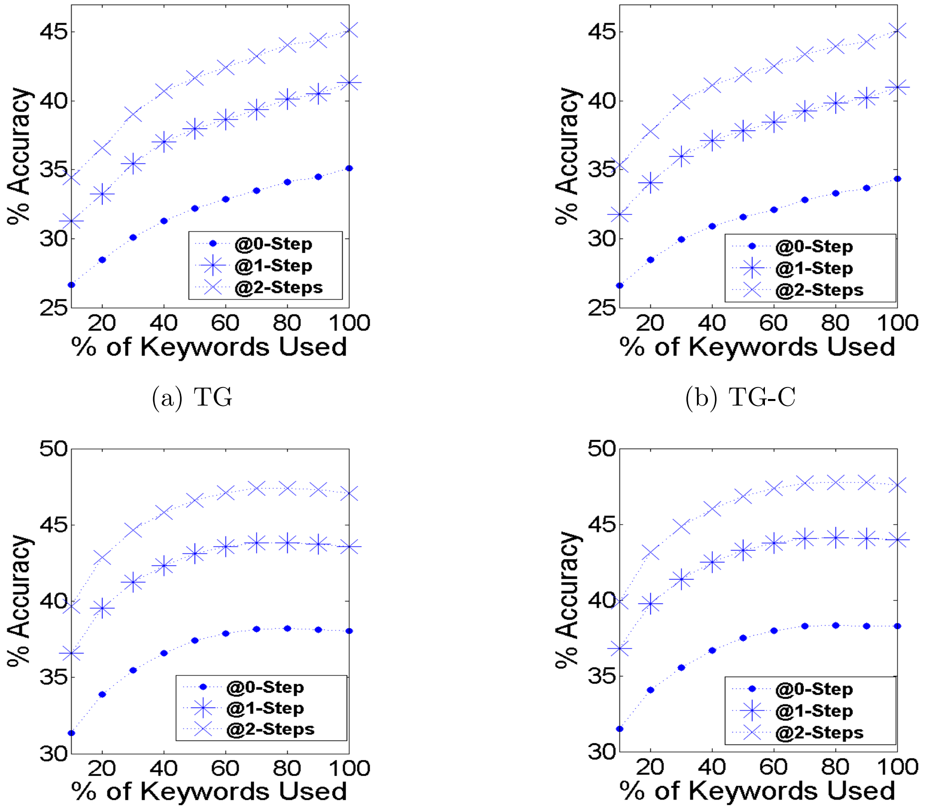
the keywords for both cases. On the contrary, at the city-level analysis, the TG-TI and TG-TI-C algorithms are always better when compared to TG.

Finally, we study the performance of our methods in the case where we consider the $1 - Step$ and $2 - Steps$ squares neighboring to the exact answer, as correct answers as well. The results of this evaluation are depicted in Figure 3.7. When using the $1 - Step$ evaluation we have up to 7% difference for the accuracy achieved by using the TG-C compared to the same method when using the exact answer case. Though the difference between the exact answer and the $1 - Step$ is so big, the difference between the same methods when using $2 - Steps$ is only up to 4% better. Probably this difference is due to the fact that neighboring squares share the same topic while the topic differs more comparing to the neighbors of the neighbors. Furthermore, when using up to 30% of the keywords, TG-TI and TG-TI-C for the exact answer have better accuracy compared to those that TG and TG-C have for the $1 - Step$. The percentage of the keywords for which TG-TI and TG-TI-C of $2 - Steps$ have better accuracy compared to the TG and TG-C of the $1 - Step$ have, is even bigger coming up to 80% of the keywords, a fact that becomes even more interesting when taking into consideration the complexity of the methods when we have $n - Step$ and $(n - 1) - Step$ evaluations. After analyzing the results in detail, we identified that in all the cases the best mean accuracy appears for TG-TI-C, in the cases of the exact answer or the $1 - Step$ match when using 80% of the keywords, while for the case $2 - Step$ we get the best accuracy when using 90% of the keywords. The second best mean accuracy in the one achieved with TG-TI, while the third best method is TG.



(a) @Top3 (b) @Top5

Figure 3.6: Accuracy for Neighbourhood Level for TG, TG-C, TG-TI, and TG-TI-C (@0-Step).



(a) TG (b) TG-C

(c) TG-TI (d) TG-TI-C

Figure 3.7: Accuracy for Neighbourhood Level (@Top1).

3.4.2 Evaluating the Algorithms with Linear Regression

Experimental Setup. We performed the experiments on a server running on Ubuntu 14.04.2 LTS, with 64GB RAM, and an Intel(R) Xeon(R) CPU E5506 @ 2.13GHz processor. For the implementation of our methods and the reimplementation of the QL and KL we used Python 2.7.

Datasets. For the evaluation of our approach, we use 3 datasets containing geotagged[1] posts from Twitter, generated in Italy, Germany and the Netherlands. In particular, we have data from 6 of the largest Italian cities, namely, Rome, Milan, Naples, Bologna, Venice and Turin, and from the capital of Germany, Berlin, and the capital of Netherlands, Amsterdam. The tweets from Italy were generated between June 20 and July 23, 2014, while the tweets from Germany and the Netherlands were generated between August 10 and September 11, 2014. The granularity of the neighborhood level we use for every city is a square with side of 1000 meters. The number of tweets is 543.295 for Italy (219.681 originated from Rome, 137.622 from Milan, 60.065 from Naples, 49.434 from Bologna, 46.982 from Turin, and 29.511 from Venice), 77.179 for Berlin and 136.189 for Amsterdam. The time windows we use have a duration of 4 hours (which can effectively capture an important event, as well as the start and the aftermath of this event), while also keeping the detailed aggregated information for every 15min time interval. As mentioned in Section 3.3.5, we use the sliding window model. We experimented sliding the window by 1 and by 2 time intervals, getting almost the same results; thus, we chose to slide our window by 2 time intervals per slide (30-minutes), which led to faster execution times. Finally, the default grid we use in this study is 20 by 20 squares.

Algorithms. We experimentally evaluate the six one-level algorithms we described in Section 3.3, namely, TG, TG-TI, TG-C, TG-TI-C, TG-CLR and TG-TI-CLR (the last two only for the neighborhood level). As baselines, we implemented the QL and KL methods [37], which aim to solve a similar problem. In order to choose the value for the μ parameter, we followed the same methodology as in the original paper [37]: we experimented with several values for the μ parameter, in the range [100,10000]), and verified that $\mu = 10000$ gave the best results in our setting, as well.

Evaluation Measures. For the evaluation of our (new and old) methods on our (new and old) datasets, we use the evaluation described in Section 3.4.1, studying the time performance, as well as the effectiveness of each approach using the precision and recall measures:

[1] Earlier studies have shown that techniques and models built for geotagged data indeed generalize to non-geotagged data, since geotagged and non-geotagged tweets have similar data characteristics [31].

Neighbourhood-Level Results

In this subsection, we present the results for the neighborhood level evaluation, for which we used data from four different European cities: Milan, Rome, Berlin and Amsterdam. As we have already mentioned, we created a grid of 400 squares (20 by 20) for each city. For the city of Rome, we additionally ran some experiments using a grid of 900 squares (30 by 30).

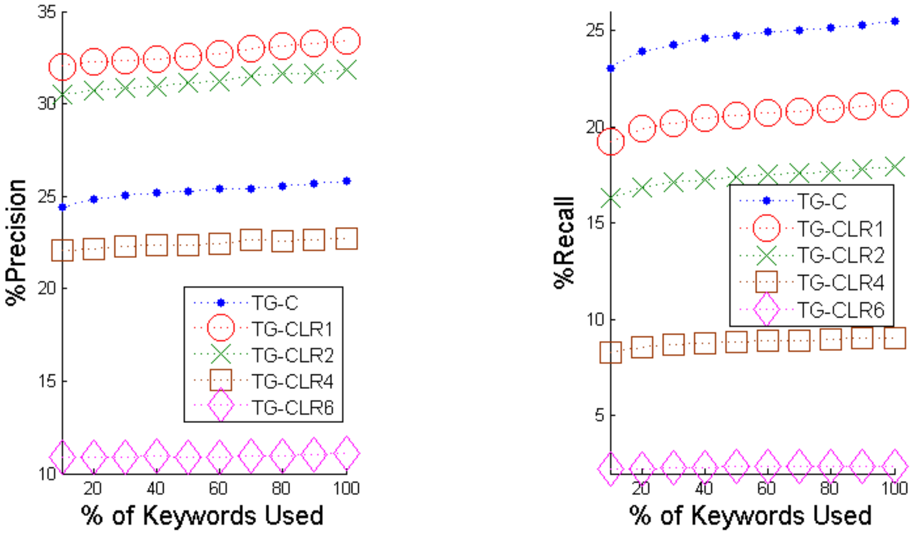
Setting the Parameters

We first identify the best threshold to use for the LR parameter. As we previously mentioned, we use a window of 4 hours (16 15-min timeslots), and sub-windows of size $n_{sub-window} = n_{window}/2 = 8$. Furthermore, the maximum LR equals to the number of slides, which is 8, as well. We experimented by setting the LR-threshold equal to $\{1, 2, 4, 6\}$, and depict the results in Figures 3.8 (precision and recall for algorithms not using Tf-Idf), 3.9 (precision and recall for algorithms using Tf-Idf), and 3.10 (F1 measure for all algorithms). For brevity, we only report the results for the city of Milan; results for the other cities are similar.

In this experiment, we had 3264 15-min timeslots, resulting into 1624 window slides. For each method, we extracted the mean precision, recall and F1 scores among all windows, while varying the percentage of the keywords used. We observe that the best mean precision is 48%, which is achieved by TG-TI-CLR1 when using 100% of the keywords (Figure 3.9(a)), while the maximum recall for this method is 32%, when using 40% of the keywords (Figure 3.9(b)). Note that the same method without the use of the trends, that is, TG-TI-C, has maximum precision and recall 40% and 39%, respectively. Regarding the TG-CLR1 algorithm, we get maximum precision 33% and maximum recall 21%, both when using 100% of the keywords. Due to these, and after finding out that the F1 score of the $CLR1$ methods isn't too different compared to those not using linear regression, we concluded that for the rest of the experimental part we are going to use only the $CLR1$ methods.

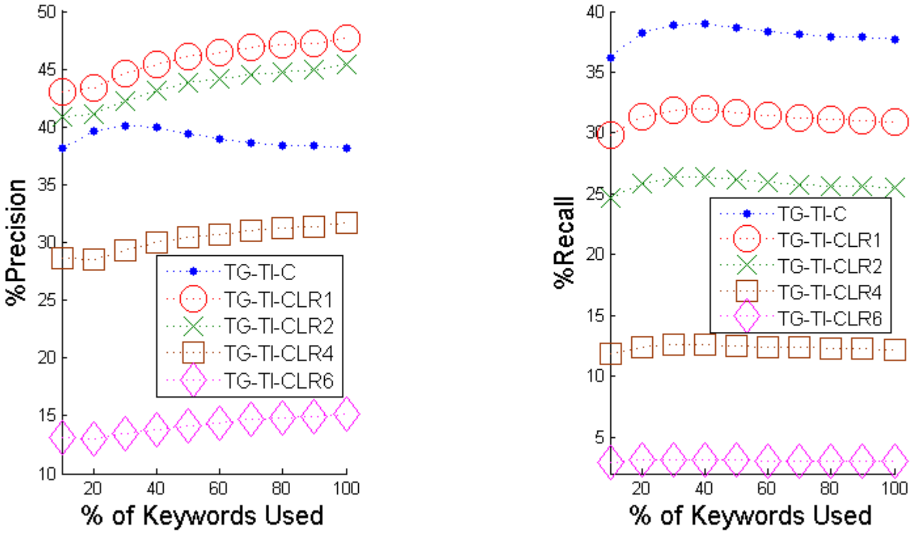
Evaluating the Correlation-Based Methods

In the following experiments, we compare the CLR methods to those that do not use correlation. In Figure 3.11, we present the mean precision and recall that our algorithms have for the city of Milan among all windows, when varying the percentage of the keywords used. As before, we only consider the first answer given by each algorithm (i.e., @Top1). The best precision is 48% and is achieved by TG-TI-CLR1 using 100% of the keywords. The maximum recall is 38% achieved by TG-TI when using 30% of the keywords. According to the F1 measure, TG-TI achieves its best using 30% of the keywords, with F1 equal to 39%. TG-TI-CLR1 achieves best F1 score 37% when using 50% of the keywords. The second best precision is 39%, achieved by TG-TI when using 30%. The



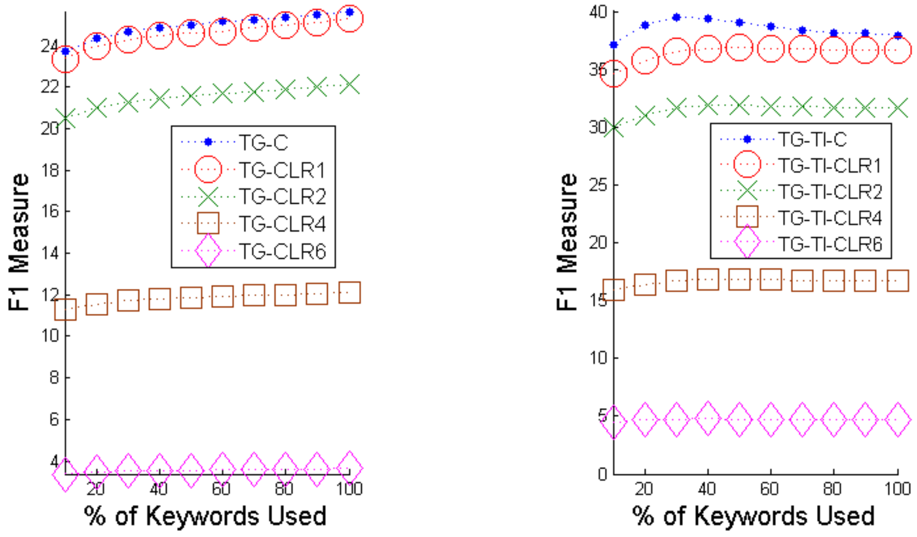
Precision Recall

Figure 3.8: TG-CLR for Different LR Parameters (@Top1 and @0-Step).



Precision Recall

Figure 3.9: TG-TI-CLR for Different LR Parameters (@Top1 and @0-Step).



TG-CLR TG-TI-CLR

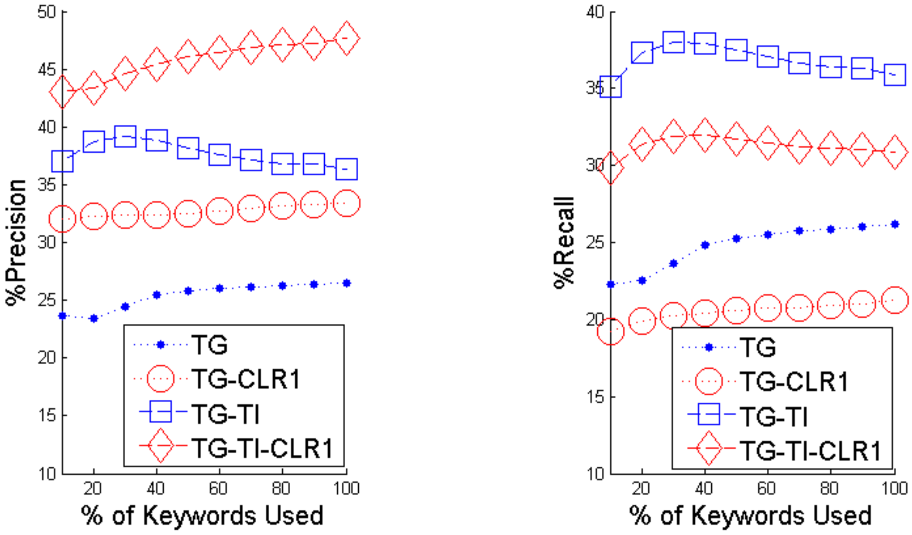
Figure 3.10: F1 for TG-CLR and TG-TI-CLR for Different LR Parameters (@Top1 and @0-Step).

best precision achieved by TG is 27%, while its best recall is 26%. TG-CLR1 reached up to 33% precision and 21% recall (both achieved when using 100% of the keywords). The mean accuracy of the random algorithm, which was choosing one square at random only among that had data at the train datasets, was less than 2%.

We note that the best precision is always observed when we use the TG-TI-CLR1 algorithm. This means that the correlation between the city and square activities is beneficial, when using the linear regression parameter that prunes activities with negative trends. As a result, we do not estimate the location of tweets that would probably be wrongly predicted, leading to a small penalty in recall, but increased precision.

We now report the results of the same experiment for the cities of Rome (in Figure 3.12), Berlin (in Figure 3.13), and Amsterdam (in Figure 3.14). The best precision we observed for the city of Rome was 48% and was achieved by TG-TI-CLR1, using 100% of keywords, while the best recall was achieved when using TG-TI method, using 40% of keywords. The same methods also resulted in the highest precision and recall for Berlin. In particular, TG-TI-CLR1 achieved a precision of 58%, for a recall of 40%. The best recall for Berlin was 47%, achieved by TG-TI, which also led to the second best precision, 51%. Regarding the city of Amsterdam, we achieve the highest precision of 44% with TG-TI-CLR1, while the best recall of 38% is achieved by TG-TI.

The results show that the behavior of the algorithms is similar across cities, while their relative performance remains the same. An interesting observation is the fact that



Precision

Recall

Figure 3.11: Trade-off Between Precision and Recall for Neighbourhood Level (Milan, @Top1 and @0-Step).

the precision and recall for Berlin are much higher than the rest of the cities. This is due to the distribution of the keywords among the squares, which resulted into more representative keyword sets for each square.

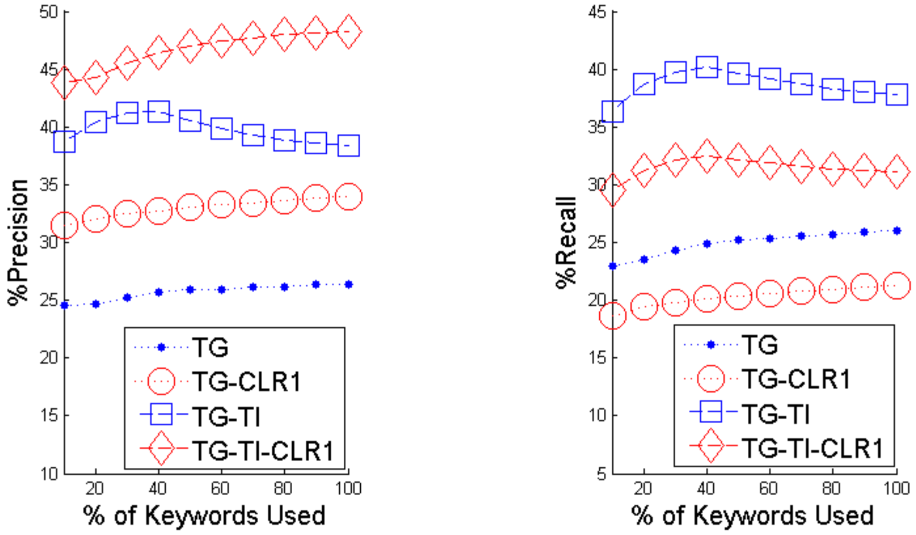
Comparing to Baselines

In this set of experiments, we compare our approach to the QL and KL baseline algorithms. We use the same spatial and temporal granularities for all algorithms. Similarly to our methods, we only consider tweets for which there exists at least one candidate location with similarity greater than 0. The results of this comparison are illustrated in Figure 3.15(a).

We observe that TG-TI-CLR1 achieves up to 18% better recall than the QL algorithm[2], and up to 22% better F1 score. This difference in performance can be explained by the different focus of the QL algorithm, which was developed to operate at much bigger spatial (in the order of zipcodes, or cities) and temporal granularities (in the order of weeks, or months) [37]. We also note that (for the same reasons) the results between QL and KL are almost the same. Therefore, in our plots we only report the F1 score for QL.

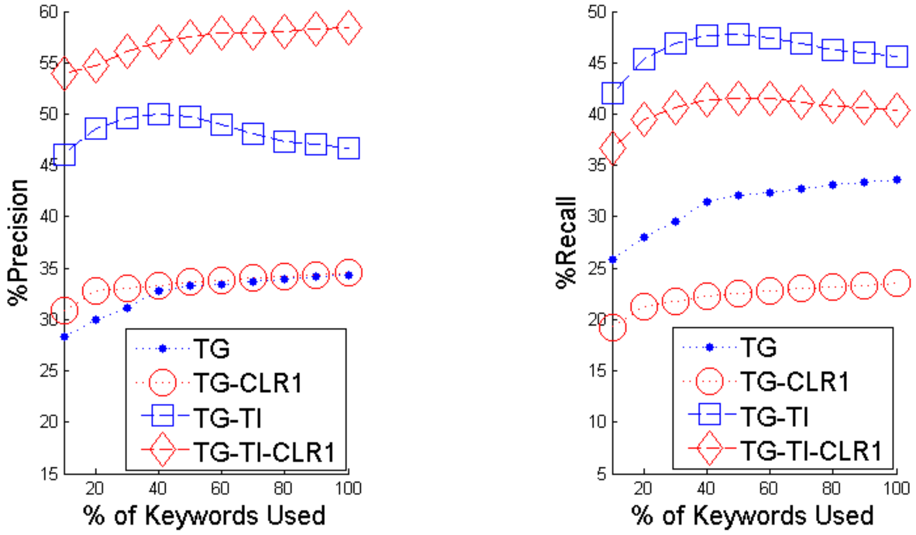
In terms of time performance, we measured the mean execution time needed per 4-hour window for the entire process: training the models, and extracting the similarities between

[2] We note that the QL results reported here are much better than those reported earlier. This is due to the different experimental setup (i.e., sliding windows) that we now use for all algorithms, which resulted in an increased number of windows with a high number of tweets, leading to higher execution times and better models.



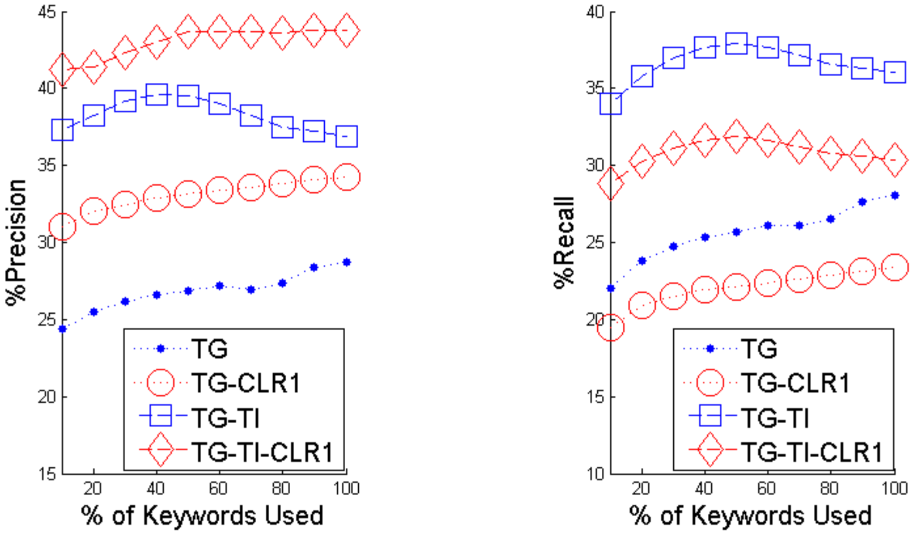
Precision Recall

Figure 3.12: Precision and Recall for the City of Rome (@Top1 and @0-Step).



Precision Recall

Figure 3.13: Precision and Recall for the City of Berlin (@Top1 and @0-Step).



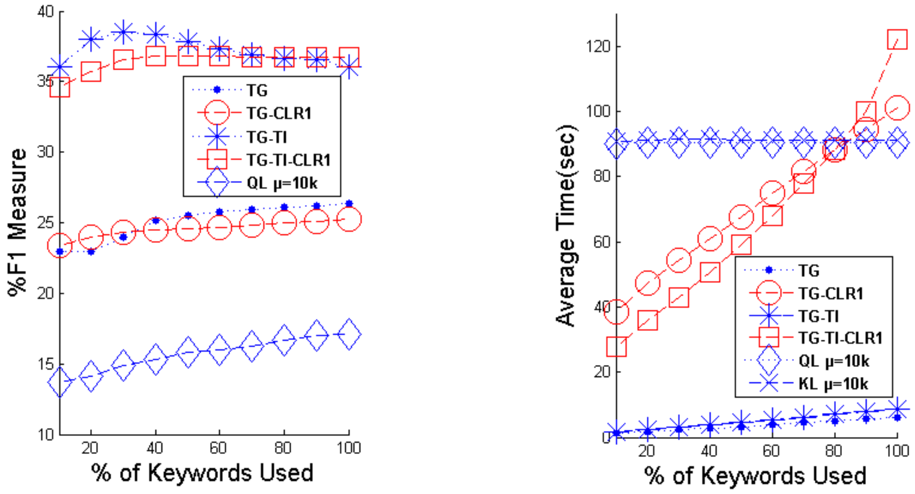
Precision Recall

Figure 3.14: Precision and Recall for the City of Amsterdam (@Top1 and @0-Step).

the query-tweets and the candidate locations. Figure 3.15(b) depicts the execution time needed for each algorithm.

As we can see in the graph, TG is the fastest algorithm. This is natural, since this algorithm does not spend time calculating the Tf-Idf, the correlations, or the linear regressions. The QL algorithm has a consistently high execution time of around 90sec, independent of the number of keywords considered. TG-TI-CLR1 performs in the middle. The interesting point is that although this algorithm has to calculate the Tf-Idf, the correlations and the linear regressions, the total time needed for each square, when using 10-70% of the keywords is smaller than the time needed for TG-CLR1. The reason is the search space pruning. When compared to TG-CLR1, the TG-TI-CLR1 algorithm prunes stopwords, and thus, eliminates the candidate locations that do not share any keyword with the tweet under examination. We also observe that when TG-TI-CLR1 achieves its best F1 score, i.e., when using 50% of the keywords, it is significantly faster than the QL algorithm.

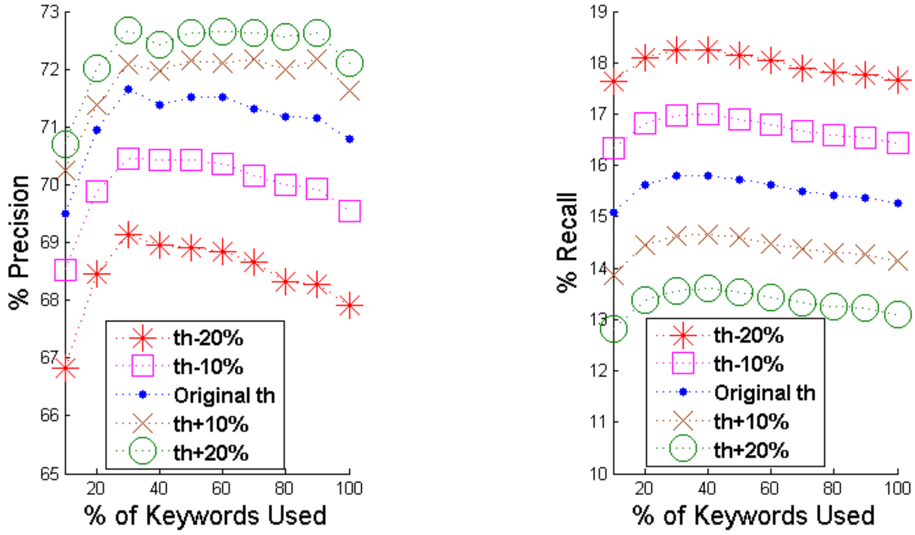
Finally, we note that the KL algorithm performs very similar to QL, but requiring at all cases a bit higher time when compared to QL (around 0.8 secs more).



F1 Score Comparison

Execution Time

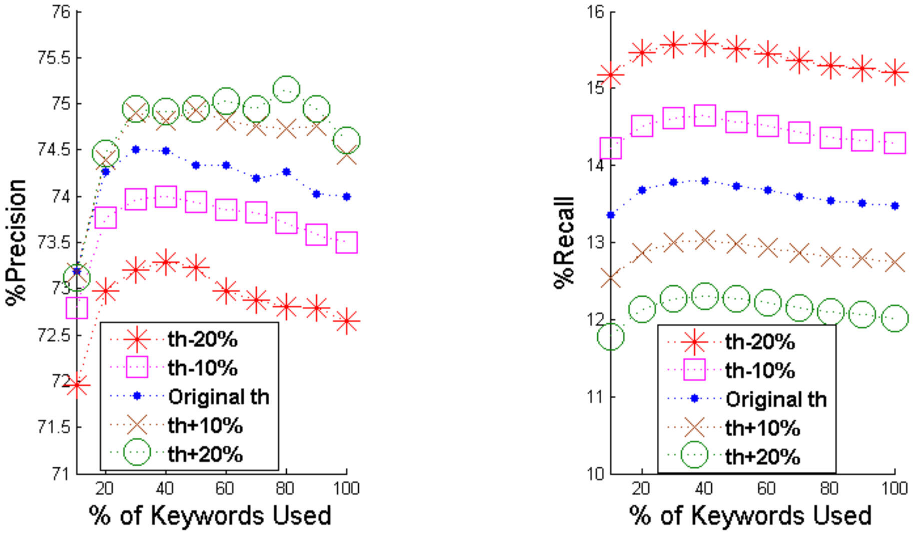
Figure 3.15: Trade-off Between F1 Score and Execution-Time for the City of Milan (@Top1 and @0-Step).



Precision

Recall

Figure 3.16: Precision and Recall on Neighbourhood Level for TG-TI when using dynamic thresholds (Th) (@Top1 and @0-Step).



Precision Recall

Figure 3.17: Precision and Recall on Neighbourhood Level for TG-TI-CLR1 when using dynamic thresholds (Th) (@Top1 and @0-Step).

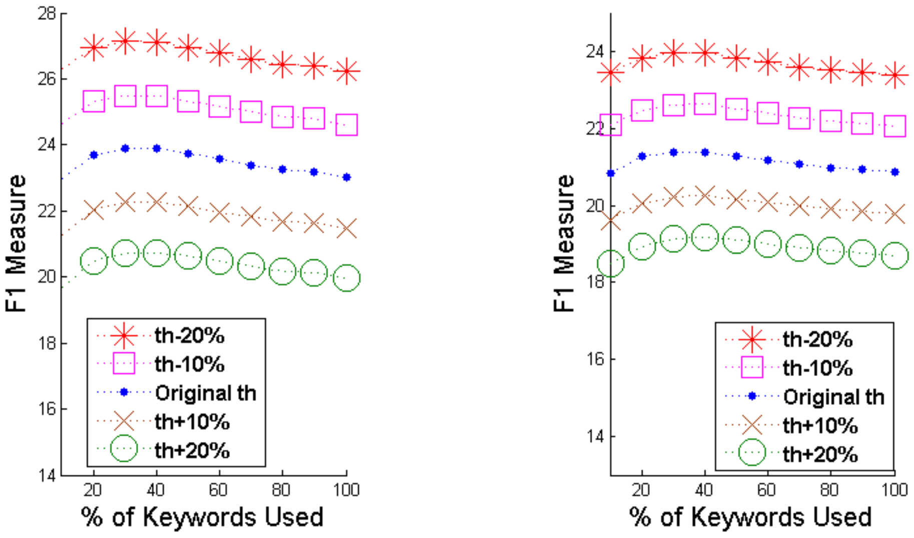
Focusing on Precision

We now examine the behavior of our algorithms when we want to achieve high precision, which is useful for several applications.

In the first set of experiments, we employ a dynamic similarity threshold that determines whether the algorithm will make a prediction for the geolocation. The thresholds we use are automatically set, based on the results of the same timeslots of the previous days: they are computed as the mean of the similarities of the correctly identified geolocations, averaged over the corresponding timeslots of the previous days. We have 48 (dynamic) thresholds, one per half-hour slide. Evidently, these thresholds lead to fewer predictions of tweet geolocations, reducing the recall, but increasing the precision.

In Figure 3.16, we present the precisions and the recalls after the introduction of the thresholds for the method TG-TI, while in Figure 3.17 we present the precision and recall for TG-TI-CLR1. We run experiments by using the exact dynamic threshold, the exact threshold +-10% and the exact threshold +-20%. Furthermore, in order to evaluate the results, we use again the balanced F1 score. The F1 score for the two methods presented before is depicted in Figure 3.18.

After evaluating our methods using the first most similar answer, we analyzed the results when taking under consideration the first 3 (Top3) and the first 5 (Top5) most similar candidates. In Figures 3.19 and 3.20, we can see depicted the mean precisions and recalls when using 10-100% of the keywords for both cases. The results show that



TG-TI TG-TI-CLR1

Figure 3.18: F1 measure for Neighbourhood Level with threshold (Th) (@Top1 and @0-Step).

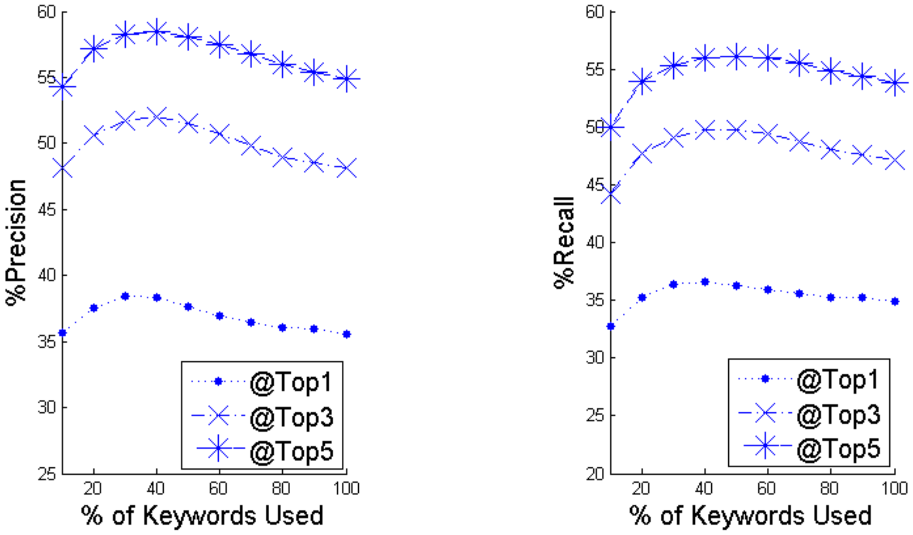
both precision and recall are benefiting, with the F1 scores increasing from around 35% to around 55% (Figure 3.21).

Finally, we study the performance of our methods in the case where we relax the definition of the correct answer to include answers that are 1 square ($1 - Step$), or 2 squares ($2 - Steps$) away from the exact answer. That is, we consider the near neighbors of the exact answer to be correct answers, as well. The results of this evaluation are depicted in Figures 3.22 and 3.23 (we report the results for the city of Milan).

When using the $1 - Step$ evaluation, we observe an increase of up to 6% for precision, and up to 4% for recall. The additional benefit for $2 - Steps$ is diminishing, exhibiting an increase of up to 4% for precision and up to 2% for recall. This effect of diminishing returns is due to the fact that immediately neighboring squares tend to share the same topic, while the topic dilutes and differs more when we move further away. In all cases, TG-TI-CLR1 accounts for the best mean precision. The second best precision in the one achieved by TG-TI, while the third best is achieved by TG-CLR1.

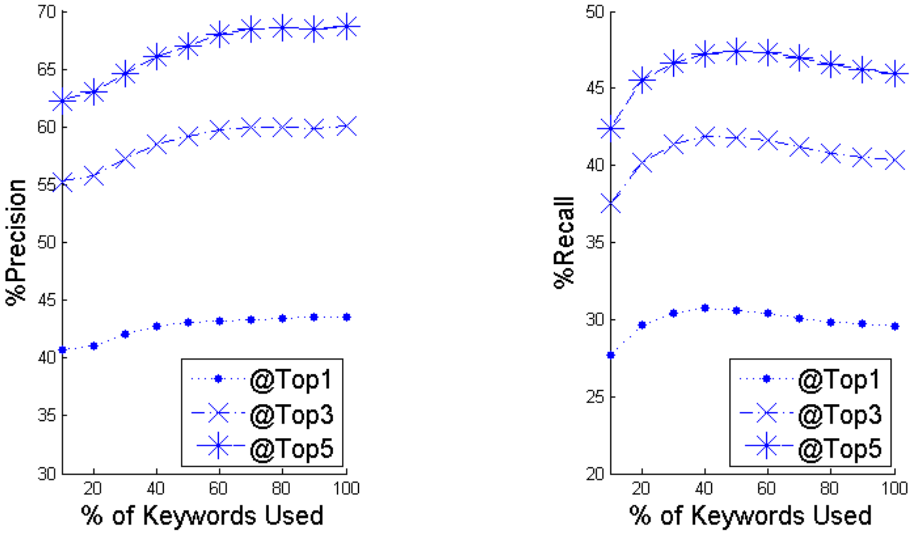
Finally, we run experiments modifying at the same time all the three parameters presented before, namely the similarity threshold, the @Step and the TopK. In Figure 3.24, we illustrate the precision and recall of the TG-TI-CLR1 method. The results show that we can achieve a significant increase in precision, but only a modest increase in recall. We note that precision hovers above the 75%, therefore, making the proposed approach attractive for applications that need access to the geolocations of tweets.

Size of Search Space



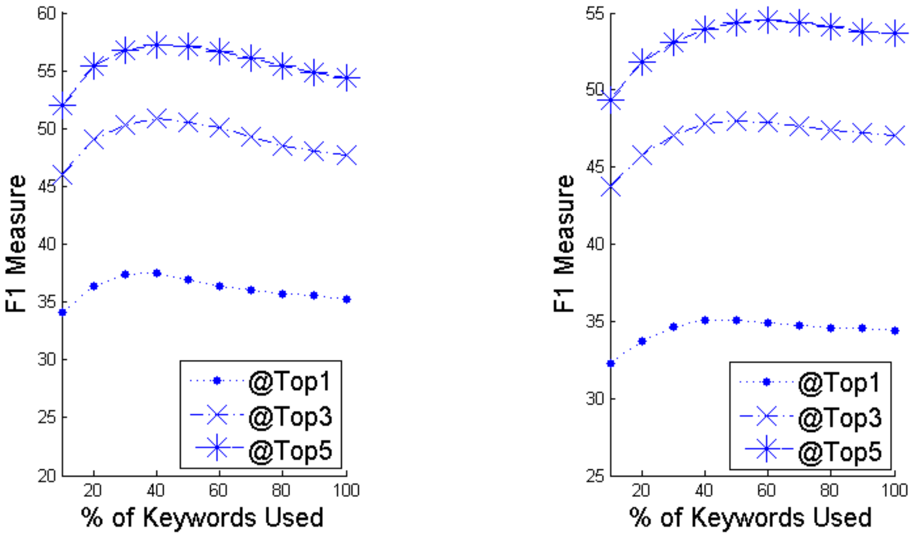
Precision Recall

Figure 3.19: Precision and Recall for TG-TI (@0-Step).



Precision Recall

Figure 3.20: Precision and Recall for TG-TI-CLR1 (@0-Step).



TG-TI TG-TI-CLR1

Figure 3.21: F1 Score for TG-TI and TG-TI-CLR1 (@0-Step).

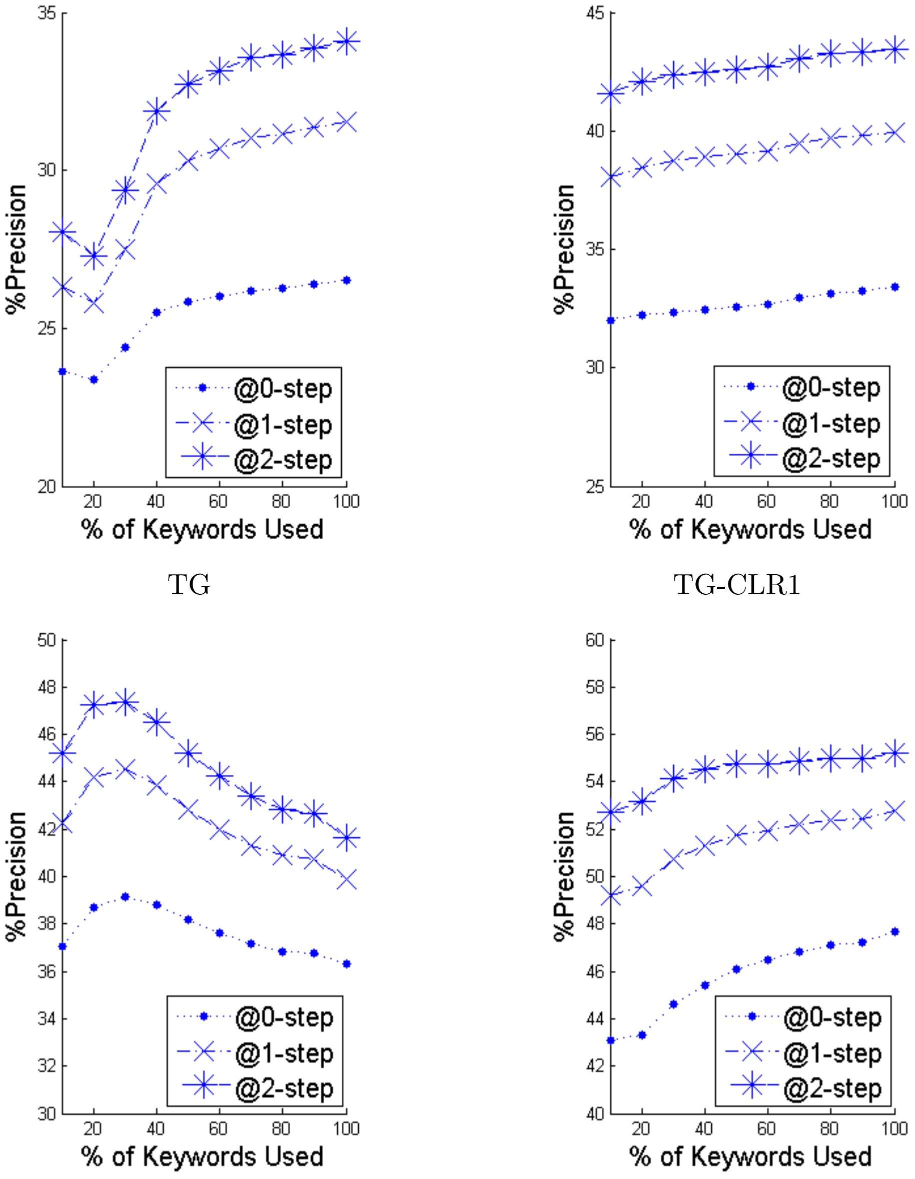
In order to evaluate our method with larger search spaces, we created a bigger grid for the city of Rome, and we ran experiments on this new dataset. In particular, we created a grid of 900 squares (30 by 30), while keeping the rest of the setup parameters the same. The size of each square is the same as before: 1km. In Figure 3.25, we compare the precision and recall of the Tf-Idf methods for the 20 by 20 and the 30 by 30 grids.

The best precision for the 30 by 30 grid is 45% and is achieved by TG-TI-CLR1 when using 100% of keywords, while the best recall is 37% and achieved by TG-TI when using 40% of the keywords. As expected due to the higher search space, the precision and recall achieved by each method are lower than those for the smaller grid: they were up to 4% lower for both algorithms, when the search space increased by 225%. These results demonstrate that the effect of the increase of the search space on the proposed algorithms is relatively small.

Two-Step TG-TI-CLR Performance with Varying Number of CGLs

In this set of experiments, we study the performance of our two-step TG-TI-CLR method, identifying first the *CGL* and afterwards combining it with the *FGL*. The number of *CGL*s (i.e., cities) varies between 1 and 7. As estimated location, we only consider the first answer given by our algorithm (i.e., @Top1). We note that the random algorithm had precision less than 0.024% and recall less than 0.12%, with the highest values occurring when using 1 city.

In Figures 3.26a-3.26b, we illustrate the precision and recall when we use 7 *CGL*s. The results for 1-6 *CGL*s are very similar, and omitted for brevity. The F1 for the cases of 1



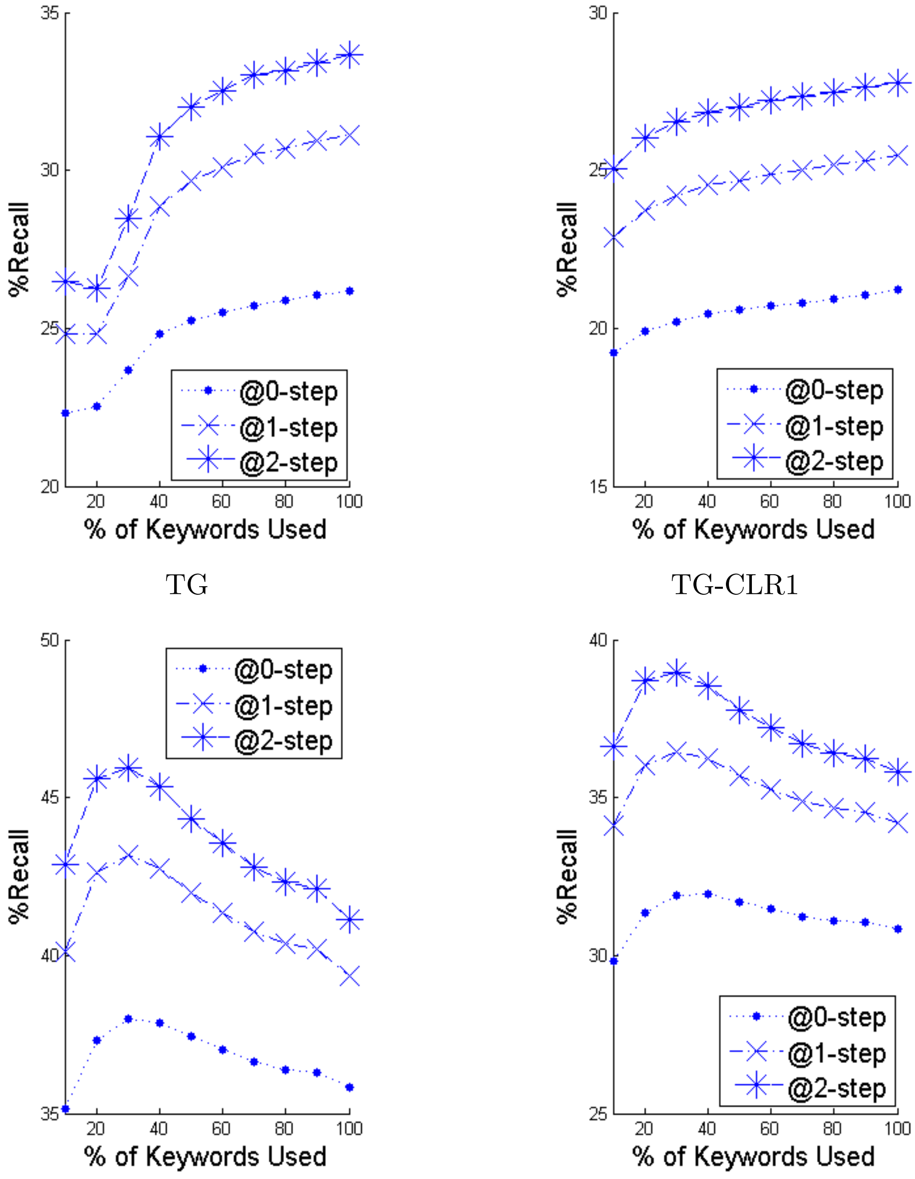
TG

TG-CLR1

TG-TI

TG-TI-CLR1

Figure 3.22: Precision for Neighbourhood Level (@Top1).



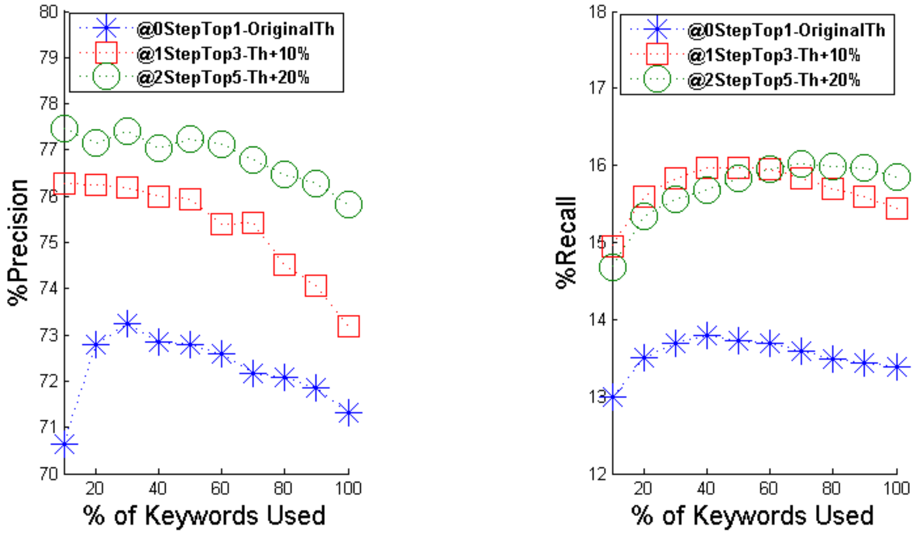
TG

TG-CLR1

TG-TI

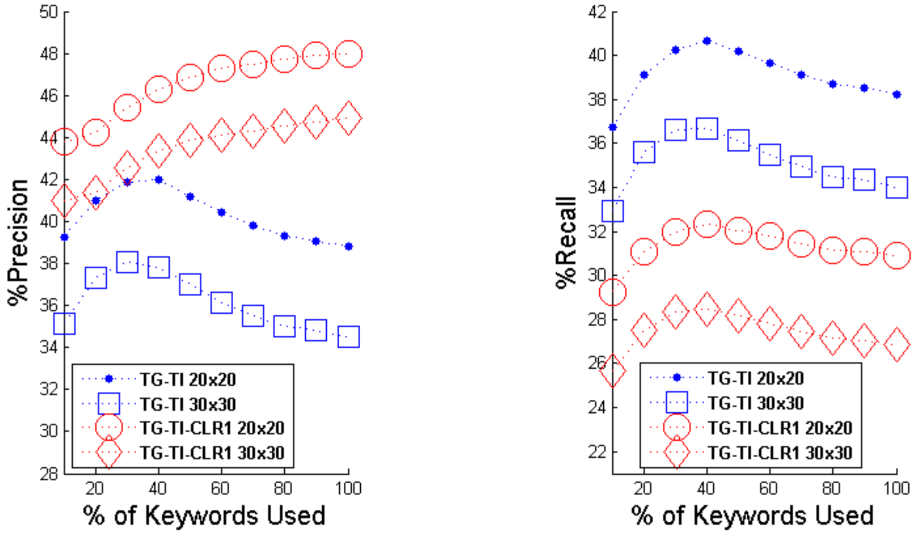
TG-TI-CLR1

Figure 3.23: Recall for Neighbourhood Level (@Top1).



Precision Recall

Figure 3.24: Precision and Recall for TG-TI-CLR1 for varying similarity threshold, TopK, and @Step.



Precision Recall

Figure 3.25: Precision and Recall Comparison for the City of Rome (grids 20x20 and 30x30, @Top1 and @0-Step).

and 7 *CGL*s are compared in Figures 3.26c and 3.26d, respectively. Using our approach, we achieve a precision of up to 89%, and a recall of up to 17%, while the best F1 was 26%. The best precision is achieved when using 60% of the keywords and a threshold of +20%, while the best recall and F1 for 70% of the keywords and "no threshold".

For the comparison to the state-of-the-art presented in Figure 3.27, we use the version of our method with threshold +20%. As depicted in the plots, our method achieves up to 80% precision and 23% recall, while KL only achieves up to 13% precision and 20% recall.

We note that as we increase the number of *CGL*s considered, we would expect to see a reduction in the precision and recall values, as a result of the increased search space. When looking at all the detailed results though, we do not observe this. On the contrary, precision slightly increases as we add *CGL*s, demonstrating the robustness of our approach.

Performance for Targeted Locations

We now evaluate the performance of the proposed approach for targeted locations of interest. The results for the Vatican and San Siro locations are presented in Figures 3.28a-3.28b, and Figures 3.28c-3.28c, respectively.

The precision for Vatican reaches a maximum of 68% when using either 10%, or 20% of the keywords and "no threshold", while recall reaches 84% when using 100% of the keywords and "no threshold". Similarly, San Siro achieves a precision of 49%, and a recall of 54%, for 10%, or 20% of the keywords, and for 100% of the keywords, respectively, and "no threshold". These numbers correspond to a pretty high performance, especially when taking into account the very high recall values.

We note that in both locations, the precision and recall values are exactly the same when using 10% and 20% of the keywords, while the precision reduces suddenly after that. A close look at the dictionaries of the two locations revealed that the most important keywords are the names of the locations. The small dictionary size employed (when using 10-20% of the keywords) is then occupied by these keywords. As the dictionary size increases, stopwords and noise are inserted, which have a negative impact on precision.

Discussion

Overall, our results show that using the correlation between local and global activity has the potential(when properly employed) to lead to significantly better accuracy.

We also observe that, contrary to previous work, the time needed to train and test our models depends on the percentage of keywords used. This allows us to achieve a trade-off between execution time and accuracy. An interesting point regarding this trade-off is the fact that the increase of the execution time that the *CLR* methods exhibit when using higher percentage of keywords, does not pay off with a proportional increase in precision and/or recall.

For the @$Top1$ case, the Tf-Idf based algorithms are the winners, providing better results than the simpler algorithms based on concordance. Furthermore, when using the Tf-Idf based algorithms, the best result is achieved when pruning some of the keywords. This is due to the fact that pruning the keywords with the lowest weight, we primarily remove stopwords, which has a positive impact on accuracy. This is also true for TG-TI-CLR1, when considering the F1 score.

Regarding the difference in precision and recall between our approach and the baselines, we believe that it is due to the very different granularity requirements of the problems, especially the temporal granularity. Even though the baselines provide good results for identifying the characteristic topics of a location (when there are enough data), our approach has an advantage for geolocalising tweets referring to time-focused events, especially those with a relatively short time-span (e.g., concerts).

3.4.3 Evaluation of SDpL and MDpL

Dataset, Keyword pruning and stopwords. The dataset that we used was the same used for Section 3.4.1, containing tweets posted from Italy between the period 20 June-23 July 2014. The only difference compared to the previous evaluation is that we even take into consideration the city of Florence. We also split the train and test data the same way we did at our previous works, 80% training and 20% testing, randomly shuffled between the train and test. Finally, we run the methods 10 times and we averaged the result. On the contrary to our previous methods, we were always using 100% of the keywords, while we were filtering out the stopwords of 11 languages, as defined by the nltk library of python.

Evaluation Measures. We initially wanted to evaluate our methods answering to all the Q-tweets, regardless of the similarity and probability that they had with the most similar location. Furthermore, we wanted to achieve a straight comparison with the methods presented in Section 3.4.1. Due to this, we initially evaluated our method on the neighborhoods of Milan. Afterwards, we evaluated our method on the city of Rome, and at the end we evaluated our method using as candidate locations, the locations of the three, five and seven most active cities of Italy (Milan, Rome, Venice, Naples, Florence, Bologna and Turin). Finally, we evaluated our method without the use of thresholds and with the use of manually assigned thresholds on the probability. In the case we had no threshold, we were answering to all the Q-Tweets. As a result the precision equals the recall (in the plots referred as accuracy).

Neighborhoods of One City

Applying our two methods on the city of Milan, we got an average accuracy of 18% for the model that was creating one single document per neighborhood (SDpL), while the accuracy of the method that was using many tweets per class (MDpL) was 39.93%. Comparing this accuracy to the one achieved in Section 3.4.1, when geolocalized all the Q-Tweets, our new method has slightly better accuracy (2%).

Having observed that Rome has higher activity, we also evaluated our method searching for tweets posted from the neighborhoods of Rome. The average accuracy that MDpL and SDpL achieved are 42.13% and 17.63% consecutively.

Increasing the search space

After the evaluation of our method on the squares of one city, we increased the search space targeting to find tweets deriving from equal squares (i.e. 1km side) of three, five and seven cities. In the case of the three cities, the accuracy was achieved by MDpL was 38.25% while SDpL achieved 15.58% accuracy. Increasing the search space to the neighborhoods of five cities, the accuracy of MDpL and SDpL was decreased to 37.2% and 14.6% respectively, while in the case of the seven cities the accuracy was reduced to 37% and 14.3%. The results of this evaluation and the trade-off between the search space and the accuracy are depicted in Figure 3.29.

Targeting to Higher Precision

Due to the fact that we target to enrich the existing datasets of geotagged tweets when having high confident for our answer, we targeted to the increase of the precision of our methods. In order to achieve the increase of the precision, we used manually assigned thresholds on the probability the most similar square has to be the origin of the tweet. The thresholds we applied were in the range between 0.1 and 0.7.

As we can see in Figures 3.30 and 3.32a, the application of the threshold on MDpL causes a tremendous increase of the accuracy. More precisely, the highest precision achieved by MDpL is 99.7%, with a recall of 9.48% and an F1 of 16.36%, and it is achieved in the case of seven cities, when applying a threshold of 0.6. On the other hand, the highest recall for MDpL is 24% and it is achieved for a threshold 0.1 at the case of one candidate city. In this case, the precision is 88.19% while the F1 is 35.1% , the highest among all the cases.

At the case of the SDpL, although the application of the threshold differentiates the precision and the recall of the algorithm, the effects are not similar. More precisely, the highest precision achieved by SDpL is 97.9% with a recall of 6.4%. As we can see in Figure 3.31a, the precision of SQpL is not affected much by the differences at the thresholds we apply. Furthermore, if we apply a threshold higher than 0.4, there is no candidate location bypassing the threshold, regardless the number of the candidate locations. This is probably caused due to the noise created when merging the documents into the single document.

Discussion

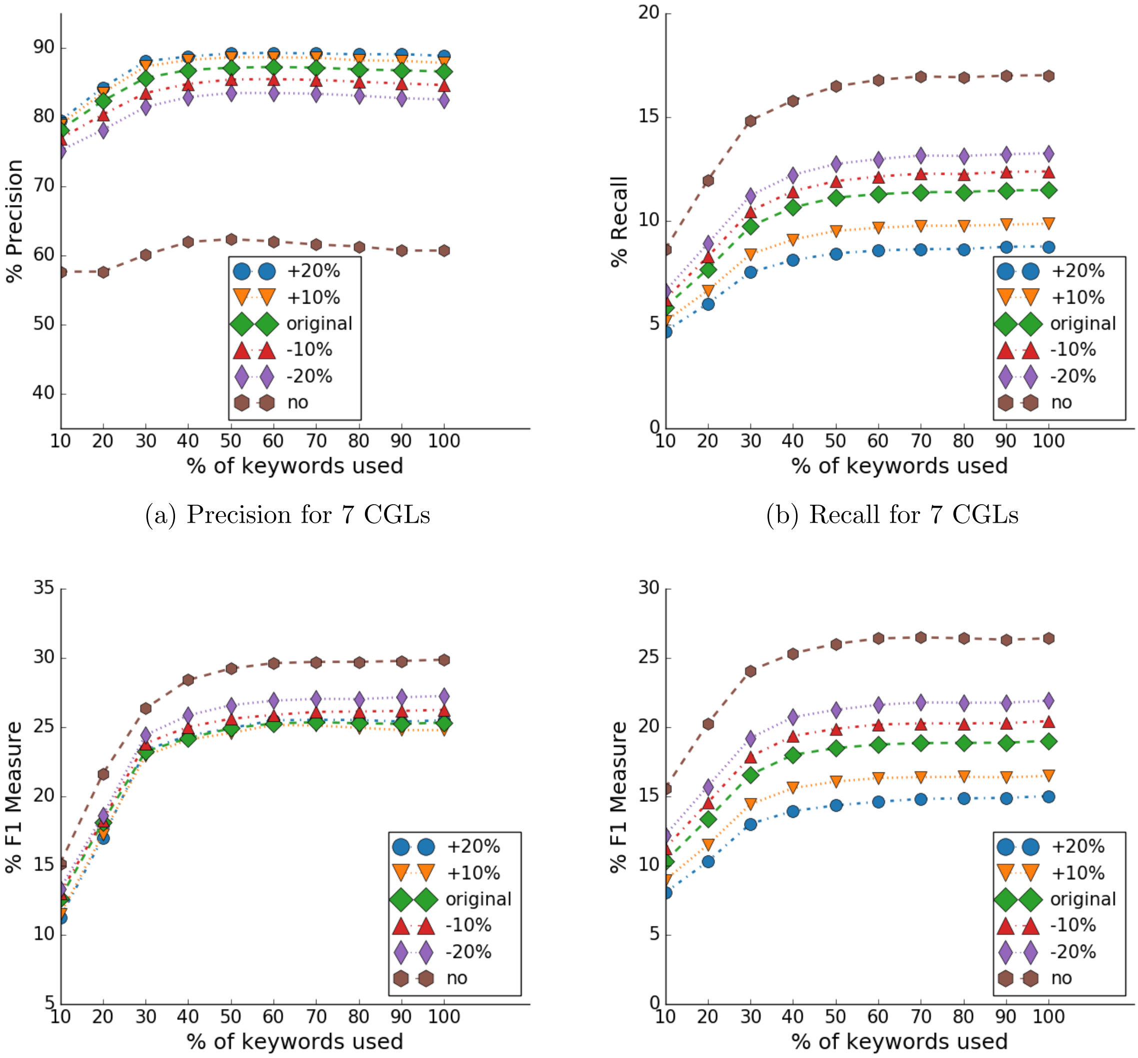
As shown in this part of the thesis, the method that uses logistic regression in order to geolocalize the non-geotagged tweets seems to be promising, achieving a slightly higher accuracy compared to the methods that use Tf-Idf. Comparing the two methods that use logistic regression, namely MDpL and SDpL, we come to the conclusion that using each tweet as a unique representative of a location, is still enough and more accurate compared to the case that we merge all the tweets from a location into one single document.

When targeting to precision, both MDpL and SDpL can achieve a very high precision, while MDpL can also keep the recall in descent levels. Due to the increase and decrease of the precision and recall of MDpL when using a threshold higher than 0.4, and combined to the fact that we have no answers from SDpL when using these thresholds, we assume that the ideal threshold relies on the timeslot and the geotagged data produced at that moment. As a consequence, the usage of a dynamic threshold and the further investigation of the ideal threshold seems to be the inevitable but also promising.

Finally, we observe that the more we increase the search space, the less the model is affected. This is probably due to the fact the locations we add to the search space have lower activity and less representative tweets. As a result, they do not affect much the training of the model.

3.5 Summary

In this chapter, we motivated the need of more geotagged information and we presented a novel method for analyzing and geolocalizing non-geotagged Twitter posts. The proposed method is the first to do so at the fine-grain of city neighborhoods, while being both effective and time efficient. Our method is based on the extraction of representative keywords for each candidate location,as well as the analysis of the tweet volume time series. Our experimental evaluation shows that we can increase the rate of the geotagged Twitter posts by 800%, with a precision of 89%.



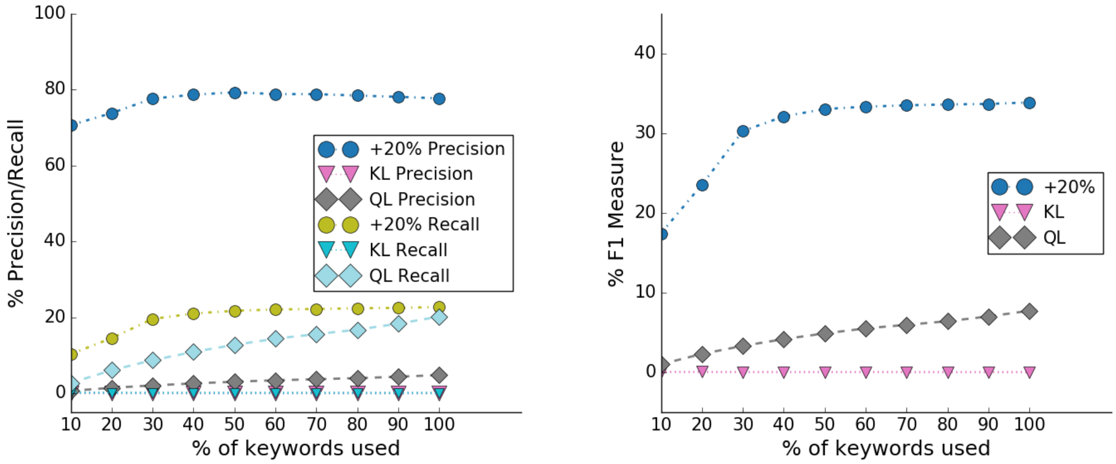
(a) Precision for 7 CGLs

(b) Recall for 7 CGLs

(c) F1 for 1 CGL (Milan)

(d) F1 for 7 CGLs

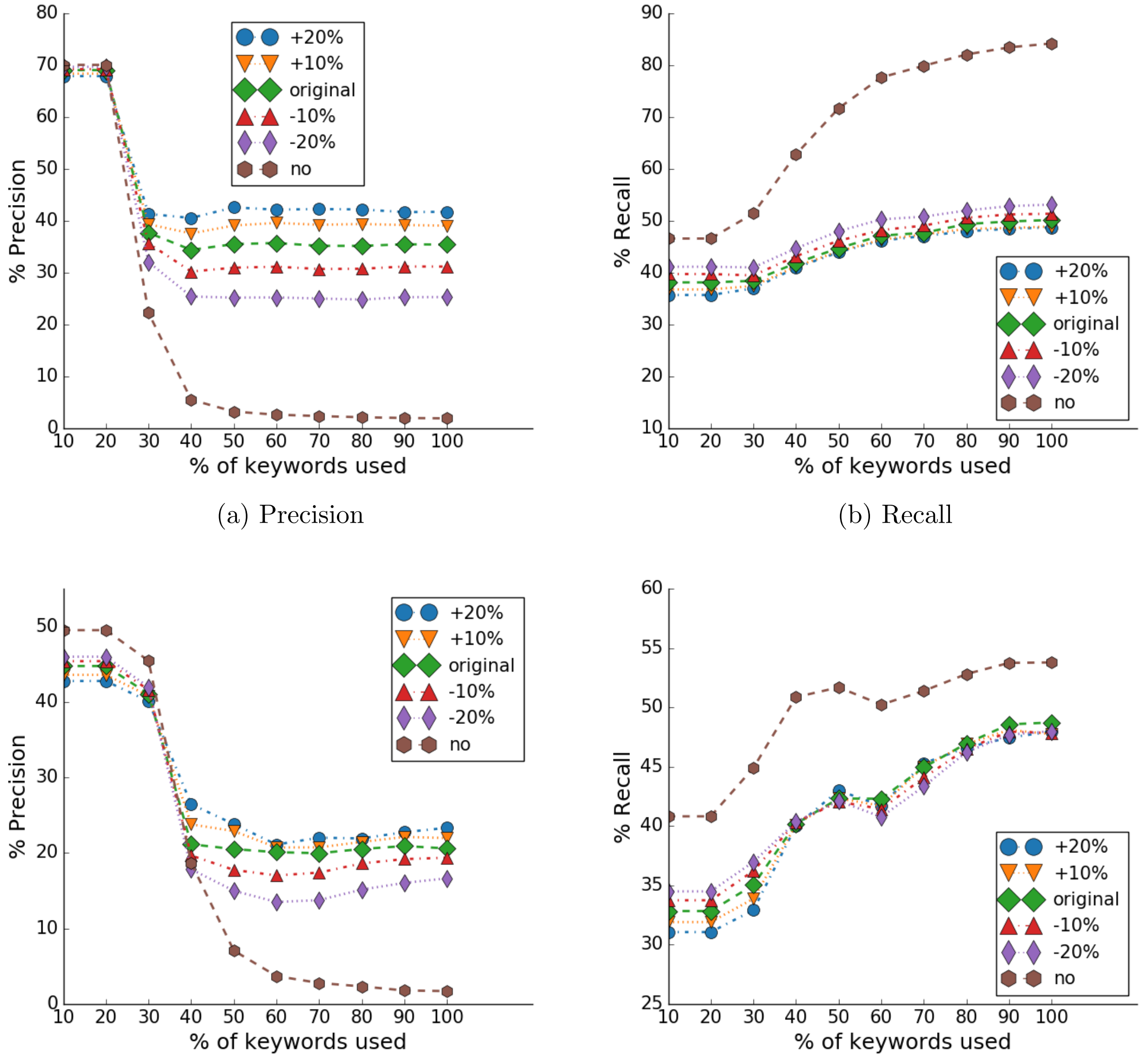
Figure 3.26: (Top) Trade-off Between Precision and Recall for 7 CGLs (Rome, Milan, Venice, Florence, Naples, Bologna, Turin, @Top1). (Bottom) F1 for 1 and 7 CGLs (@Top1).



(a) Precision and Recall

(b) F1

Figure 3.27: Precision, Recall and F1 Comparison for 7 CGRs (@Top1)



(a) Precision

(b) Recall

(c) Precision

(d) Recall

Figure 3.28: (Top) Vatican (1.3km-side square / @Top1). (Bottom) San Siro (0.8km-side square / @Top1).

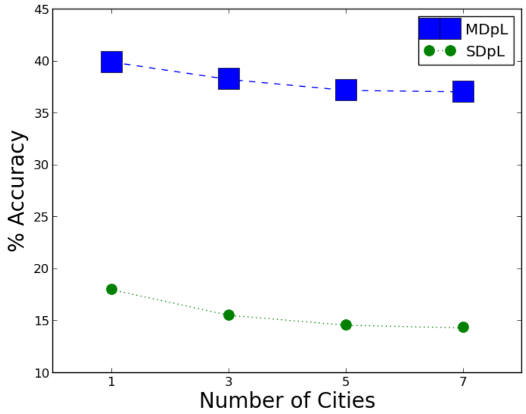
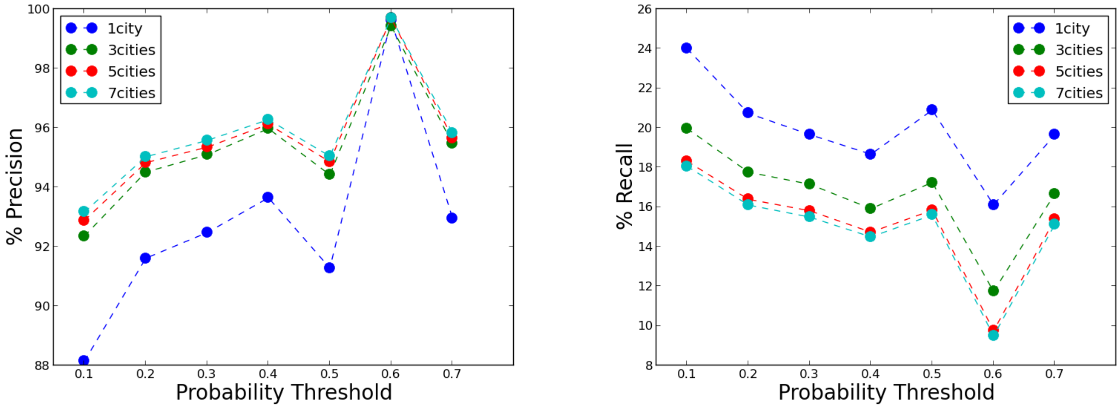


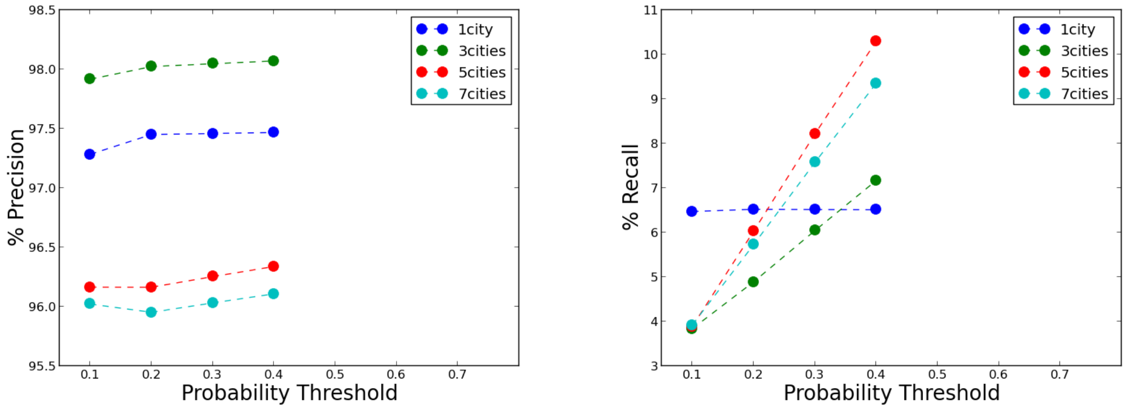
Figure 3.29: Trade-off Between Accuracy and the Number of Candidate Locations (MDpL and SDpL)



(a) Precision

(b) Recall

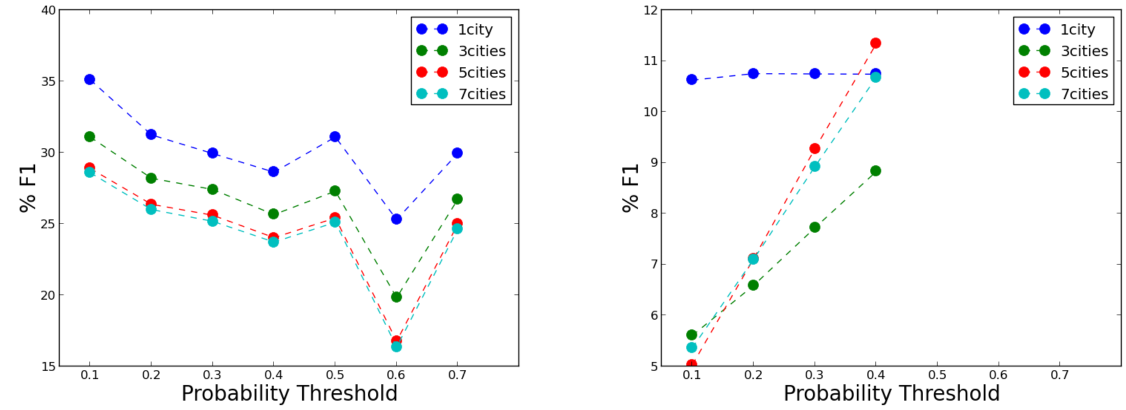
Figure 3.30: Precision and Recall of MDpL



(a) Precision

(b) Recall

Figure 3.31: Precision and Recall of SDpL



(a) MDpL

(b) SDpL

Figure 3.32: F1 of MDpL and SDpL

Chapter 4

A system for Geolocalization of non-Geotagged Posts

In this chapter we present the TweeLoc system, which is built around the TG-TI-CLR1 algorithm (as presented in 3.3.4) and targets to provide the user a user-friendly interface, while presenting a more simple view of the geolocalization of non-geotagged tweets. As we have already mentioned, in the process of analysis of identifying the information originating from social networks, and especially Twitter, an important aspect is that of the geographic coordinates, i.e., geolocalisation, of the relevant information. Geolocalized information can be used by a variety of applications in order to offer better, or new services. However, only a small percentage of the twitter posts are geotagged, which restricts the applicability of location-based applications. In this work, we describe TweeLoc, our prototype system for geolocalizing tweets that are not geotagged, which can effectively estimate the tweet location at the level of a city neighborhood. TweeLoc, which is language agnostic as TG-TI-CLR1 is, employs a dashboard that visualizes the social activity of the geographic regions specified by the user, and provides relevant easy-to-access statistics. Moreover, it displays information on the way that these statistics evolve over time. Our system can help end-users and large-scale event organizers to better plan and manage their activities.

4.1 The TweeLoc System

We now describe the TweeLoc architecture which is presented in Figure 4.1.

The input to our system are the tweets deriving from the public API of Twitter, or alternatively from a *json* file that contains historical tweets, as well as a file with all required initialization parameters. The parameters are user-defined, and they refer to the

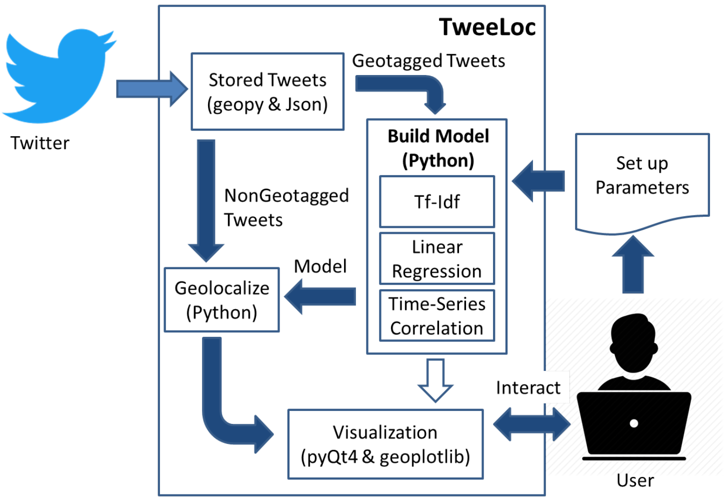


Figure 4.1: TweeLoc Architecture

bounding box of the CGLs in interest, the space resolution of the FGLs (by default: 1 square km), the length of a timeslot in minutes (by default: 15), the number of timeslots in a window (by default: 16), the percentage of tweets to use for training (by default: 80%), the elasticity of the threshold (by default: +20%), whether we focus on a specific language or not (by default: no), the set of stopwords to filter out during preprocessing (by default: no stopwords filtered), and the percentage of keywords we want to keep in our keyword-vectors (by default: 60%).

TweeLoc accesses the Twitter stream using the python library "geopy" [1]. The downloaded tweets are processed in batches (one timeslot at a time): initially stored in a *json* file, and then "fed" to our system for building the model of each location (CGL and FGL). In this way, we can process both live and historical data using the same workflow. Note that the latency that this choice imposes to the processing of the live data (as low as a few minutes) is not a show-stopper for the applications targeted by TweeLoc.

The proposed system utilizes the TG-TI-CLR1 algorithm (described earlier) for building the model and estimating the locations of non-geotagged tweets. This part of the system was built using Python 2.7. The geolocalized tweets are then passed on to the visualization layer, which overlays their positions on maps, along with additional statistics.

The geographical maps that we use are composed of tiles downloaded from "Openstreetmap". These tiles change whenever we zoom-in or zoom-out. The visualizations that use heatmaps are using a modified version of the "geoplotlib"[2] Python library. Finally, we use the python library "pyQt4"[3] that handles graphic elements, and is useful for

[1] https://github.com/geopy/geopy
[2] https://github.com/andrea-cuttone/geoplotlib
[3] https://pypi.python.org/pypi/PyQt4

visualizing individual tweets on a geographical map, along with the tweet text and other metadata.

4.2 System Functionality

TweeLoc is a system that can work with both static and live Twitter data. In what follows, we describe the functionalities of our system, as well as the different ways the user is able to interact with the system. The goal is to present the benefits of TweeLoc's fine-grained geolocalization, and its ability to support location-based applications that would otherwise not be possible.

In the following paragraphs, we describe the functionalities our system provides to the user.

1. Hotspot Identification: The first functionality of our system is to allow the user experience how TweeLoc provides a much more detailed spatial exploration of the data than previous methods. TweeLoc first displays to users a geographical map of the selected area, overlayed with a heatmap of all geolocalized tweets, as shown in Figure 4.2a (the black color corresponds to places with low activity, red with medium activity, and yellow with high activity). Unlike earlier approaches, the user will be able to zoom in a specific city in order to create a fine-resolution map (an example is shown in Figure 4.2b). At this level of detail, the user can observe the Twitter activity as it unfolds in the different neighborhoods of a city, and identify the most popular spots in the city.

In this case, the user can choose among the different datasets, and also interactively decide on which city (and for the case of the static datasets, the time interval, as well) to focus on.

2. Activity Analysis: The second functionality that our system provides to the user, is focused on the analysis of the activity dynamics of the tweets. The interface depicted in Figure 4.3a visualizes a heatmap based on the number of tweets that were posted from each individual FGL (i.e., square in the grid). In this view, when the user hovers with the mouse over a square, a bubble appears that shows the representative keywords of that square, corresponding to the content of the tweets of that square. The user could also switch to an alternative view, visualizing a differential heatmap (Figure 4.3b), which visualizes the way that the activity of each FGL changes (i.e, increases, or decreases) between two timeslots. In this case, each square shows the percentage of the activity change, and is colored in green when the activity increases over time, otherwise in red. In all heatmap views, the upper right corner of the window displays the name of the heatmap, the starting time of the window, and its length in minutes. This implementation enables TweeLoc to reveal the activity of different neighborhoods in a city and identify hotspots, explain this activity in terms of the contents of the tweets, and also explore how this activity evolves over time.

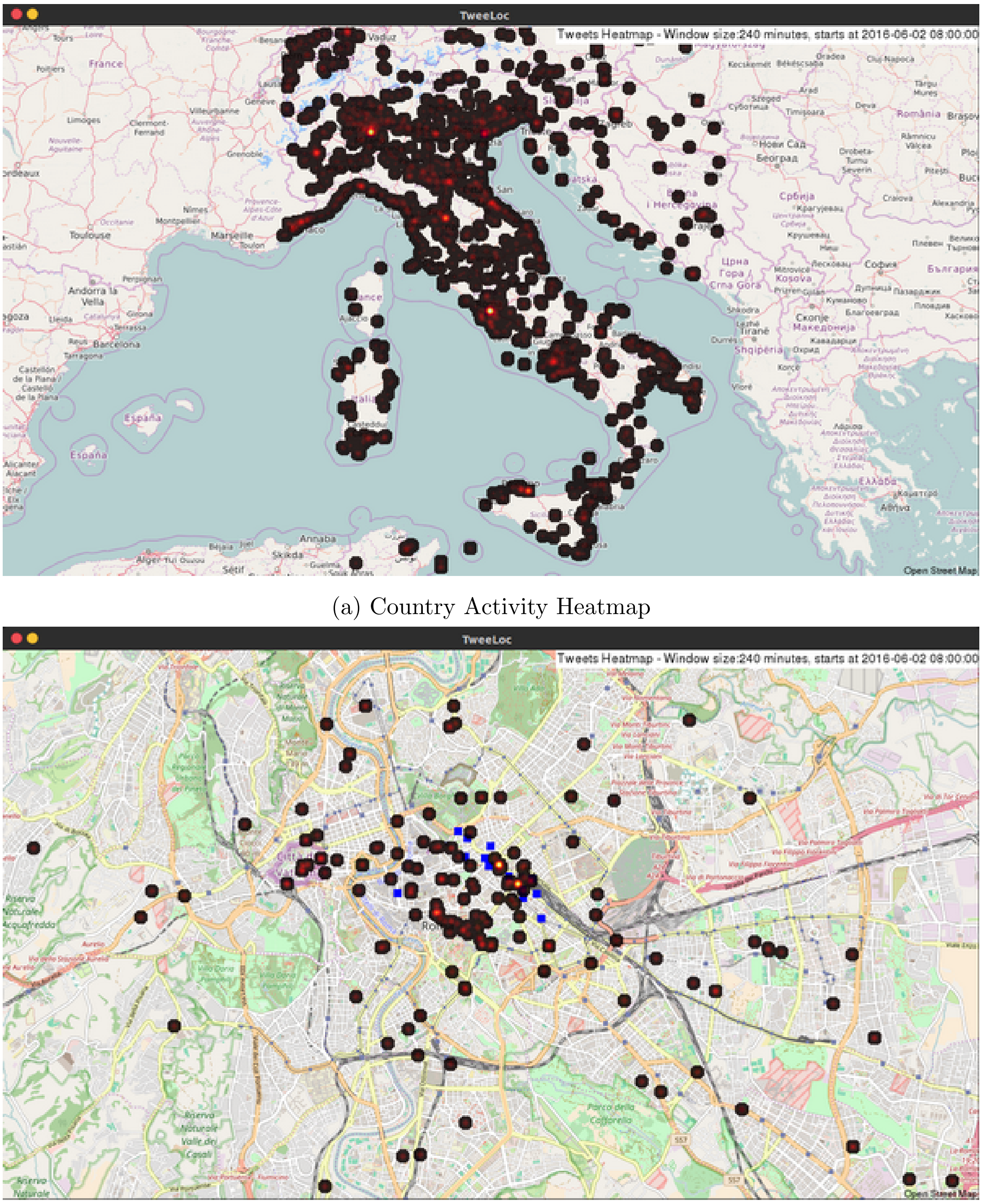
Utilizing this implementation, the users are able to explore the Twitter activity dynamics for different cities, and also decide on the dataset used (including the live stream). They will also be able to navigate across time (except for the live dataset), effectively changing the time window (i.e., timeslot) under consideration.

3. Targeted Statistics: The third functionality that our system provides to the user, is the ability to check the location of specific, individual tweets, as they appear in the live stream. The text of the tweets will be displayed on the screen, and when clicked on, the system will display the predicted position of that tweet on the map, by automatically zooming-in to the *FGL* identified as the tweet location (Figure 4.4a). The system can additionally display a list of representative keywords for all the tweets posted from that same *FGL*. Furthermore, by clicking on the *FGL* position, a new window will pop-up, depicting the volume of tweets over time that were posted from that *FGL* (Figure 4.4b). The interface will also provide a "Locate all" button, for geolocalizing all the individual tweet posts currently displayed on the screen.

In this implementation, the users can choose individual tweets from the live stream.

4.3 Summary

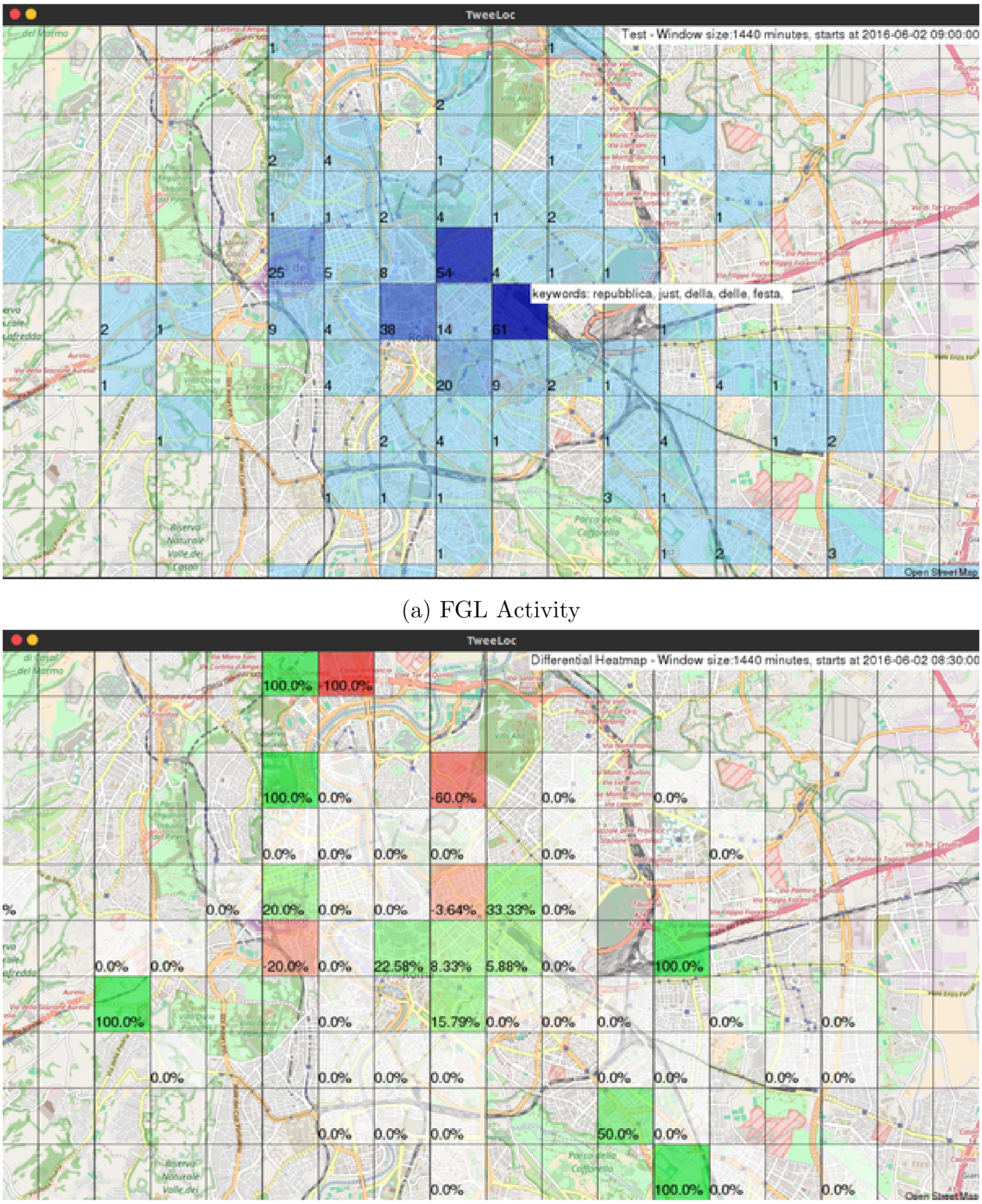
In this chapter, we described our TweeLoc system, which geolocalizes non-geotagged Twitter posts and allows users to visually examine the results and their evolution over time. Our system allows the user to get a better idea of how the activity of a particular location changes, which the most important keywords are, as well as to geolocalize individual tweets of interest.


TweeLoc
Tweets Heatmap - Window size:240 minutes, starts at 2016-06-02 08:00:00
Open Street Map
TweeLoc
Tweets Heatmap - Window size:240 minutes, starts at 2016-06-02 08:00:00
Open Street Map

(a) Country Activity Heatmap

(b) Rome Activity Heatmap

Figure 4.2: Country and City Activity Heatmaps



TweeLoc

Test - Window size:1440 minutes, starts at 2016-06-02 09:00:00

keywords: repubblica, just, della, delle, festa,

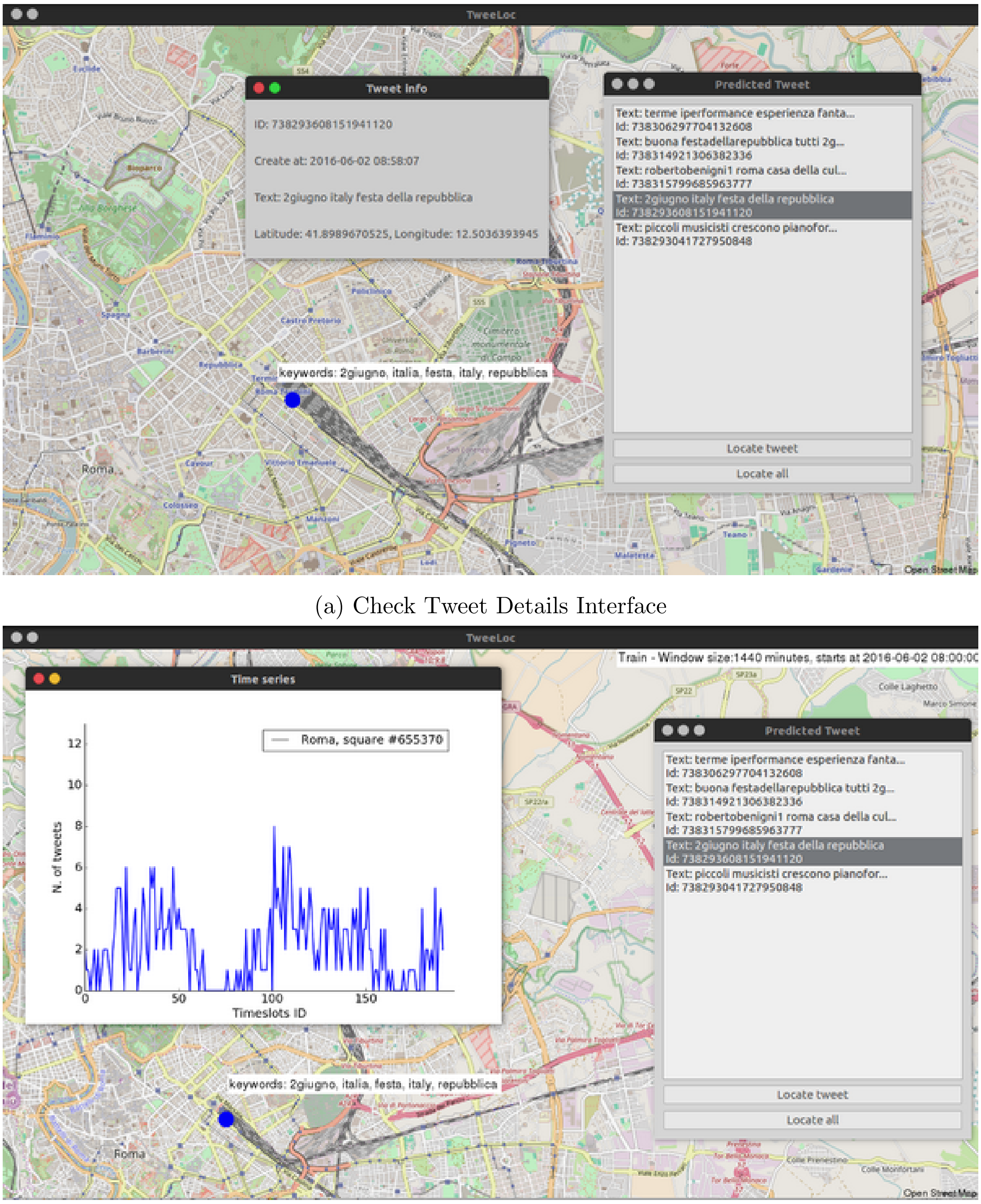
(a) FGL Activity

TweeLoc

Differential Heatmap - Window size:1440 minutes, starts at 2016-06-02 08:30:00

(b) Differential Heatmap

Figure 4.3: FGL Activity and Differential Heatmaps



ID: 738293608151941120

Create at: 2016-06-02 08:58:07

Text: 2giugno italy festa della repubblica

Latitude: 41.8989670525, Longitude: 12.5036393945

Text: terme iperformance esperienza fanta...
Id: 738306297704132608
Text: buona festadellarepubblica tutti 2g...
Id: 738314921306382336
Text: robertobenigni1 roma casa della cul...
Id: 738315799685963777
Text: 2giugno italy festa della repubblica
Id: 738293608151941120
Text: piccoli musicisti crescono pianofor...
Id: 738293041727950848

(a) Check Tweet Details Interface

Train - Window size:1440 minutes, starts at 2016-06-02 08:00:00

Text: terme iperformance esperienza fanta...
Id: 738306297704132608
Text: buona festadellarepubblica tutti 2g...
Id: 738314921306382336
Text: robertobenigni1 roma casa della cul...
Id: 738315799685963777
Text: 2giugno italy festa della repubblica
Id: 738293608151941120
Text: piccoli musicisti crescono pianofor...
Id: 738293041727950848

(b) FGL Activity

Figure 4.4: FGL and Activity Tweet Details

Chapter 5

What do Geotagged Tweets Reveal about the Users?

5.1 Introduction

People's attention tends to be drawn by important-unique events, such as concerts and football games. Many of them are even willing to travel long distances in order to attend events they regard as unique. As a result, the everyday pattern that a user has, changes. This includes changes in the routes the user normally follows and the calling and social activities. In this chapter, we investigate the difference in the activity and movement between users that either attend a unique event or visit an important location and users that do not. Furthermore, based on the activity of users that attend an event, we investigate the way we can get a representative sample of users that has the potential to reveal some important characteristics. Example of such characteristics are the main routes the users tend to follow and important locations the users head to.

5.2 Problem Description

The problem we want to investigate in this chapter is the identification of activity differences between the users who have attended an important event (i.e. concert) and those who haven't, while examining the distances a user is willing to travel in order to attend such an important event. Furthermore, we want to study the reasons that force a user to generate a geotagged message.

Finally, we would like to examine the extraction of a sample of users in a social network, that could allow us to reproduce the main routes that the users prefer to follow in order to move.

In the context of this work, we concentrate on users who attend major events or sights, such as concerts, or an important touristic attraction. Furthermore, we focus on Twitter, a social network that has more than 313M users, 80% of which are on mobile devices.

5.3 Proposed Approach

In this section, we describe the method we developed for tackling the problems previously described (for the general schema, refer to Algorithm 8).

Our method is based on the creation of social ties [77], where as social ties we define the connection between users, that maybe do not have many characteristics in common but at a time-interval t where at the same location, sharing the experience of a unique event or important location, while afterwards they left the location independently.

Initially we set the temporal and spatial parameters we are interested in, afterwards we remove the spam or bot accounts based on the activity of the account. Finally, we follow the geolocalized posts a non-bot user sent during a predefined period of time.

In the following sections, we elaborate on the methods discussed above.

5.3.1 Setting the temporal and spatial parameters

We start by setting up the temporal and spatial parameters we are interested into:

1. loc_{ev}: the location the event is going to take place
2. win_{ev}: the period of time we will identify users who visited loc_{ev}
3. CGL: the coarse-grain location for which we will track the movement of the users
4. $WinInterest$: the period of time we will follow the users' geotagged posts

5.3.2 Get the Event and CGL users

In order to get the initial sample of our users, we use the spatio-temporal parameters and we check our dataset for users who posted at least one geotagged tweet from the event

Algorithm 8 Get Representative Sample and Characteristics

INPUT: Temporal and Spatial parameters.

OUTPUT: A representative sample of users and its activity and movement.

```
    $P_{WinInterest}, Q_{WinInterest} \leftarrow GetUsers(loc_{ev}, win_{ev}, CGL, WinInterest)$    ▷ get the users from the event *location* and the $CGL$
    $users, activity, movement \leftarrow$ Percentage of top uses in P, Q    ▷ get the representative users' sample
    **return** $users, activity, movement$
```

location, before, during or after the event (win_{ev}). Afterwards, we get all the geotagged tweets these users posted, for a predefined period of time ($WinInterest$).

Having already extracted the users who attended the event, we get the rest of the users from our CGL that have at least one geotagged tweet during the win_{ev} and they have no tweets from the loc_{ev} during this time interval.

The steps that we follow in order to get the users we are interested in, are presented in Algorithm 9.

5.3.3 Cleaning the Dataset

There are a lot of accounts that are either bots sending posts with the same content for a long period of time, or accounts that are sending posts with different content, from the exact same location. Due to the fact that these accounts do not offer any important information, while also creating noise in our analytics, we chose to filter them out. In order to identify these accounts, we use three naive conditions:

1. at least 30% of the messages posted by this account had the same prefix
2. at least 50% of the messages posted by this account had the same latitude
3. at least 50% of the messages posted by this account had the same longitude.

If an account meets at least two of the three conditions, we filter out the account (Algorithm 10).

5.3.4 Activity and Movement Comparison

After the extraction of the datasets of the location place and the CGL, we compare their activity using the cumulative distribution function (CDF). Using the CDF, we can

Algorithm 9 Get Users

```
procedure GetUsers($loc_{ev}, win_{ev}, CGL, WinInterest$)
    for all $u \in \{loc_{ev}\}$ do        ▷ get first sample of users in $loc_{ev}$ and their activity
        $U_{loc_{ev}win_{ev}} \leftarrow$ all users at $loc_{ev}$ at time-window $win_{ev}$
        $P^{u,CGL}_{WinInterest} \leftarrow$ all tweets from user $u$ at time-window $WinInterest$
    for all $u \in \{CGL\}$ do        ▷ get all users in $CGL$ and their activity
        if then$u$ not in $U_{FGL_{win_{ev}}}$
            $Q^{u,CGL}_{WinInterest} \leftarrow$ all tweets from user $u$ at time-window $WinInterest$
    $P_{WinInterest} \leftarrow$ SpamFilter($P^{CGL}_{WinInterest}$)        ▷ clean spam and bot accounts from $P^{CGL}_{WinInterest}$
    $Q_{WinInterest} \leftarrow$ SpamFilter($Q^{CGL}_{WinInterest}$)        ▷ clean spam and bot accounts from $Q^{CGL}_{WinInterest}$
return $P_{WinInterest}, Q_{WinInterest}$
```

compare the activity between the users who visited the event locations and those who did not. Furthermore, we check the distribution of the points they moved during the $WinInterest$. In order to achieve this, we compare the difference between the maximum and minimum latitude and longitude the user appeared. The idea we want to verify using those two steps is that users tend to travel long distances in order to visit a unique event or a unique location. Furthermore, we want to examine the fact that the users are more willing to share their location in case they attend important events, as opposed to their normal activity patterns.

5.4 Experimental Evaluation

In order to evaluate our ideas, we used geotagged posts from Twitter. The datasets used, contain events such as unique concerts and important touristic locations. In this chapter, we present a set of activity and movement analytics, while we provide the reader with visualizations of the location we get the tweets from.

5.4.1 Datasets

For the evaluation of our methods, we used two datasets. The first contains geotagged tweets generated from Italy for the period between 1st of June and 15th of November 2016. The second dataset contains geotagged tweets generated from the central Europe, covering Belgium, Germany, the Netherlands, Luxembourg and a part of France (up to Paris) and are posted during the period between 1 of April and 1 of July 2015. We focused on important locations and events that took place during these time intervals. More precisely, we targeted users who posted geotagged posts from Vatican and the concert of Bruce Springsteen (which in our experiments is referred as $Concert1$) that took place in Rome, the concert of Taylor Swift (which in our experiments is referred as $Concert2$) that took place in Koln and the European Parliament in Brussels.

Algorithm 10 Spam and Bot Filtering

```
procedure SpamFilter($Users, P^{Users}_{WinInterest}$)
    for all $u \in \{Users\}$ do
        if $\leq 30\%$ of the $P^{u}_{WinInterest}$ have the same prefix then
            if $\leq 50\%$ of the $P^{u}_{WinInterest}$ have same latitude then
                if $\leq 50\%$ of the $P^{u}_{WinInterest}$ have same longitude then
                    Add $u$ to $sample$
  return $sample$
```

Rome

For the city of Rome, we focused on two different types of users, the users who attended a unique event, and those who visited an important location.

People Attending *Concert*1

We initially focused on a important event that took place in Rome and attracted a lot of people. This event was the concert of Bruce Springsteen (i.e. *Concert*1) that took place at the location "Circus Maximus" on 16 of July 2016. We found the users that visited this location and posted a geotagged post since the midnight of the previous day. The time windows that we used were 24 hours and 48 hours (it was a 2-day concert), searching for posts initially posted up to the end of the concert (i.e. 24 hours) and afterwards also the following day (i.e. 48 hours). Having identified the users who generated messages from this location during our window, we followed all their geotagged posts for the period between 1st of June and 15th of November 2016.

After further analyzing the activity of these users, we found that it was a sample of 67 non-spamming users and the:

1. 100% of the users, have average activity of 25 posts, while the standard deviation is 31
2. 75% of the most active users, have average activity of 33 posts, while the standard deviation is 32
3. 50% of the most active users, have average activity of 45 posts, while the standard deviation is 32
4. 25% of the most active users, have average activity of 68 posts, while the standard deviation is 33.

When we decrease the number of users in our sample by keeping a percentage of the most active ones, the standard deviation of the activity of the users is not affected much, while the mean activity of the users decreases. This fact implies that the distribution of the activity of the users is similar for all the users in our sample.

In Figure5.1a we depict the locations these 67 users "appeared" at, while in Figures 5.1b,5.1c,5.1d we can see respectively the locations the 75%, 50% and 25% most active users posted geotagged tweets from, for the period June to November. In all the plots we present in this Section, each color represents a different user[1] As we can see in Figure 5.1a, the combination of mobility and activity patterns of these 67 users cover the entire country of Italy: they are able to form the main shape and the main routes of the country. This is

[1] Due to the relatively high number of users, different users may share the same color.

still true when we consider the 50% most active of these users (see Figure 5.1c), and almost true even when we limit the number of the users to 17 (25% of most active, Figure 5.1d).

These results reveal some very interesting characteristics of our dataset (and users). They indicate that an extremely small number of users is mobile enough in order to cover the entire country. Recall that the users in the sample we examined belong to a particular demographics group, namely, they all attended a specific music concert. Nevertheless, this observation can lead to interesting marketing applications.

After having checked the activity and the locations of the people identified using the 24-hour window, we analyzed the people identified by the 48-hour window. The volume of the sample was increased to 144 users and their average activity and its standard deviation was:

1. 100% of the users, have average activity of 19 posts, while the standard deviation is 27

2. 75% of the most active users, have average activity of 25 posts, while the standard deviation is 29

3. 50% of the most active users, have average activity of 36 posts, while the standard deviation is 31

4. 25% of the most active users, have average activity of 57 posts, while the standard deviation is 33.

As we noticed in the case of the 24-hour window, sub-sampling with the most active users does not affect much the standard deviation of the activity. Furthermore, the mean activity is slightly decreased compared to the one of the case of the 24-hour window, while the standard deviation is similar. This implies that the activity of the 68 users identified at the concert location during the second day, does not differ to the activity of the users of the first day.

In Figure 5.2, we can see the locations of the 144 users identified at the concert for the 48-hour window. The fact that we increased the window, appending users to our dataset, provided us with more geotagged tweets. Due to this, we have more points in our plots, showing more precisely the map of Italy and the main roads. Furthermore, comparing Figures 5.1c (which is formed by 34 users) and 5.2d (which is formed by 36 users) we notice that the shape of Italy formed by the 36 users is much more representative. This is due to the fact that the users, whose activity is depicted in Figure 5.2d, have in general higher activity.

Finally, in order to check the impact of the concert to the area, we slightly modified our parameters, targeting users that visited the concert area one week before the concert took place. Even though the area is located in the center of Rome, only 6 users had posted geotagged messages from this location during a 24-hour window. This means that the concert was indeed the reason that the users made geotagged posts (as can also be verified by the content of the posts).

Vatican

Having analyzed the activity of the users who attended an important unique event such as a concert, we turned our focus on one of the most important locations of Rome, the Vatican. We followed exactly the same procedure we did in the case of the concert, modifying only the location whose visitors we were interested in.

After analyzing the activity of the visitors' of Vatican using the 24-hour window, we found that 48 users posted geotagged tweets from Vatican during this window. The activity of these users was for:

1. 100% of the users, the average activity is 23 posts, while the standard deviation is 42

2. 75% of the most active users, the average activity is 30 posts, while the standard deviation is 47

3. 50% of the most active users, the average activity is 42 posts, while the standard deviation is 54

4. 25% of the most active users, the average activity is 69 posts, while the standard deviation is 69.

After changing the length of the window to 48 hours, the volume of the users has been increased to 91, while the average activity and standard deviation have been decreased. More precisely:

1. 100% of the users, have average activity of 25 posts, while the standard deviation is 56

2. 75% of the most active users, have average activity of 32 posts, while the standard deviation is 64

3. 50% of the most active users, have average activity of 45 posts, while the standard deviation is 74

4. 25% of the most active users, have average activity of 79 posts, while the standard deviation is 96.

Contrary to the case of the *Concert*1, the standard deviation of the activity of the users that visited Vatican is affected when limiting the sample to the most active users. In Figures 5.3 and 5.4, we depict the locations of the users that visited Vatican, the same day that the concert was, and posted a geotagged post from Vatican for a 24-hour and 48-hour window, consecutively. As opposed to the case of the users who attended the *Concert*1, the shape of the map of Italy that is formed is not very clear. This difference is more obvious when comparing the Figure 5.1b, which was created using a sample of 50 users, with the Figure 5.3a, which is created using a sample of 48 users. In the case of the 48-hour window, this comparison is possible between the Figures 5.2b (108 users) and 5.4a (91 users). Possible explanations for this behavior include the fact that users have traveled from other locations in order to attend the concert, or that the majority of the users, who visited Vatican, are tourists whose home-location is outside of Italy. Nevertheless, these results highlight the different mobility and activity behaviors of these two different samples of users.
*Concert*2

After the analysis of the activity using the dataset from Italy, we wanted to evaluate our ideas using the second dataset, containing posts from the central Europe. The concert of the famous singer Taylor Swift took place in the German city Koln on 16 of June 2015. The procedure that we followed was the same as before, modifying only the period that we were following the users. More precisely, we were interested in posts posted between 1 of April and 1 of July 2015, by the users who attended the concert. After analyzing the activity of these users who passed by the concert area between the 20 of June and 21 June (24-hour window), we found that they were 44 users. The average activity and its standard deviation were:

1. for 100% of the users, the average activity is 19 posts, while the standard deviation is 49

2. for 75% of the most active users, the average activity is 25 posts, while the standard deviation is 55

3. for 50% of the most active users, the average activity is 36 posts, while the standard deviation is 66

4. for 25% of the most active users, the average activity is 65 posts, while the standard deviation is 87.

As happened with *Concert*1, *Concert*2 was also a 2-day concert. Due to this, we increased our window to 48 hours, targeting to get even the users who attended the concert at the second day. The number of the users was increased to 59 users while the activity was modified as follows:

1. 100% of the users, had average activity of 17 posts, while the standard deviation is 43
2. 75% of the most active users, had average activity of 22 posts, while the standard deviation is 48
3. 50% of the most active users, had average activity of 31 posts, while the standard deviation is 57
4. 25% of the most active users, had average activity of 53 posts, while the standard deviation is 76.

Contrary to the users who attended $Concert1$, both the mean activity of the users who attended $Concert2$ and its standard deviation are affected when keeping the most active users. This is probably due to the small sample. Nevertheless, we can see in Figures 5.5 and 5.6 that the mobility of these users is high, indicating that the users tend to travel in order to attend a unique event, such as a concert.

The concert area is located in the center of the city. Due to this, we checked the activity close to the concert area 2 weeks before the concert. Unfortunately, we got only 4 users that posted from this area for this period. Due to this, the results cannot be comparable.

European Parliament

Finally, we wanted to analyze another important area in order to check if the activity of these users follows the example of Vatican. Such an important area is the "European Parliament", located in Brussels. Although the areas that we targeted in the previous cases were of equal size, due to the fact that people do not have access to the parliament, we extended the area of interest so that we capture even tweets around the parliament. The period that we were interested in was the same as the period we used for the concert in Koln. The positions of the users that posted geotagged posts around the parliament at the period between 19 of June and 20 of June 2016 (24-hour window) are depicted in Figures 5.7 and 5.8. The number of the users that visited the parliament during the 24-hour window was 24, while for the 48-hour window was 46.

After analyzing the activity of these users, we found out that for the 24 users, when keeping:

1. 100% of the users, the average activity is 32 posts, while the standard deviation is 28

2. 75% of the most active users, the average activity is 41 posts, while the standard deviation is 28
3. 50% of the most active users, the average activity is 57 posts, while the standard deviation is 23
4. 25% of the most active users, the average activity is 77 posts, while the standard deviation is 19.

Regarding the 48-hour window, the 46 users had:

1. average activity is 27 posts, while the standard deviation is 27
2. average activity is 34 posts, while the standard deviation is 28
3. average activity is 46 posts, while the standard deviation is 28
4. average activity is 69 posts, while the standard deviation is 22.

After further analyzing the results and comparing them to the previously analyzed locations, we noticed that contrary to the other locations, the higher the pruning of the users, the lower the standard deviation. This characteristic implies the homogeneity of the activity of the users that post when visiting important places, while there is no other interesting unique event around this location.

5.4.2 Top and Random Users from Italy

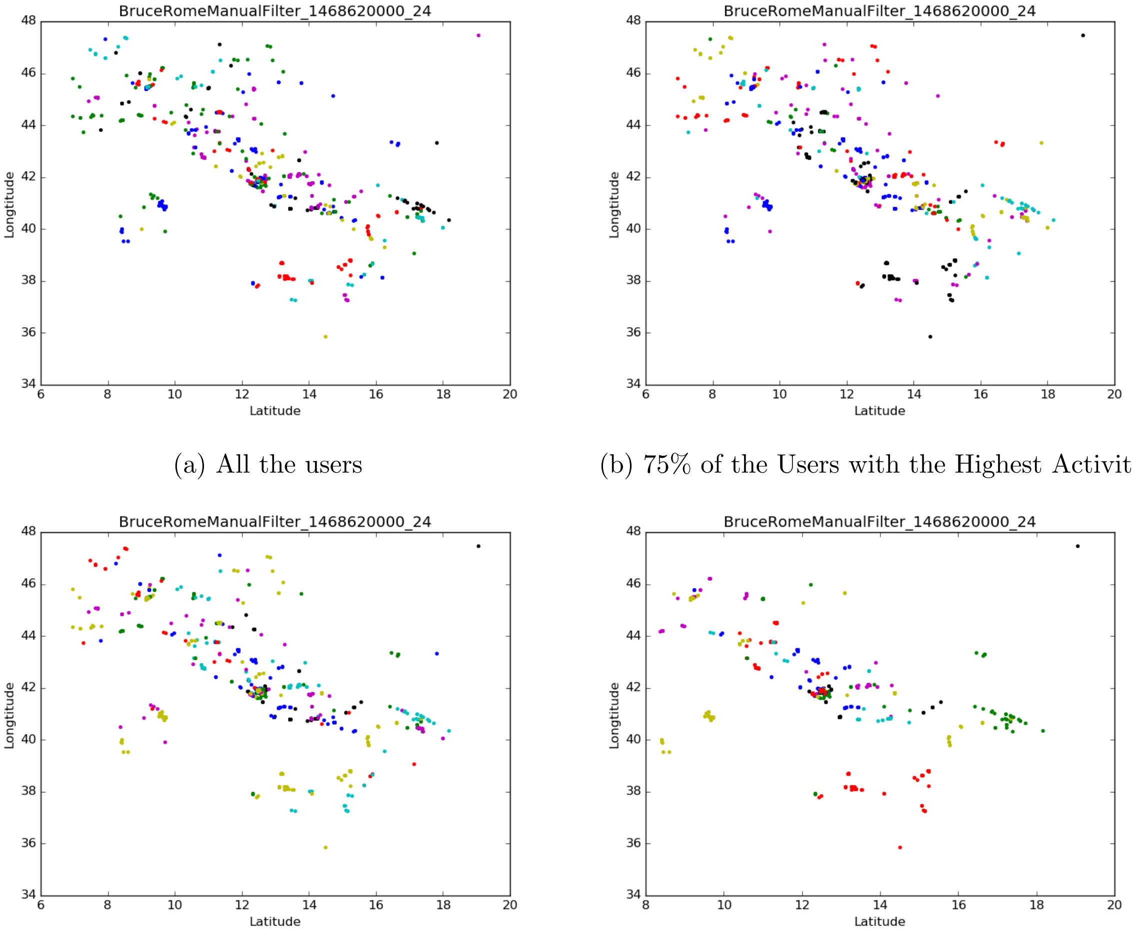
Having analyzed the activity of the users who either attended an event (i.e. concert) or visited an important location, we wanted to compare their activity with the people who haven't been located in one of the previous cases. In order to achieve our target, we have identified and followed users:

1. in Italy
2. in Rome

that at the day of the *Concert*1, were not located at the location the *Concert*1 took place.

In order to make the comparison fare, we keep ontly n users, where n is the number of the

1. 100% of the number of the users who attended the *Concert*1
2. 75% of the number of the users who attended the *Concert*1
3. 50% of the number of the users who attended the *Concert*1



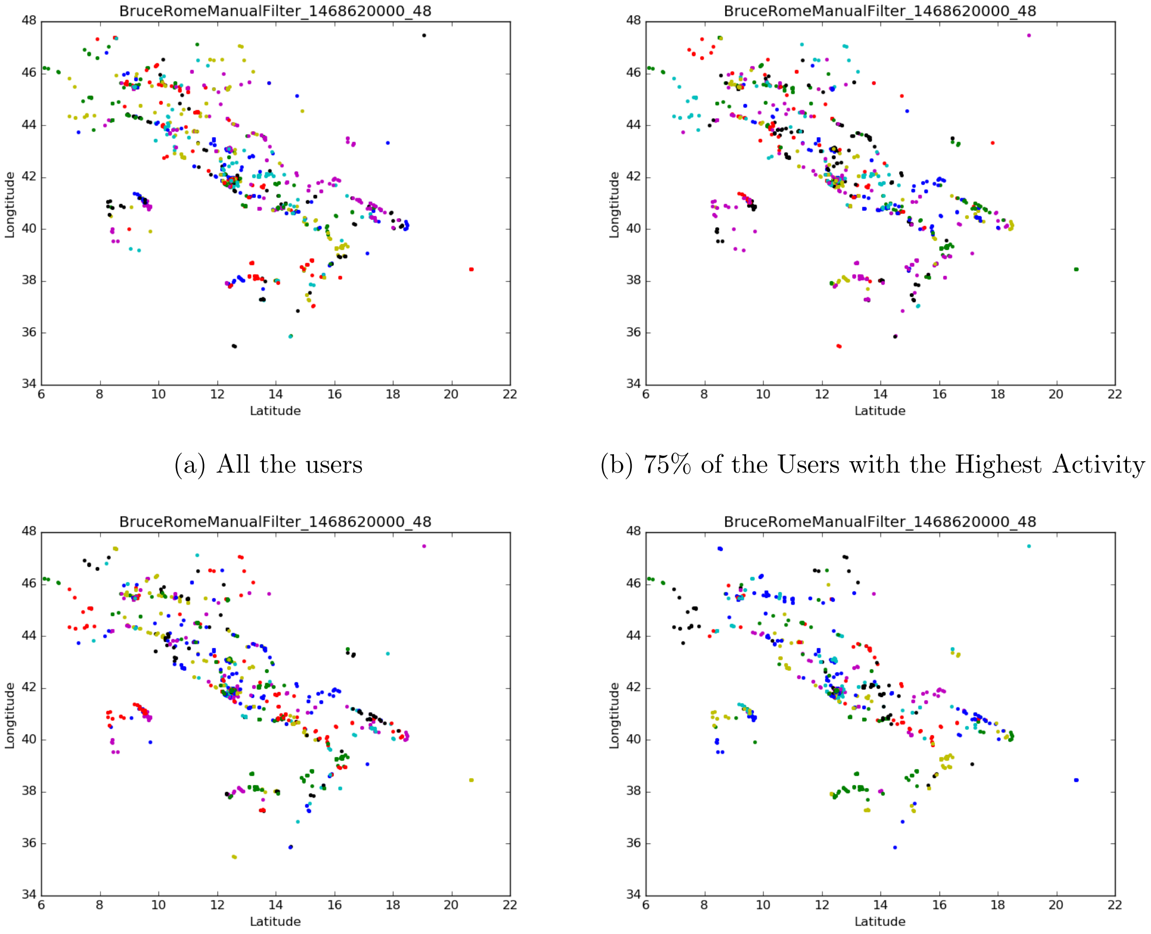
(a) All the users

(b) 75% of the Users with the Highest Activity

(c) 50% of the Users with the Highest Activity

(d) 25% of the Users with the Highest Activity

Figure 5.1: *Concert*1, window of 24 hours (67 users)



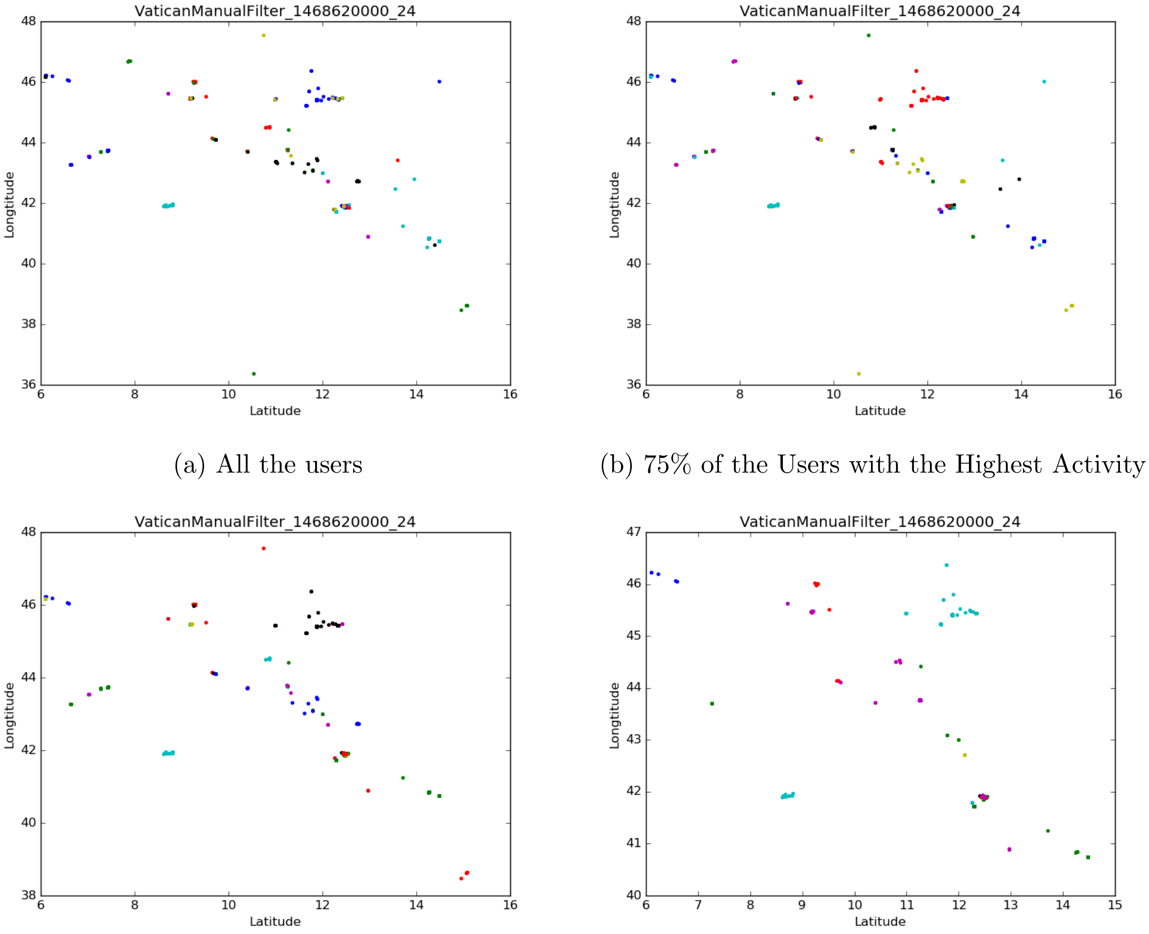
(a) All the users

(b) 75% of the Users with the Highest Activity

(c) 50% of the Users with the Highest Activity

(d) 25% of the Users with the Highest Activity

Figure 5.2: *Concert*1, window of 48 hours (144 users)



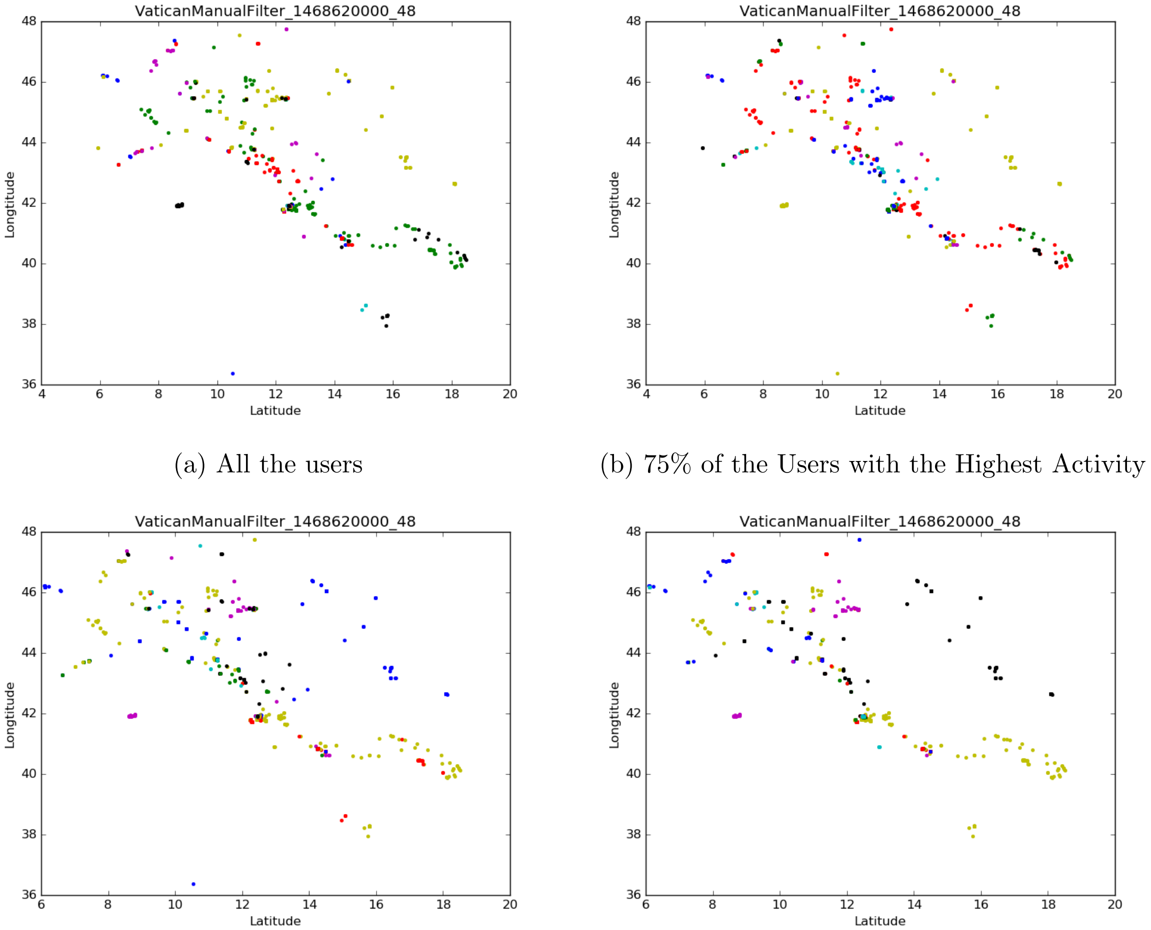
(a) All the users

(b) 75% of the Users with the Highest Activity

(c) 50% of the Users with the Highest Activity

(d) 25% of the Users with the Highest Activity

Figure 5.3: Vatican Visitors, window of 24 hours (48 users)



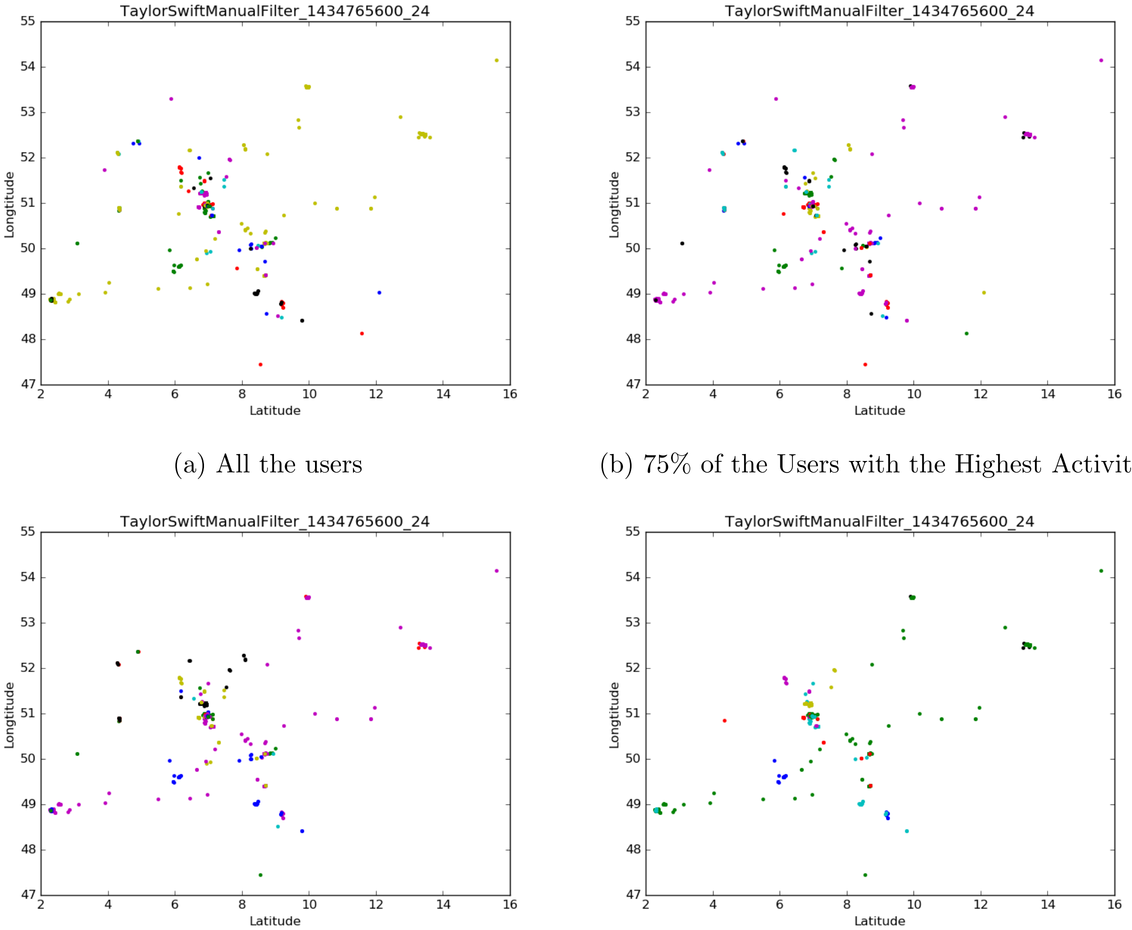
(a) All the users

(b) 75% of the Users with the Highest Activity

(c) 50% of the Users with the Highest Activity

(d) 25% of the Users with the Highest Activity

Figure 5.4: Vatican Visitors, window of 48 hours (91 users)



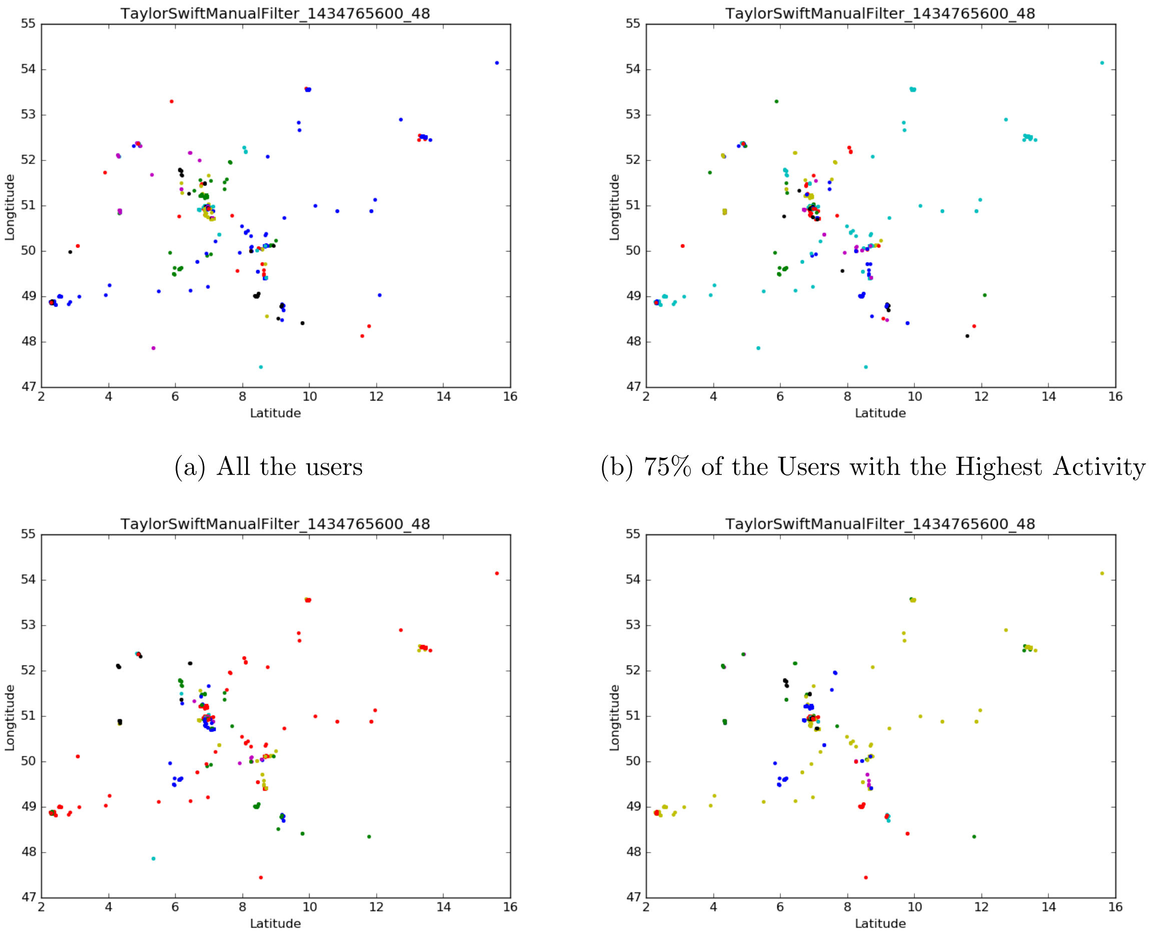
(a) All the users

(b) 75% of the Users with the Highest Activity

(c) 50% of the Users with the Highest Activity

(d) 25% of the Users with the Highest Activity

Figure 5.5: *Concert*2, window of 24 hours (44 users)



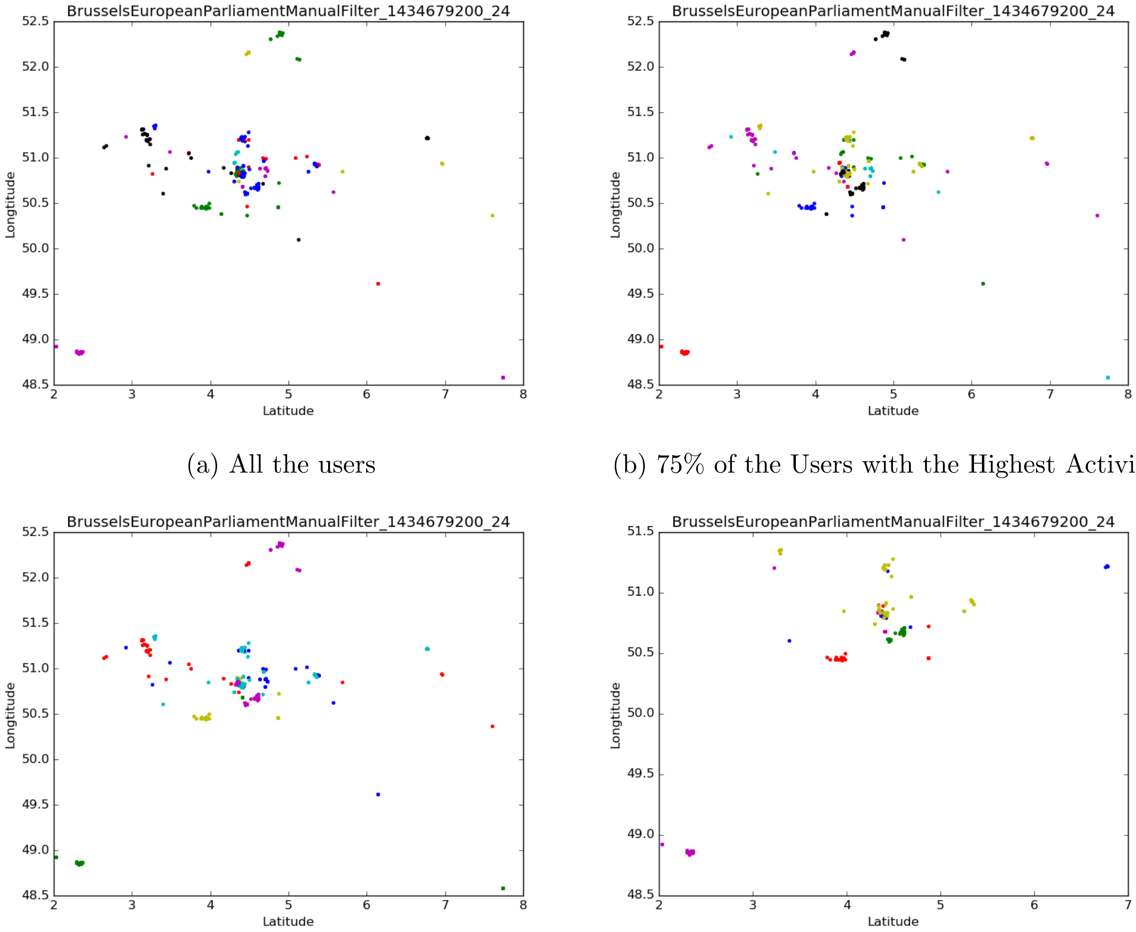
(a) All the users

(b) 75% of the Users with the Highest Activity

(c) 50% of the Users with the Highest Activity

(d) 25% of the Users with the Highest Activity

Figure 5.6: *Concert*2, window of 48 hours (59 users)



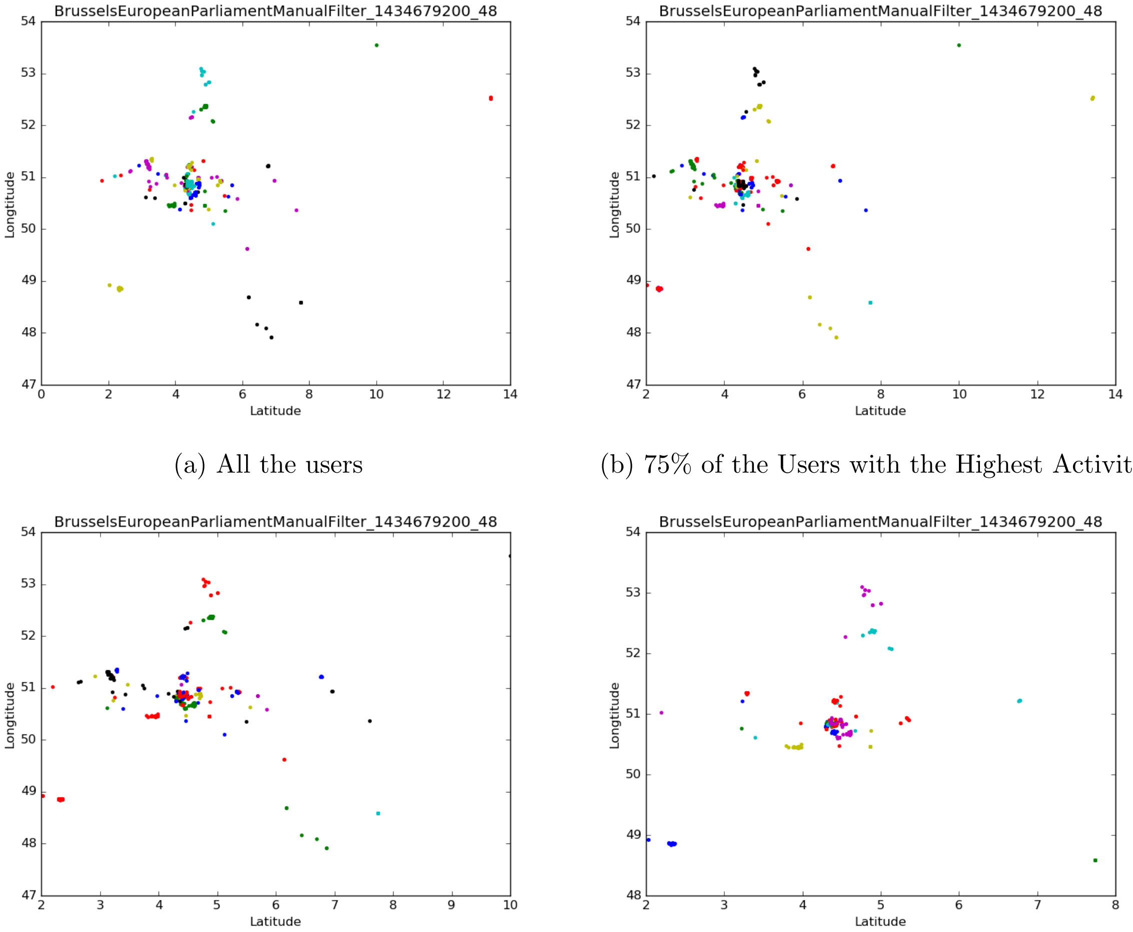
(a) All the users

(b) 75% of the Users with the Highest Activity

(c) 50% of the Users with the Highest Activity

(d) 25% of the Users with the Highest Activity

Figure 5.7: Parliament Visitors Manually Filtered, window of 24 hours (24 users)



(a) All the users

(b) 75% of the Users with the Highest Activity

(c) 50% of the Users with the Highest Activity

(d) 25% of the Users with the Highest Activity

Figure 5.8: Parliament Visitors Manually Filtered, window of 48 hours (46 users)

4. 25% of the number of the users who attended the *Concert*1

Having extracted the appropriate number of the users to be used, we experimented with the cases of the

1. n Random people from the *Concert*1
2. n Random people from Italy/Rome
3. Top n users from the *Concert*1
4. Top n users from Italy/Rome

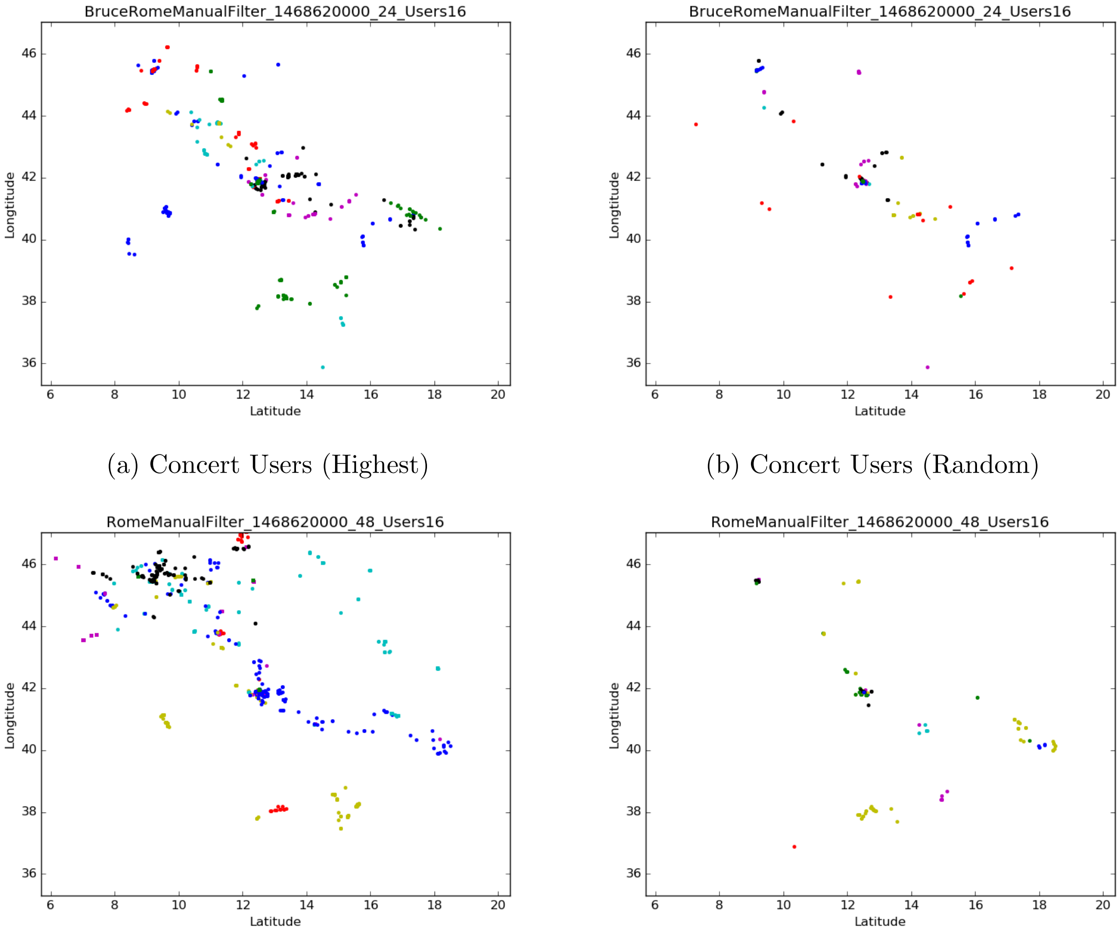
Impressively, after having analyzed the user activity, we found out that there is a non-spam user with 26756 tweets that exchanges messages having his location identification "on".

Rome Visitors Compared to *Concert*1 Attendees

In this part, we present the plots with the comparison of the locations between the *n* Random and *n* Top users from the *Concert*1 and Rome.

We initially use the 25% of the volume of the users who attended the *Concert*1, where n equals to 16 users. In Figure 5.9 we present the locations these 16 most active or random users from Rome or *Concert*1 appeared. As we can see, on the contrary to the plots presented at the beginning of the Section, although there are some routes can be assumed, the representation of both the routes and the map of Italy is not good. When increasing the number of the users to 33 (i.e. 50% of the volume of the users posted geotagged post from the location of the concert), both the representation of the routes and the map of Italy is much more accurate. The depiction of the 33 users' location is depicted in Figure 5.10. If we increase the volume of the users more, to 50 (75% of the users attended the concert) or 67 (100% of the users attended the concert), the representation of the map and the routes become even more clear. The plots of these two cases are depicted in Figures 5.11 and 5.12 respectively.

After further analyzing the distribution of the location of the users, we found out that the distribution of the locations of the users who attended *Concert*1 is much higher compared to those who posted geotagged post from Rome. This strengthens the assumption we previously did, that the users travel from other locations in order to attend a unique event such as a concert.



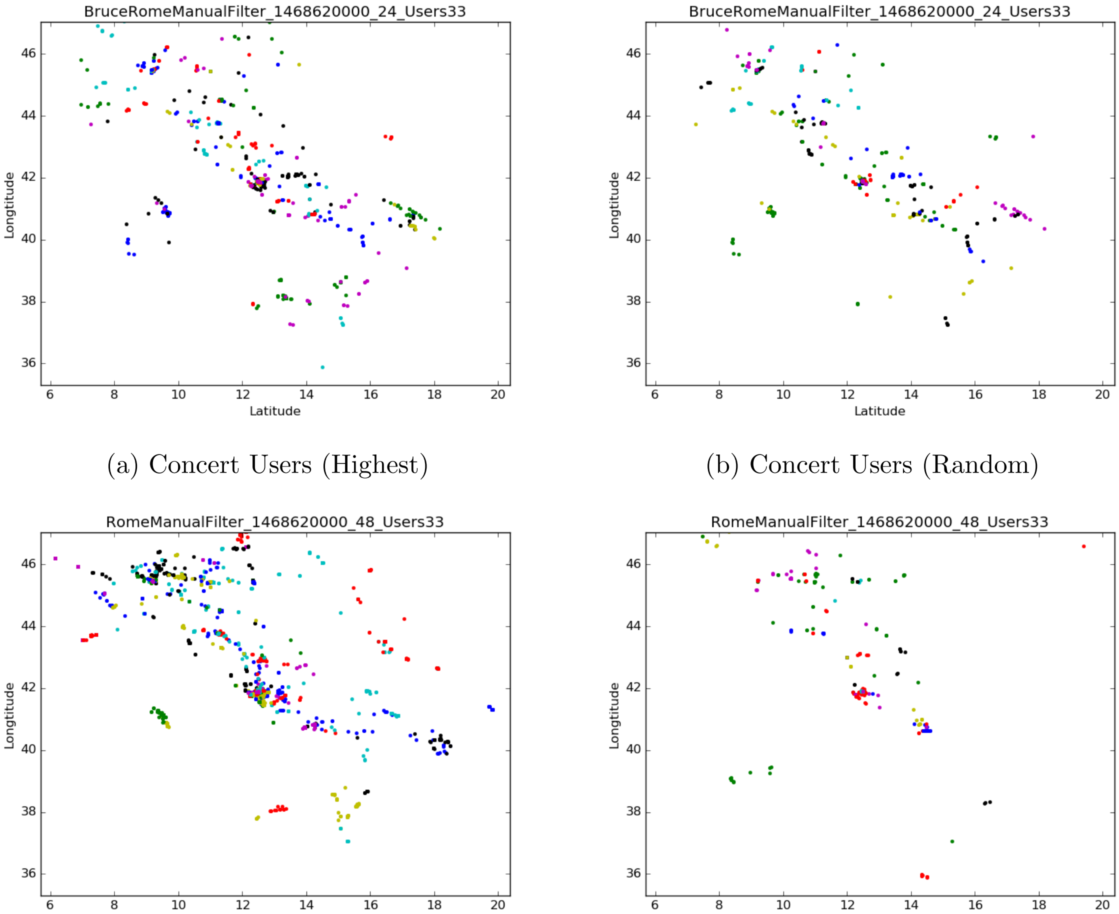
(a) Concert Users (Highest)

(b) Concert Users (Random)

(c) Rome Visitors (Highest)

(d) Rome Visitors (Rome)

Figure 5.9: *Concert*1 and Rome Visitors (Random 16 VS Top 16)



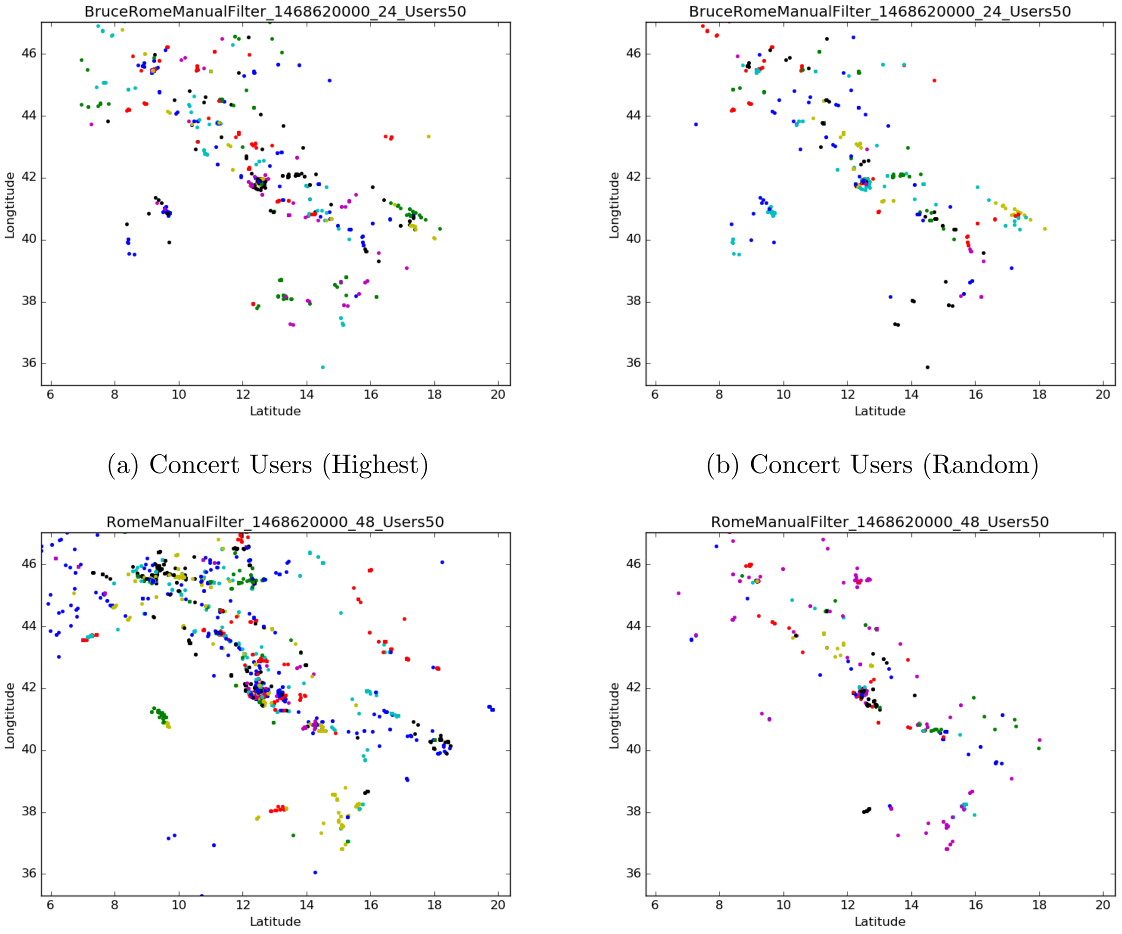
(a) Concert Users (Highest)

(b) Concert Users (Random)

(c) Rome Visitors (Highest)

(d) Rome Visitors (Rome)

Figure 5.10: *Concert*1 and Rome Visitors (Random 33 VS Top 33)



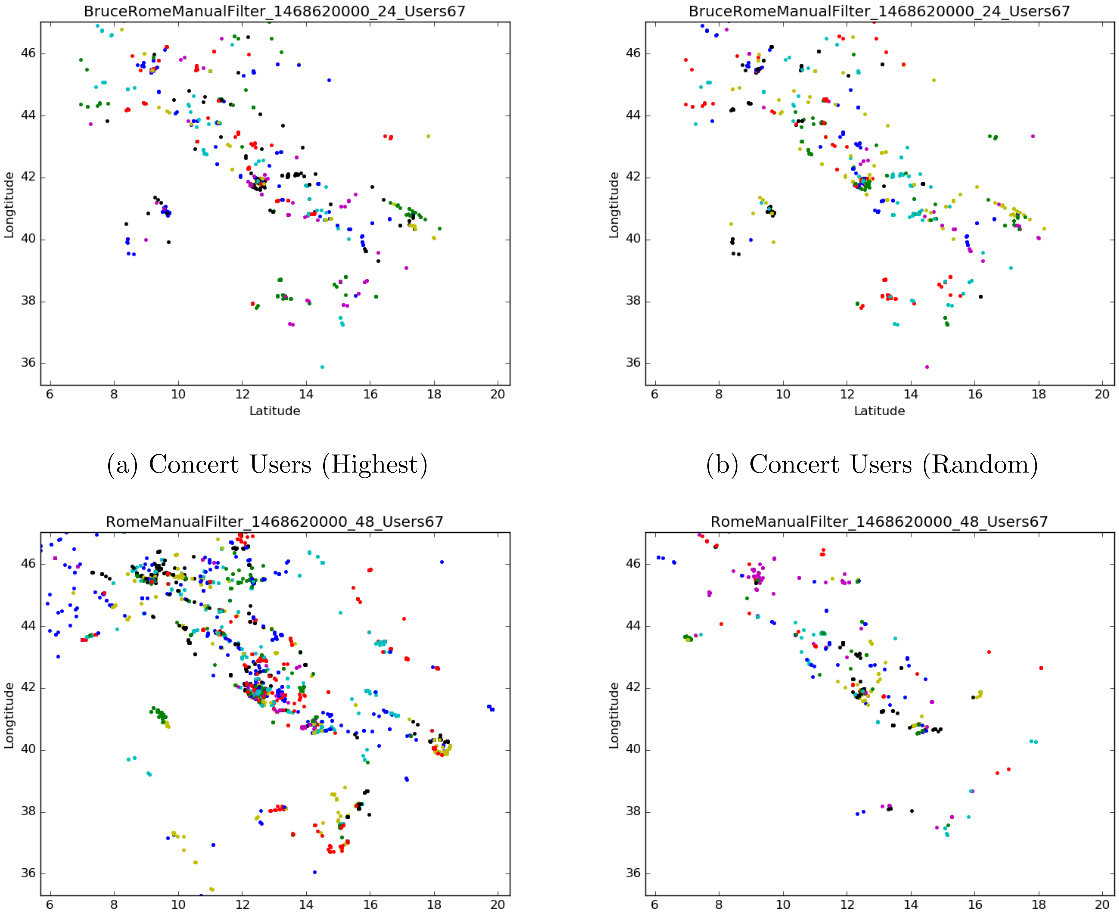
(a) Concert Users (Highest)

(b) Concert Users (Random)

(c) Rome Visitors (Highest)

(d) Rome Visitors (Rome)

Figure 5.11: *Concert*1 and Rome Visitors (Random 50 VS Top 50)



(a) Concert Users (Highest)

(b) Concert Users (Random)

(c) Rome Visitors (Highest)

(d) Rome Visitors (Rome)

Figure 5.12: *Concert*1 and Rome Visitors (Random 67 VS Top 67)

Italy

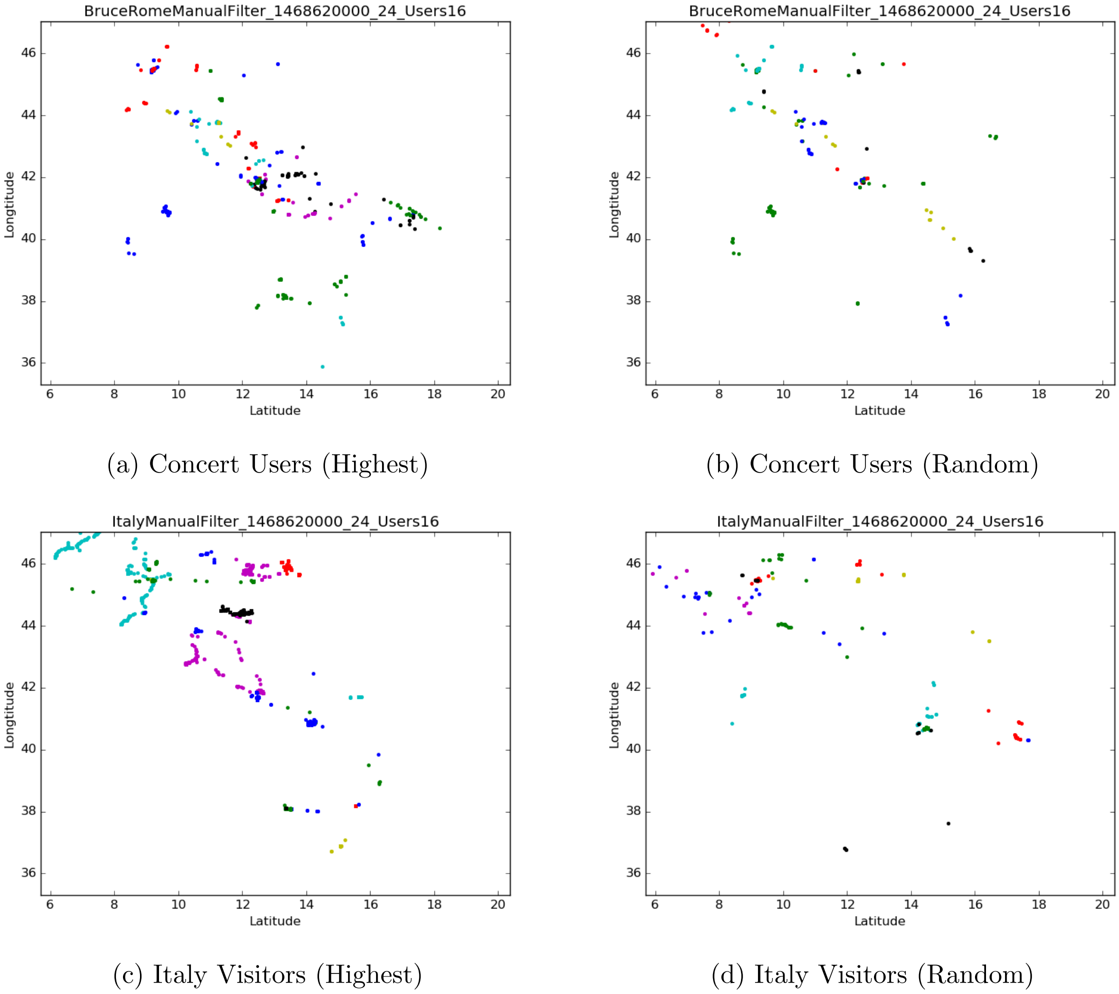
Having compared the activity and the location between the users of Rome and those of *Concert*1, we wanted to compare the *n* Random and n Top users from the *Concert*1 and Italy. Similarly to the case of the comparison of the two groups of users in the case of Rome and *Concert*1, when using the (either random or most active) 25% of the volume users who attended the concert, the representation of the map of Italy are not clear. On the other hand, some routes are depicted clearly, but this relies on the movement of the top users (Figure 5.13c). The case that we use the 25% of the users (16 users) is depicted in Figure 5.13. The representation of the routes become more clear when increasing the number of the most active users of Italy to 33 (50%), 50 (75%) and 67 (100%). Regardless this increase of the number of the users, the map of Italy is still not as clear as it is in the case of the users who attended the concert. After further analyzing the locations of the users from Italy dataset, we find out that the distribution of the location is still much smaller than the one the users who attended the concert have. The plots of the cases we use 50%, 75% and 100% of the Top or Random users are depicted in Figures 5.14, 5.15 and 5.16 respectively.

5.4.3 Cumulative Distribution Function and Movement

In this part, we investigate the cumulative distribution function and the movement of the users who attended *Concert*1 and those who didn't.

As we can see in Figure 5.17, the comparison between the activity of the users who attended the concert differs from the one of the users of Italy. More precisely, the percentage of the users who attended the concert and has a unique tweet, is double compared to the percentage of the users who were located in Italy but not in the concert area. Furthermore, we notice that the cumulative distribution function (CDF) of the users who attended the concert is very similar to those who visited Vatican, while the same happens with the users who were visiting Rome. After manually checking the tweets of the users who were at Vatican or Rome in general, we found out that the posts generated by the users who had posted only a few geotagged tweets, had been posted from unique locations such as Vatican, Colosseum or other historical monuments of Rome.

Furthermore, we compared the movement of the users who attended the concerts and those who did not. We found out that the mean difference of the maximum and minimum latitude and longitude that the concert users appeared is 301 km and 307 km respectively, while the median is 247 km in both dimensions. In the case of the users who were located in Italy but not at the concert the mean of the latitude and longitude difference is 272 km and 326 km while the median is 181 km and 231 km respectively. Regarding the users



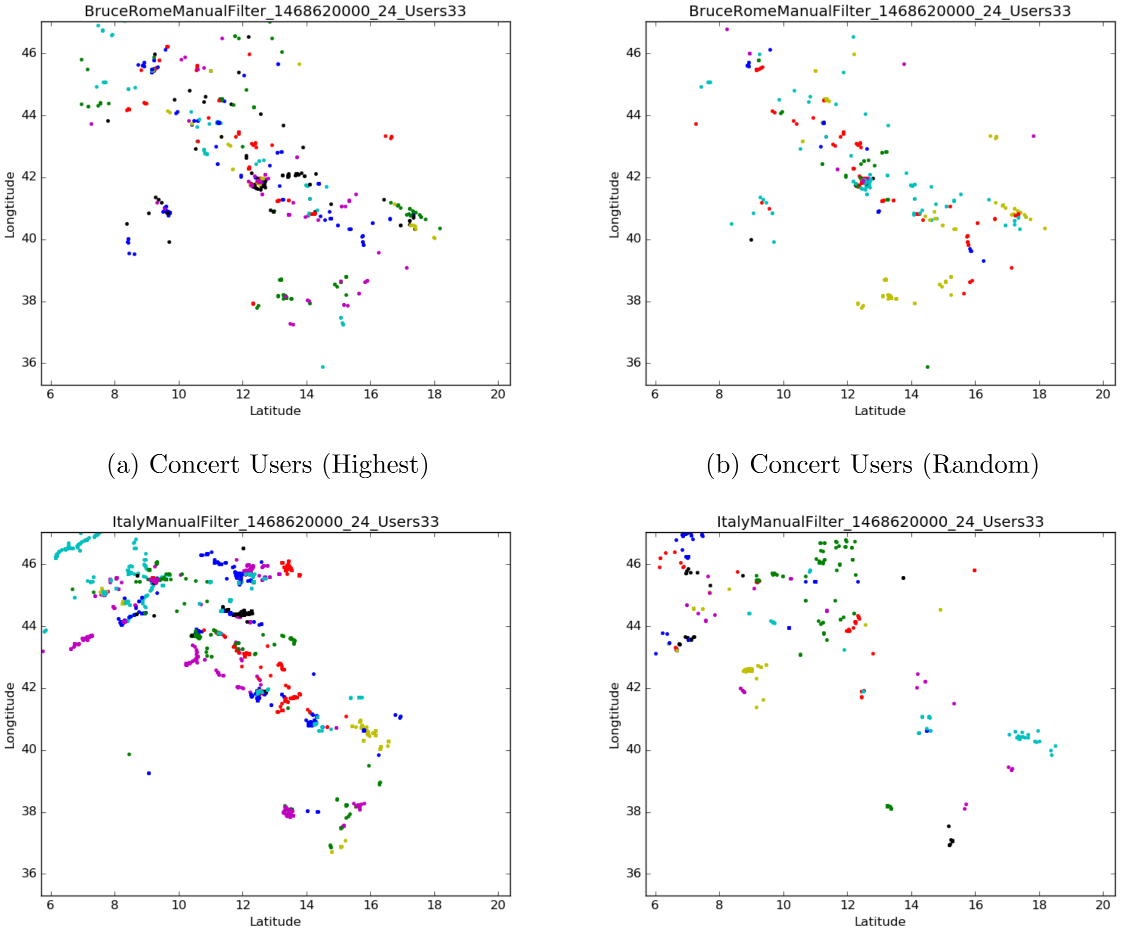
(a) Concert Users (Highest)

(b) Concert Users (Random)

(c) Italy Visitors (Highest)

(d) Italy Visitors (Random)

Figure 5.13: *Concert*1 and Italy Visitors (Random 16 VS Top 16)



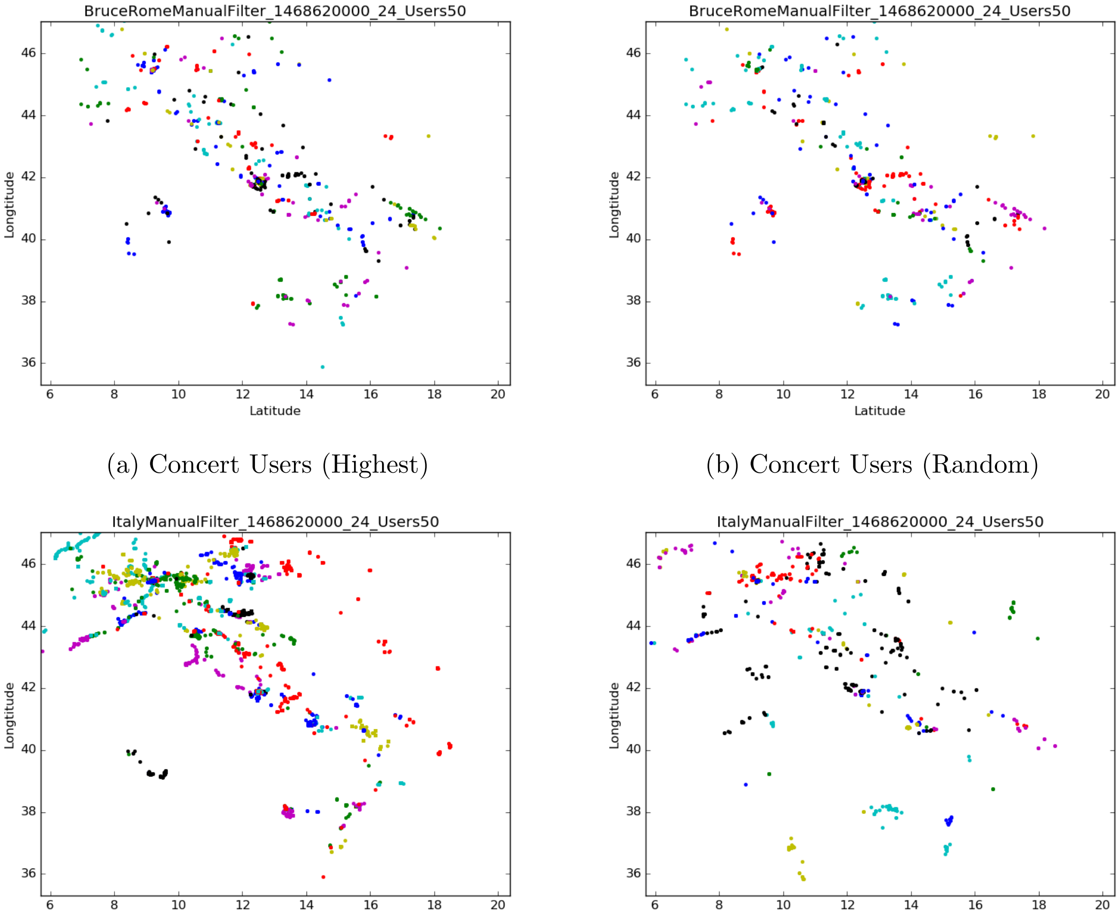
(a) Concert Users (Highest)

(b) Concert Users (Random)

(c) Italy Visitors (Highest)

(d) Italy Visitors (Random)

Figure 5.14: *Concert*1 and Italy Visitors (Random 33 VS Top 33)



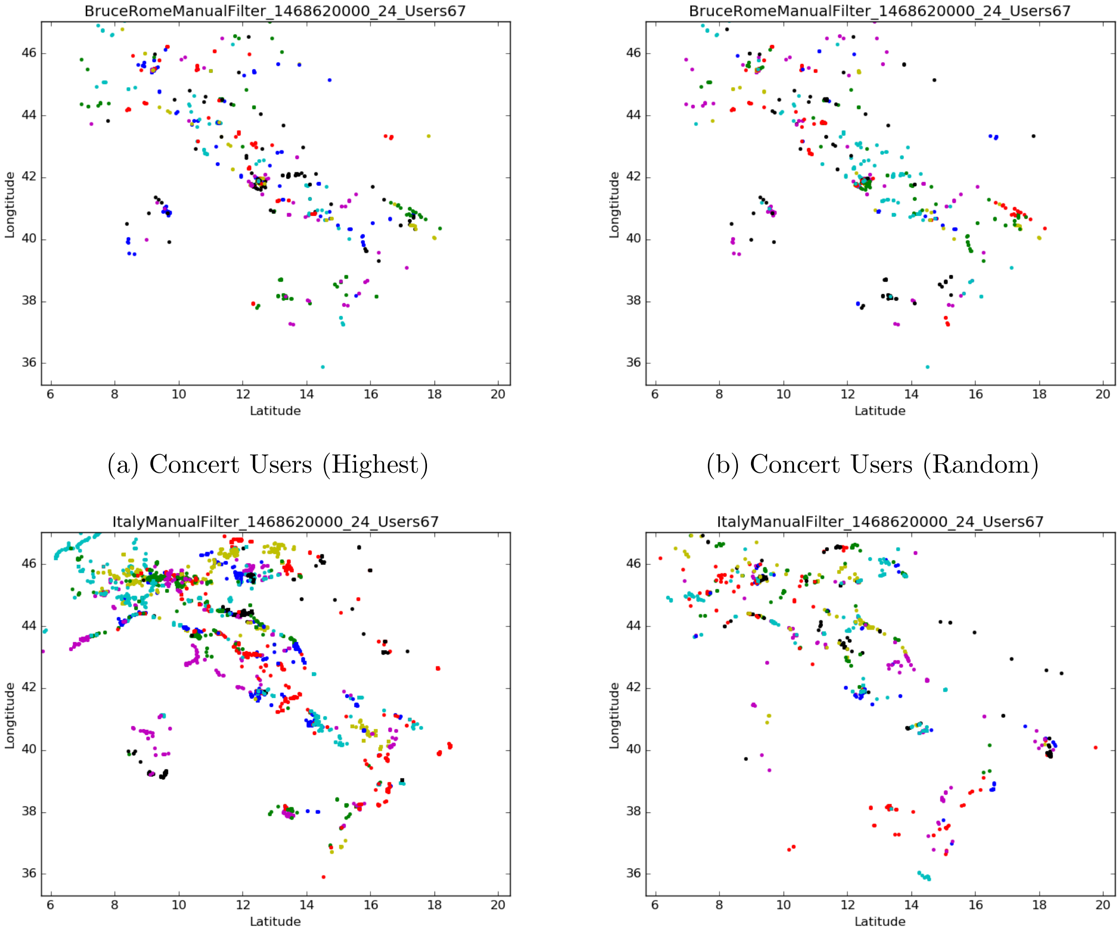
(a) Concert Users (Highest)

(b) Concert Users (Random)

(c) Italy Visitors (Highest)

(d) Italy Visitors (Random)

Figure 5.15: *Concert*1 and Italy Visitors (Random 50 VS Top 50)



(a) Concert Users (Highest)

(b) Concert Users (Random)

(c) Italy Visitors (Highest)

(d) Italy Visitors (Random)

Figure 5.16: *Concert*1 and Italy Visitors (Random 67 VS Top 67)

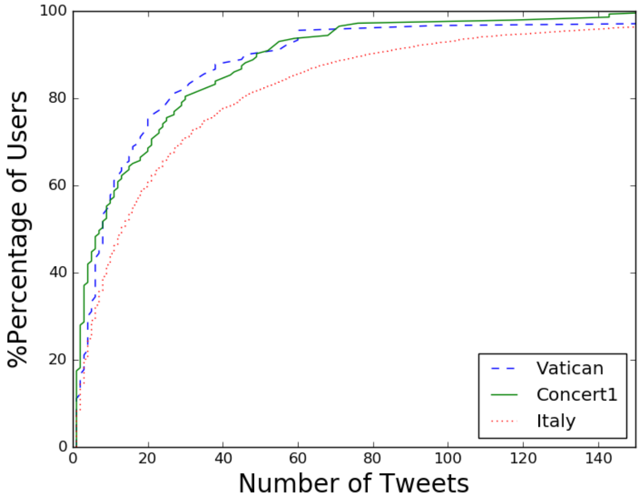


Figure 5.17: CDF: Comparison of Number of Tweets

of Rome who haven't attended the concert, these numbers become 273 km and 294 km when it's the mean and 209 km and 228 km when is the median. Finally, regarding the users who visited Vatican, the numbers get reduced to 261 km and 255 km when it's the mean and 218 km and 163 km when it's the median km difference.

The difference at the locations the users of each group appeared, constitutes one more hint, reinforcing our initial hypothesis that users who attend important unique events, such as concerts, tend to travel from other locations in order to attend the event. Furthermore, these numbers combined with the distribution of the locations the attendees of a concert appear, indicates that the users who attend unique events also tend to travel more.

5.5 Summary

In this chapter, we present a framework that examines the differences of the activity and mobility patterns of people that attend or visit an important/unique event or location. Our experimental evaluation indicates that the users are willing to travel from far locations in order to attend a unique event. Furthermore, we investigate the volume of the users needed in order to identify main routes and locations that attract people, coming to the interesting conclusion that less than 35 users who attended a unique event are enough to allow us identify main routes and shapes of regions or countries. Finally, our experimental evaluation shows that user presence in special events or locations (such as an important touristic attraction, or a major concert) affects the normal activity patterns, increasing the likelihood of making geotagged posts.

Chapter 6

Identifying Abnormal Spatio-Temporal Patterns in Mobile Phone Usage Data

6.1 Introduction

The availability of datasets coming from the telecommunications industry, and specifically those relevant to the use of mobile phones, are helping to conduct studies on the patterns that appear at large scales, and to better understand social behaviors. This study aims at developing methods for enabling the extraction and characterization of normal behavior patterns, and the identification of exceptional, or divergent behaviors. We study call activity to classify the observed behaviors that exhibit similar characteristics, and we analyze and characterize the anomalous behaviors. Moreover, we link the identified behaviors to important events (e.g., national and religious holidays) that took place in the same time period, and examine the interplay between the behaviors we observe and the nature of these events. The results of our work could be used for early identification of exceptional situations, monitoring the effects of important events in large areas, urban and transportation planning, and others.

6.2 Problem description

In this thesis, we concentrate on the identification and investigation of the anomalous behaviors discovered in some cell-towers, and the examination of the reasons that could cause such a behavior. The second problem we tackle is to characterize the way that a social event affects to the calling activity of a region, or to the entire country in general.

Finally, we analyze the social response to some major events, and investigate how different events affect the mobility of users.

Problem 1: Given a set of Call Detail Records $C_{t_j}^{l_1}, ..., C_{t_j}^{l_i}$, $t_1 \leq t_j \leq t_2$, describing the calling or SMS activity of a cell-tower, where l_i is the location the cell-tower is located and t_j is the time interval during which we recorded the activity, we wish to identify the location l with abnormal activity.

The timestamps t_1 and t_2 represent the start and end times, respectively, of the time interval we are interested in.

Problem 2: Given a set of Call Detail Records $C_{t_j}^{l_1}, ..., C_{t_j}^{l_i}$, $t_1 \leq t_j \leq t_2$, describing the calling or SMS activity of a cell-tower, where l_i is the location the cell-tower is located and t_j is the time interval during which we recorded the activity, we wish to identify the location t with abnormal activity.

In the context of this work, we concentrate on finding the spatial or temporal divergences and examine the reason that caused them, using a dataset containing important events.

6.3 Methodology

D4D [1], providing a dataset containing Call Detail Records. In this part, we present the necessary preprocessing that we performed before applying our techniques and the method we developed in order to analyze the calling activity of the entire country of Ivory Coast, creating clusters and extracting usage patterns. Furthermore, our method allows us to identify activities in specific regions or even in the entire country that are not normal.

Aggregate Communication Between Cell-Towers Data

The dataset we used for the development of our method contains data about the number of calls and their total duration. The data was grouped by their origin and their destination cell-tower. Furthermore the dataset contains timestamps about the time that the calls were initialized, but not the time that they were terminated.

6.3.1 Preprocessing of Datasets

The dataset was structured in such a way that an immediate analysis was not possible in order to make clear conclusions about the changes of the calling activity. Before starting the development of our methods, we had to manipulate the data in a way that we would

[1] Data 4 Development (D4D) is a competition launched by Orange of Ivory Coast

keep just the most useful (for us) data and turn them in a more usable form. In the following part of this section, we describe these preprocessing steps.

Useful Variables

The methods used in the first dataset have only two types of values. The first is the hourly number of calls for each cell-tower, and the second is the total duration of these calls. Due to the volume of the data, we decided to aggregate the 24 hourly values that each cell-tower has for each day into a single daily value. Even though this aggregation leads to some information loss, it allows us to perform an initial fast analysis, which can subsequently be refined, by using the hourly data values, for the cases in which we detect an abnormal behavior.

We note that many cell-towers did not contain 24 values for every day in the dataset (due to the missing data problem we discussed earlier). Moreover, some cell-towers did not have values for each day of the period that the available dataset was produced, but this did not cause a problem for our analysis.

Apart from the two variables provided in the dataset, we derived and used a third variable that helped us to perform our analysis. This variable is the "duration per call" (dpc) that can be extracted by the division of the daily duration of calls by the number of calls, for each cell-tower. The values for this variable were calculated according to Equation 6.1.

$$dpc_{i,j} = \frac{total\_duration_{i,j}}{number\_of\_calls_{i,j}}, i, j \in N \tag{6.1}$$

Normalizing the Data

There are some cell-towers that are in urban areas and some others that are in areas that don't have many citizens. This has as a result that the first group of cell-towers have a continuously high activity, with respect to both the number of calls and their duration. Furthermore there are some days, like public holidays, that have more calls than the days when there is not any special event. These two factors do not allow us to cluster the data because the days or the cell-tower that have this overhead would always be reported as outliers.

In order to eliminate this problem we normalize the data by using z-normalization. In statistics, the z-normalization ensures that all elements of the input vector are transformed into the output vector whose mean-μ is 0, while the standard deviation-σ (and variance) is 1. For this transformation, we used Equation 6.2.

$$x'_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j}, i, j \in N \tag{6.2}$$

We normalize the values in two ways. First we normalize by day in order to have normalized data with respect to each individual day. This can be achieved by finding the mean value and the standard deviation for each day and then compute the new value according to each day. This kind of normalization helps us to identify patterns across days. In this case we use Equation 6.2, where i is the cell id and j is the day.

In addition, we normalize by the cell-tower, using each individual cell-tower's mean value and standard deviation. This action helps us to identify patterns for the cell-towers. In this case we use Equation 6.1 again, but in contrast to the previous case, i is the day and j is the cell id.

6.3.2 Analysis of Anomalous Behavior

In order to achieve the extraction of either the temporal or the spatial outliers, we developed a set of methods. We describe these methods used after the preprocessing, at the following subsections.

Identifying Outliers

After the normalization of the data we have to compare the values with respect to the day or the cell-tower. In order to compare these values we calculate the mean and the standard deviation for each cell-tower or for each day, depending on the analysis that we intend to do. Furthermore, we calculate the difference of each point from its neighbors and from the mean. If we have a point A that is much farther away from the mean than a point B that is the closest to A and between A and the mean, then the point A is marked as an outlier. In practice, we implement a simple density-based clustering algorithm to create one main cluster that contains the mean value and to separate this cluster from the points that are much different than the cluster. Such an example is the plot depicted in Figure 6.1.

In this figure, we have an analysis of the data points clustered by day after we normalized by the day. As we can see there are two days, the day 66 and 67, that have some cell-towers whose points are much farther from the mean than the rest of the points, creating a gap between them and the rest of the cluster. By analyzing these outliers, and after having set a threshold[2] of '3.5', we found that the cell-towers that have these values, never had such a calling activity during the rest of the period that covers our dataset.

The algorithm that implements our method is shown in Algorithm 11.

[2] We set this threshold-radius manually after observing the data.

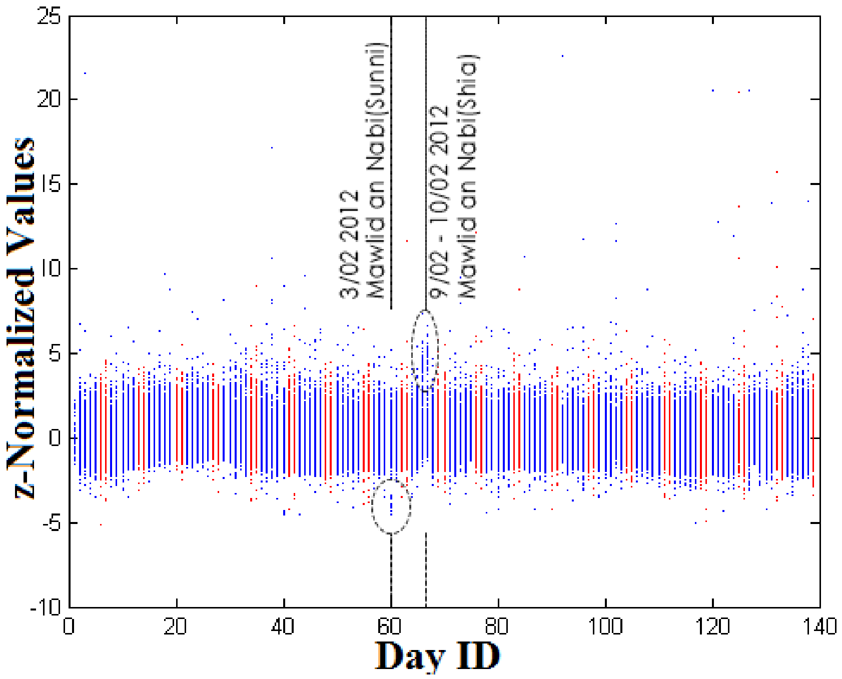


Figure 6.1: Daily plot for dpc normalized by day.

Algorithm 11 Grouping By Distances

```
procedure GroupByDistances(threshold)
    $x'_{i,j} \leftarrow NormalizedValues$
    for $i = 1 \rightarrow MaxID$ do    ▷ The MaxID is either the maximum ID of the cell-tower or the maximum ID for the days, depending on the analysis we intend to do. In most of the cases the day.
        $Distances \leftarrow allthedistances$    ▷ between each point that belongs to i and i's mean
        $array \leftarrow sortthedataaccordingtothedistances$
    for $i = 1 \rightarrow MaxID$ do
        for all $points \in i$ do
            if $distance \geq closerpoint$ then    ▷ the point that is closest and between the examined point and the mean
                while $(NotTheEnd)$ do
                    while $(NoPointWithGreaterDistance)$ do
                        $checkNextPoint()$
                        if it is close to the previous point then    ▷ they propably form a sub-cluster return $point$    ▷ as a possible outlier
```

Identifying Outliers Using the Standard Deviation

A second method that we used to identify the outliers is the comparison of the standard deviations. This method can be mainly applied on the daily values because each day has more or less the same features. More specifically each day has (almost) the same number of values and each value derives from a cell-tower that is every day at the same longitude and latitude. The only difference is that if an event is local then it will be hard to detect using the normalization by cell-tower. This results in the creation of datasets that have some steady main features plus some features that change and allow us to analyze them.

Following this method, we look at the standard deviation for each day and we compare it with the standard deviations of the 12 adjacent days, 6 before the day under examination and 6 after. This helps us to draw a conclusion on whether the calling pattern is more or less the same for this day as it should be, in respect to the period that we analyze. In case the standard deviation is not similar to the majority of the compared days, we can come to the conclusion that some event, such as a public holiday has taken place.

The pseudocode for this technique is shown in Algorithm 12.

Algorithm 12 Grouping By The Standard Deviation

```
procedure GroupByStd(threshold)
    $x'_{i,j} \leftarrow NormalizedValues$
    for $j = 1 \rightarrow MaxID$ do    ▷ The MaxID is either the maximum ID of the cell-tower or the maximum ID for the days, depending on the analysis we intend to do. In most of the cases the day.
        $Difference \leftarrow 0$
        $Similar \leftarrow 0$
        for $k = (j-6) \rightarrow j+13$ do ▷ Compare with the 6 previous days and the 6 days after
            if $i \neq j$ then
                if $(\sigma[i] \neq (threshold * \sigma[j]))||(\sigma[i] \leq (threshold * \sigma[j]))$ then
                    $Difference \leftarrow Difference + 1$
                else
                    $similar \leftarrow similar + 1$
        if $Difference \geq 6$ then return $i$
```

Correlated Abnormal Behaviors

We have already analyzed the cases that a value is an outlier for a cell-tower, or for a day. The problem that rises is the importance and the weight that the value has in general. If, for example, cell-tower 1 has for one day 100 calls and for the next day again 100, this could be possibly a normal pattern according to the cell-tower whose values remain more or less stable. What happens, though, if the values for all the cell-towers apart from this one are increased during the second day? This means that cell-tower 1 is an outlier. If we perform only a normalization with respect to the cell tower this outlier could be lost.

In order to avoid this situation, we have to correlate the two normalized values. This can be achieved by the subtraction of the two normalized values, and then look for outliers in this new space. This correlation can be achieved by using Equation 6.3, where x'_1 and x'_2 are the values derived from the two normalization procedures.

$$weight_{i,j} = x'_1 - x'_2, i, j \in N \tag{6.3}$$

6.4 Experimental Evaluation

In this Section we initially describe our dataset and afterwards present the results we had after applying our methods on it.

6.4.1 Description of Datasets

The dataset provided to us describes the aggregated communication between cell-towers. The data describe the activity for the whole country of Ivory Coast and were collected from December 2011 to April 2012 (five-month period) and consists of 175.645.538 rows.

In preprocessing the data we observed that the volume of missing data was rather large, which made it difficult to make accurate predictions or connections with the events, during the subsequent analysis phases. This problem was created due to some technical problems, and as a result led to the loss of the origin, or the destination cell-tower id. The missing cell-towers were recorded as '-1'. More precisely, just for the first dataset, the amount of missing data was too big that for each cell-tower we had an average of 143.162 records, while at the same time the number of records for cell-tower '-1' were 1.846.084.

At this point we have to mention that even though the cell-tower ids range from 1 to 1238, there are some ids that don't belong to a cell-tower. Furthermore, there are some cell-towers that do not have any record during the whole five-month period. As a result we have just 1214 cell-towers with records plus one, the cell-tower '-1' that represents the missing data.

Events Data

In order to collect some interesting events that took part during the five-months period covered by our sample, we used the Google Search Engine and we manually extracted the most important events related to Ivory Coast. Examples of such events are public holidays, important festivals, sport events, concert shows, and news that could change the activity of a user.

The extracted events refer only to the time period between the beginning of December 2011 and the end of April 2012. These events are listed in Table 6.1, and include events of both both regional and national importance.

Date	Location	Event	Event type
Dec 25, 2011	Ivory Coast	Christmas Day	public holiday
Jan 01, 2012	Ivory Coast	New Year's Day	public holiday
Feb 05, 2012	Ivory Coast	Day after the Prophet's Birthday (Maouioud)	public holiday
Feb 13, 2012	Ivory Coast	Post African Cup of Nations Recovery	public holiday
Apr 09, 2012	Ivory Coast	Easter Monday	public holiday
Feb 05, 2012	Ivory Coast	Mouloud	public holiday
Feb 22, 2012	Ivory Coast	Ash Wednesday	public festival
Jan 14, 2012	Ivory Coast	Arbeen Iman Hussain	public festival
Jan 8, 2012	Ivory Coast	Baptism of the Losd Jesus	public festival
Mar 25- Apr 1, 2012	Bouake	Carnaval	public festival
Apr 1- May 1, 2012	Ivory Coast	Fete du Dipri	public festival
Apr 6, 2012	Ivory Coast	Good Friday	public festival
Feb 9, 2012	Ivory Coast	Mawlid an Nabi (Shia)	public festival
Feb 4, 2012	Ivory Coast	Mawlid an Nabi (Sunni)	public festival
Feb 5, 2012	Ivory Coast	Yam	public festival
Dec 7, 2011	Ivory Coast	Anniversary of the death of Felix Houphouet Boigny	public festival
Apr 13-14, 2012	Abidjan	Assine Fashion Days in Cote D'Ivoire	show concert
Apr 1-4, 2012	Yamoussoukro	Education international 22nd congress	conference meeting
Apr 25, 2012	Sakre	Violence attack in Sakre	emergency event
Dec 17-18, 2011	Yale	Violence	emergency event
Jan 7, 2012	Abidjan	Hilary Clinton's visit	news event
Jan 7-8, 2012	Abidjan	Kofi Annan's visit	news event
Mar 12-13, 2012	Abidjan	Election of National Assembly President and Prime Minister	news event
Dec 11, 2011	Abidjan	New parliament election	news event
Jan 30, 2012 19-20	Ivory Coast	ACNF 2012 match vs Angola	sport
Jan 26, 2012 20-21	IvoryCoast	ACNF 2012 match vs Burkino Faso	sport
Jan 22, 2012 17-18	IvoryCoast	ACNF 2012 match vs Sudan	sport
Feb 4, 2012 20-21	IvoryCoast	ACNF 2012 match vs Equatorial Gulnea	sport
Feb 8, 2012 20-21	IvoryCoast	ACNF 2012 match vs Mall	sport
Feb 12, 2012 20:30-21:30	IvoryCoast	ACNF 2012 final match vs Zambia	sport

Table 6.1: List of important regional and national events in Ivory Coast, for the time period between December 2011 and April 2012.

6.4.2 Anomalous behaviors

In this section we present some results after we applied the method described in section 6.3 on the first dataset.

The first step is to compute the duration per call (dpc) at each cell-tower. The next step is to normalize the data by day and by the cell-tower, as described in 6.3.1. After the normalization step we have six values, two for each initial variable, the daily number of calls (nb), the daily duration of the calls (dur) and the dpc for each cell-tower.

We cluster normalized data in the way described in section 6.3.2. This has a result to identify behaviors that are not normal. Such type of behaviors we can see in Figure 6.1. In this figure we can see the dpc normalized by each day values. At the x axis we have the day id and at the y axis we have the normalized value of the dpc by day for each day. With red color we can see the weekends and with blue color the weekdays. This difference at the colors makes it easier for us to achieve even a visualized comparison between the values. As we can see almost all the days follow the same pattern, having values that are in a small range plus some values that are (unique) outliers. Investigating these outliers, we found that are mostly the same cell-towers. This allows us to consider it as a normal pattern for these specific cell-towers and we don't analyze them more.

Although most of the days in the sample follow the same pattern, there are some days like the days with ids 60, 66 and 67 that have a sub-cluster of outliers. Investigating these outliers we found out that, for the days 66 and 67, these cell-towers are only 36 and they are close to each other. Furthermore, we found out that the calling activity referring to the dpc for these cell-towers is unique for these days and they don't have such an activity for the rest of the five-month period. For the day 60 we have the same conclusions as we did with the days 66 and 67 with the difference that the outliers are negatives.

Finally we came to the conclusion that specific events or actions change the calling activity for these days for these specific cell-towers. You can see the cell-towers that are outliers for the days 66,67 and the day 60 in Figures 6.2,6.3 respectively.

One more fact that evaluated our conclusions is the analysis that we did for the number of the calls for each day when this number was normalized by the day. In order to achieve this type of analysis we used the method described in section 6.3.2. By analyzing these values we found out that just a single event could cause a difference on the calling activity for just a region(when it is a local event) or even for the whole country.

Such an example is depicted in Figure 6.4 where we can see the difference of the calling activity for the whole country during the Christmas and the new year event, the easter, the days that we have unique events such as festivals and the rest of the days. We depict with blue color the weekdays and with red color the weekends.

Finally we evaluate the method described in 6.3.2 by analyzing the duration of the

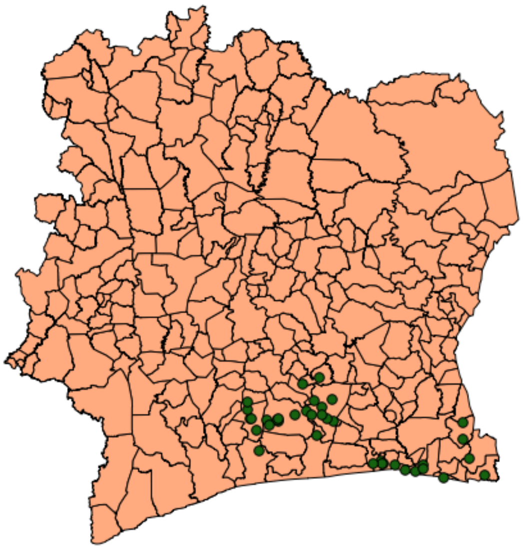


Figure 6.2: Cell-Towers Positive Outliers for 9-10 of February 2012

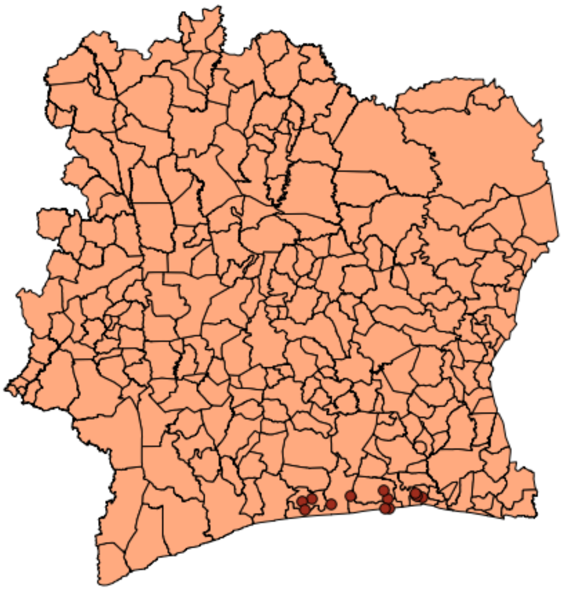


Figure 6.3: Cell-Towers Negative Outliers for 3 of February 2012

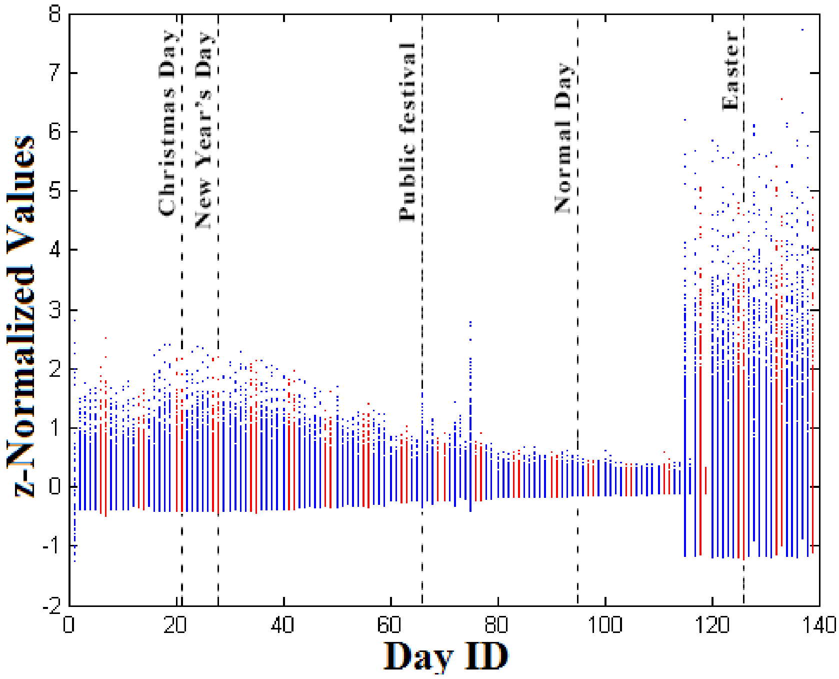


Figure 6.4: Number of calls for each day (normalized by day)

calls for each cell-tower while we have normalized it both by the day and the cell-tower. Subtracting the second value from the first we get a result that for cell-towers with ids 731 to 750 the weights for the weekends are mostly clustered at the positive values while the weights for the weekdays are clustered to the negative values. This makes it clear that these cell-towers have specific patterns for the weekdays and the weekends. In Figure 6.5 we have this analysis visualized. Again with blue color we have the weekdays and with red color the weekends. In x axis we have the cell-tower id and in y axis we have the normalized daily duration of calls.

6.5 Summary

In this chapter, we studied the call activity and mobility patterns, clustering the observed behaviors that exhibited similar characteristics, and characterizing the anomalous behaviors. We analyzed a Call Detail Record (CDR) dataset, containing (aggregated) information on the calls among mobile phones. Employing density-based algorithms and statistical analysis, we developed a framework that identifies abnormal locations, as well abnormal time intervals. The results of this work can be used for early identification of exceptional situations, monitoring the effects of important events in urban and trans-

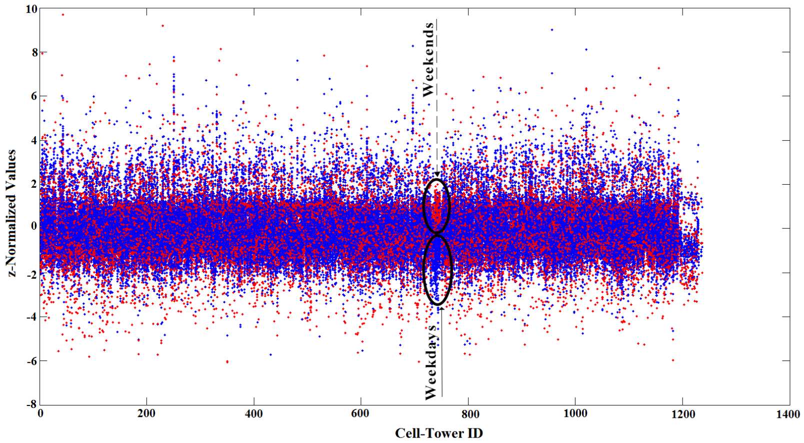


Figure 6.5: Weights For The Correlation of The Two Types of Normalized Values For The Duration (for each cell-tower)

portation planning, and others.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The development of social networks such as Twitter, Facebook and Google+ allow users to share their beliefs, feelings, or observations with their circles of friends. This phenomenon has been amplified by the proliferation and ubiquitous use of mobile devices, which nowadays offer a rich set of functionalities. For example, Twitter has more than 313 million users, 80% of which are users on mobile devices. One of the most important functions that mobile devices offer is that they provide to users the ability to share in real time (via social media platforms, such as the ones mentioned above) information about their lives and their activities, as well as their current locations. The importance of geolocalized information is underlined by the numerous applications across domains (e.g., targeted advertising, mobility recommendations, tourist applications, etc.) that use this information. Nevertheless, only a tiny percentage of social media data is geotagged (2% for Twitter), and increasing this percentage is an important and challenging problem. Moreover, information extracted from social media data can be complemented by the analysis of mobile phone usage data, in order to provide further insights on human activity patterns.

In this study, we studied the call activity and mobility patterns, clustering the observed behaviors that exhibited similar characteristics, and characterizing the anomalous behaviors. We analyzed a Call Detail Record (CDR) dataset, containing (aggregated) information on the calls among mobile phones. Employing density-based algorithms and statistical analysis, we developed a framework that identifies abnormal locations, as well abnormal time intervals. The results of my work can be used for early identification of exceptional situations, monitoring the effects of important events in urban and transportation planning, and others.

Subsequently, we focused on the problem of geolocalizing social media posts, and in particular tweets. Although there are a few studies that try to identify the location a tweet was posted from, they operate on a coarse-grained granularity such as a region, city, or zip-code. The target of our research was the development of algorithms for the identification of locations in the granularity of a city neighborhood. The challenges related to this task were to achieve high precision and recall, despite the significantly increased size of the search space, while maintaining low execution times. The approach we proposed builds topic models that are used for the location prediction, by creating representative vectors based on Tf-Idf and Logistic Regression. In addition, it uses algorithms based on Linear Regression and Pearson correlation in order to exploit the information on the tweet activity time-series of the different neighborhoods. In experiments with seven Italian cities, the proposed method achieved up to 89% precision with 17% recall (i.e., increasing the currently available geotagged tweets by 800%). The suite of algorithms we developed is now part of the TweeLoc system, which includes several different online visualizations: it depicts the (predicted) location of tweets on a geographic map, shows the most important keywords associated with these tweets, displays aggregated statistics for the activity in each neighborhood and how the activity is changing over time, highlights the neighborhoods with the largest increase/decrease in activity, and allows the analyst to select specific posts from the Twitter stream and show their geolocations along with statistics relevant to these geolocations.

7.2 Future Work

Our future works can be organized along 4 distinct axes: enhancing of the geolocalization techniques, extending the applicability of the proposed techniques, predicting mobility patterns based on social media posts, and predicting specific human activities based on social media and forum activity.

7.2.1 Enhancing the Geolocalization Techniques

The accurate identification of the actionable insights could allow us to identify undesirable situations in real time, allowing us to react fast. Having already developed a framework that increases the number of the geotagged posts, we would like to further improve the efficiency of our method by combining our method with other sources that can either verify the independence of the source, or to boost the accuracy.

Usage External Information Sources:
Initially, we would like to import third party information such as information from OpenStreet Maps, web information and articles that describe future events. Using these information, could allow us to get more representative keywords of a location, dramatically

boosting the accuracy of our methods, especially in the streaming case that tweets of the most recent timeslots can differ to those of the previous timeslots.

Usage of Sentiment Analysis:
Another potential parameter that could increase the accuracy of our method could be the usage of the sentiments expressed in the tweets of an area. Events such as concerts tend to be the reason for the generation of post with positive sentiments, while on the other hand events such as traffic jams will cause the generation of posts containing negative sentiments. The sentiments of the tweet we want to geotag could be used in order to match the tweet with areas containing similar sentiments.

Natural Language Processing Techniques:
Finally, it would be interesting to replace traditional Tf-Idf representations with Graph-based ones [42] that could capture the order of the text. In addition, we are planning to make use of word embeddings like “word2vec” [46] and “Glove” [58] that try to capture semantic information.

7.2.2 Extending the Applicability of the proposed Techniques

Hidden Insights:
Having increased the number of the geotagged posts deriving from a location, we would also like to use the hidden insights. This way, we can examine the volume of the hidden information we can get in cases of crisis or need. Such an example could be the geolocalization of posts describing a car accident. Information such as the number of the cars involved at the accident or the case that people are trapped in the car, could be identified in the post content. Furthermore, new information such as hotels and restaurants not registered on sites like Foursquare could be identified, enriching the knowledge of a locations.

Other Social Media and Forums:
Finally, we would like to apply our algorithms on other social media (i.e. Facebook, Instagram) and forums, verifying the efficiency of our methods and the independence from the source.

7.2.3 Predict Mobility Based on Social Media Posts

Although the mobility of the users can be tracked in real time with the usage of the CDRs, it is not possible to be predicted accurately. Our target is to predict the routes that the users are going to follow, combining the CDRs with the data shared on social media such as Twitter. The usage of the social media could allow us to predict upcoming events, while the usage of the CDRs could help us to identify the location these events are going to take place. Furthermore, the increase of the social and CDR activity could help us to identify locations that the users are going to move towards. The usage of the CDRs having as destination location the locations previously identified, could help us to assume the origin locations that the attendants are going to move from.

7.2.4 Predicting Human Activity Based on Social Media and Forum Posts

Finally, we would like to go one step further and predict important human behaviors. Based on our experience, people tend to share their problems and their opinions at forums or social media, seeking for opinions or solutions. Having proven by a variety of studies that social media can be used for reconstructing an accurate picture of the current events, we would like to investigate ways that social media and forum discussions could be used for the prediction of important human activities. As an important human activity, we could consider the prediction of the number of returns of devices such as gateways, or the number of the users that are going to attend an event such as concert.

Bibliography

[1] Facebook,https://www.facebook.com/.

[2] Google+,https://plus.google.com.

[3] Twitter,https://twitter.com.

[4] H. Abdelhaq, C. Sengstock, and M. Gertz. Eventweet: Online localized event detection from twitter. *Proceedings of the VLDB Endowment*, 6(12), 2013.

[5] O. Ajao, J. Hong, and W. Liu. A survey of location inference techniques on twitter. *Journal of Information Science*, 41(6):855–864, 2015.

[6] M. A. Azam, L. Tokarchuk, and M. Adeel. Human behaviour detection using gsm location patterns and bluetooth proximity data. In *The 4th international conference on mobile ubiquitous computing, systems, services and technologies-UBICOMM*, pages 428–433, 2010.

[7] J. P. Bagrow, D. Wang, and A.-L. Barabasi. Collective response of human populations to large-scale emergencies. *PloS one*, 6(3):e17680, 2011.

[8] M. Balduini, S. Bocconi, A. Bozzon, E. Della Valle, Y. Huang, J. Oosterman, T. Palpanas, and M. Tsytsarau. A case study of active, continuous and predictive social media analytics for smart city. In *ISWC Workshop on Semantics for Smarter Cities (S4SC).*

[9] M. Balduini, E. Della Valle, D. Dell'Aglio, M. Tsytsarau, T. Palpanas, and C. Confalonieri. Social listening of city scale events using the streaming linked data framework. In *ISWC*. 2013.

[10] M. Berlingerio, F. Calabrese, G. Di Lorenzo, X. Dong, Y. Gkoufas, and D. Mavroeidis. Safercity: a system for detecting and analyzing incidents from social media. In *Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on*, pages 1077–1080. IEEE, 2013.

[11] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 2003.

[12] S. Brdar, D. Culibrk, and V. Crnojevic. Demographic attributes prediction on the real-world mobile data. *MDC'12*, 2012.

[13] F. Calabrese, M. Colonna, P. Lovisolo, D. Parata, and C. Ratti. Real-time urban monitoring using cell phones: A case study in rome. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):141–151, 2011.

[14] F. Calabrese, F. C. Pereira, G. Di Lorenzo, L. Liu, and C. Ratti. The geography of taste: analyzing cell-phone mobility and social events. In *International Conference on Pervasive Computing*, pages 22–37. Springer, 2010.

[15] J. Candia, M. C. González, P. Wang, T. Schoenharl, G. Madey, and A.-L. Barabási. Uncovering individual and collective human dynamics from mobile phone records, 2008.

[16] H.-w. Chang, D. Lee, M. Eltaher, and J. Lee. @ phillies tweeting from philly? predicting twitter user locations with spatial word usage. In *ASONAM*, 2012.

[17] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *CIKM*, 2010.

[18] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090. ACM, 2011.

[19] D. Choujaa and N. Dulay. Tracme: Temporal activity recognition using mobile phone data. In *Embedded and Ubiquitous Computing, 2008. EUC'08. IEEE/IFIP International Conference on*, volume 1, pages 119–126. IEEE, 2008.

[20] A. Crooks, A. Croitoru, A. Stefanidis, and J. Radzikowski. # earthquake: Twitter as a distributed sensor system. *Transactions in GIS*, 17(1):124–147, 2013.

[21] E. M. Daly, F. Lecue, and V. Bicer. Westland row why so slow?: fusing social media and linked data sources for understanding real-time traffic conditions. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pages 203–212. ACM, 2013.

[22] N. Eagle and A. S. Pentland. Reality mining: sensing complex social systems. *Personal and ubiquitous computing*, 10(4):255–268, 2006.

[23] P. S. Earle, D. C. Bowden, and M. Guy. Twitter earthquake detection: earthquake monitoring in a social world. *Annals of Geophysics*, 54(6), 2012.

[24] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing. A latent variable model for geographic lexical variation. In *EMNLP*, 2010.

[25] V. Etter, M. Kafsi, and E. Kazemi. Been there, done that: What your mobility traces reveal about your behavior. In *Nokia Mobile Data Challenge 2012 Workshop. p. Dedicated task*, volume 2, 2012.

[26] D. Fox. Location-based activity recognition. In *the Proc. of the 30th Conf. on Advances in Artificial Intelligence*, pages 51–51, 2007.

[27] V. Frias-Martinez, V. Soto, H. Hohwald, and E. Frias-Martinez. Characterizing urban landscapes using geolocated tweets. In *SocialCom-PASSAT*, 2012.

[28] B. Furletti, L. Gabrielli, C. Renso, and S. Rinzivillo. Identifying users profiles from mobile calls habits. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, pages 17–24. ACM, 2012.

[29] H. Gao, J. Tang, and H. Liu. Mobile location prediction in spatio-temporal context. In *Nokia Mobile Data Challenge 2012 Workshop. p. Dedicated task*, volume 2, 2012.

[30] C. Giatsidis, F. D. Malliaros, D. M. Thilikos, and M. Vazirgiannis. Corecluster: A degeneracy based graph clustering framework. In *AAAI*, pages 44–50, 2014.

[31] B. Han, P. Cook, and T. Baldwin. Text-based twitter user geolocation prediction. *JAIR*, 2014.

[32] S. Hasan, X. Zhan, and S. V. Ukkusuri. Understanding urban human activity and mobility patterns using large-scale location-based data from online social media. In *Proceedings of the 2nd ACM SIGKDD international workshop on urban computing*, page 6. ACM, 2013.

[33] N. Hossain, T. Hu, R. Feizi, A. M. White, J. Luo, and H. A. Kautz. Precise localization of homes and activities: Detecting drinking-while-tweeting patterns in communities. In *ICWSM*, pages 587–590, 2016.

[34] C.-M. Huang, J. Jia-Chin Ying, and V. Tseng. Mining users behavior and environment for semantic place prediction. In *Nokia Mobile Data Challenge 2012 Workshop. p. Dedicated task*, volume 1, 2012.

[35] Y. Ikawa, M. Enoki, and M. Tatsubori. Location inference using microblog messages. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 687–690. ACM, 2012.

[36] S. Johnson. *The ghost map: The story of London's most terrifying epidemic–and how it changed science, cities, and the modern world*. Penguin, 2006.

[37] S. Kinsella, V. Murdock, and N. O'Hare. I'm eating a sandwich in glasgow: modeling locations with tweets. In *SMUC*, 2011.

[38] M. Kwan, C. Arrowsmith, and W. Cartwright. Visualizing population movements within a region, 2011.

[39] K. Leetaru, S. Wang, G. Cao, A. Padmanabhan, and E. Shook. Mapping the global twitter heartbeat: The geography of twitter. *First Monday*, 18(5), 2013.

[40] C. Li and A. Sun. Fine-grained location extraction from tweets with temporal awareness. In *SIGIR*, 2014.

[41] A. Lima, M. De Domenico, V. Pejovic, and M. Musolesi. Exploiting cellular data for disease containment and information campaigns strategies in country-wide epidemics. *arXiv preprint arXiv:1306.4534*, 2013.

[42] F. D. Malliaros and K. Skianis. Graph-based term weighting for text categorization. In *Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on*, pages 1473–1479. IEEE, 2015.

[43] F. D. Malliaros and M. Vazirgiannis. Clustering and community detection in directed networks: A survey. *Physics Reports*, 533(4):95 – 142, 2013.

[44] E. Malmi, T. M. T. Do, and D. Gatica-Perez. From foursquare to my square: Learning check-in behavior from multiple sources. In *The 7th International AAAI Conference on Weblogs and Social Media*, 2013.

[45] M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the twitter stream. In *SIGMOD*, 2010.

[46] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[47] K. Mo, B. Tan, E. Zhong, and Q. Yang. Report of task 3: Your phone understands you.

[48] R. Montoliu, A. Martínez-Uso, J. Martínez-Sotoca, and J. McInerney. Semantic place prediction by combining smart binary classifiers. In *Nokia Mobile Data Challenge 2012 Workshop. p. Dedicated task*, volume 1, 2012.

[49] V. Murdock. Your mileage may vary: on the limits of social media. *SIGSPATIAL Special*, 3(2):62–66, 2011.

[50] D. Naboulsi, R. Stanica, and M. Fiore. Classifying call profiles in large-scale mobile traffic datasets. In *INFOCOM, 2014 Proceedings IEEE*, pages 1806–1814. IEEE, 2014.

[51] S. M. S. J. T. Nadeem and M. C. Weigle. Demographic prediction of mobile user from phone usage. *Age*, 1:16–21.

[52] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A.-L. Barabási. Structure and tie strengths in mobile communication networks, 2007.

[53] B. Pan, Y. Zheng, D. Wilkie, and C. Shahabi. Crowd sensing of traffic anomalies based on human mobility and social media. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 344–353. ACM, 2013.

[54] S. M. Paradesi. Geotagging tweets using their content. In *FLAIRS Conference*, 2011.

[55] P. Paraskevopoulos, T.-C. Dinh, Z. Dashdorj, T. Palpanas, and L. Serafini. Identification and characterization of human behavior patterns from mobile phone data. *NetMob*, 2013.

[56] P. Paraskevopoulos and T. Palpanas. Fine-grained geolocalisation of non-geotagged tweets. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pages 105–112. ACM, 2015.

[57] P. Paraskevopoulos, G. Pellegrini, and T. Palpanas. When a tweet finds its place: Fine-grained tweet geolocalisation. In *International Workshop on Data Science for Social Good (SoGood), in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML PKDD)*, 2016.

[58] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[59] S. Phithakkitnukoon, T. Horanont, G. Di Lorenzo, R. Shibasaki, and C. Ratti. Activity-aware map: identifying human daily activity pattern using mobile phone data. In *the Proc. of the 1st Intl. Conf. Human Behavior Understanding*, pages 14–25, 2010.

[60] C. Ratti, A. Sevtsuk, S. Huang, and R. Pailer. Mobile landscapes: Graz in real time. In *Location based services and telecartography*, pages 433–444. Springer, 2007.

[61] C. Ratti, S. Sobolevsky, F. Calabrese, C. Andris, J. Reades, M. Martino, R. Claxton, and S. H. Strogatz. Redrawing the map of great britain from a network of human interactions. *PloS one*, 5(12):e14248, 2010.

[62] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

[63] A. Schulz, A. Hadjakos, H. Paulheim, J. Nachtwey, and M. Mühlhäuser. A multi-indicator approach for geolocalization of tweets. In *ICWSM*, 2013.

[64] P. Serdyukov, V. Murdock, and R. Van Zwol. Placing flickr photos on a map. In *SIGIR*, 2009.

[65] L. Sloan and J. Morgan. Who tweets with their location? understanding the relationship between demographic characteristics and the use of geoservices and geotagging on twitter. *PloS one*, 10(11):e0142209, 2015.

[66] T. Sohn, A. Varshavsky, A. LaMarca, M. Y. Chen, T. Choudhury, I. Smith, S. Consolvo, J. Hightower, W. G. Griswold, and E. De Lara. Mobility detection using everyday gsm traces. In *International Conference on Ubiquitous Computing*, pages 212–224. Springer, 2006.

[67] J. Steenbruggen, M. T. Borzacchiello, P. Nijkamp, and H. Scholten. Mobile phone data from gsm networks for traffic parameter and urban spatial pattern assessment: a review of applications and opportunities. *GeoJournal*, 78(2):223–243, 2013.

[68] Y. Tanahashi, J. R. Rowland, S. North, and K.-L. Ma. Inferring human mobility patterns from anonymized mobile communication usage. In *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*, pages 151–160. ACM, 2012.

[69] M. Tsytsarau, S. Amer-Yahia, and T. Palpanas. Efficient sentiment correlation for large-scale demographics. In *SIGMOD*, 2013.

[70] M. Tsytsarau and T. Palpanas. Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery*, 2012.

[71] M. Tsytsarau and T. Palpanas. Nia: System for news impact analytics. *KDD Workshop on Interactive Data Exploration and Analytics (IDEA)*, 2014.

[72] M. Tsytsarau, T. Palpanas, and M. Castellanos. Dynamics of news events and social media reaction. In *SIGKDD*, 2014.

[73] M. Tsytsarau, T. Palpanas, and K. Denecke. Scalable discovery of contradictions on the web. In *WWW*, 2010.

[74] M. Tsytsarau, T. Palpanas, and K. Denecke. Scalable detection of sentiment-based contradictions. *DiversiWeb, WWW*, 2011.

[75] S. Van Canneyt, O. Van Laere, S. Schockaert, and B. Dhoedt. Using social media to find places of interest: a case study. In *SIGSPATIAL (GEOCROWD)*, 2012.

[76] J. Wang and B. Prabhala. Periodicity based next place prediction. In *Nokia Mobile Data Challenge 2012 Workshop. p. Dedicated task*, volume 2, 2012.

[77] S. Wuchty. What is a social tie? *Proceedings of the National Academy of Sciences*, 106(36):15099–15100, 2009.

[78] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Who, where, when and what: discover spatio-temporal topics for twitter users. In *SIGKDD*, 2013.

[79] R. Zafarani and H. Liu. Evaluation without ground truth in social media research. *Communications of the ACM*, 58(6):54–60, 2015.

[80] Y. Zhu, E. Zhong, Z. Lu, and Q. Yang. Feature engineering for place category classification. In *Proceedings of Nokia Mobile Data Challenge Workshop, in Conjunction with Pervasive*, volume 12, 2012.