



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

DETECTING EMERGENCY SITUATIONS BASED ON LOCATION IN TWITTER

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS, MENCIÓN COMPUTACIÓN

HERNÁN ANDRÉS SARMIENTO ALBORNOZ

PROFESOR GUÍA:
BÁRBARA POBLETE LABRA
PROFESOR CO-GUÍA:
JAIME CAMPOS MUÑOZ

MIEMBROS DE LA COMISIÓN:
JUAN MANUEL BARRIOS NUÑEZ
JOSE MIGUEL PIQUER GARDNER
CARLA TARAMASCO TORO

SANTIAGO DE CHILE
2019

RESUMEN

Durante situaciones de emergencia tales como terremotos o atentados terroristas, los usuarios en redes sociales comparten masiva y colectivamente mensajes relacionados al evento. En muchas ocasiones, estos usuarios se transforman en testigos presenciales en el lugar de los hechos. En efecto, estos usuarios suelen compartir información relevante de manera mucho más rápida que los medios tradicionales de comunicación. Por consiguiente, así como los sensores físicos se activan cuando son estimulados, estos usuarios actúan como sensores ciudadanos localizados en lugares geográficos que reaccionan y detectan eventos de manera similar.

La mayoría de los métodos para detectar situaciones de emergencia usando redes sociales, se basan en identificar características en los mensajes que contienen palabras claves específicas para cierto dominio. Sin embargo, los métodos basados en palabras claves requieren modelos que son entrenados con datos históricos en dominios específicos, múltiples idiomas y para diferentes tipos de eventos (por ejemplo, terremotos, aluviones, incendios forestales, etc). Además de ser costoso, estos enfoques podrían fallar al detectar situaciones que ocurren de manera inesperada, tales como catástrofes no comunes o ataques terroristas.

Así mismo, las menciones colectivas de palabras clave no son el único tipo de fenómenos de auto-organización que pueden surgir cuando se produce una situación extrema en el mundo real. Para aprovechar esta información, utilizamos la actividad de geolocalización de manera auto-organizada para identificar situaciones de emergencia en Twitter. En nuestro trabajo proponemos detectar dichos eventos mediante el seguimiento de las frecuencias y las distribuciones de probabilidad del tiempo de llegada entre mensajes relacionados con ubicaciones específicas. Usando un clasificador estándar que es independiente de las características específicas del dominio, estudiamos y describimos situaciones de emergencia basadas únicamente en características correspondientes a la ubicación de los mensajes.

En nuestro trabajo introducimos el concepto de propagación geográfica para saber si un evento corresponde o no a una situación de emergencia. Este término define si un evento de emergencia es focalizado o difuso, lo cual permite reducir la cantidad de falsos positivos en eventos que no están relacionados a situaciones de emergencia.

Finalmente, nuestra contribución es proponer una prueba de concepto que permita detectar eventos de crisis mediante un modelo que no utiliza atributos del texto para clasificar, sino más bien la distribución y frecuencia que tienen los mensajes en ventanas de tiempo. Además nuestros hallazgos indican que las anomalías en la actividad de los usuarios en redes sociales, están relacionados con la ubicación los cuales proporcionan información para detectar automáticamente situaciones de emergencia independientes de su dominio. Los resultados indican que es posible detectar hasta un 80% de las situaciones de crisis sin utilizar características o atributos de texto.

ABSTRACT

During high-impact events such as earthquakes or terrorist attacks, social media users share massively and collectively messages related to the event. Usually, these users are eyewitness in the place where the event occurs. In fact, they often share relevant information faster than traditional news media. Consequently, as physical sensors become activated when stimulated, localized citizen sensors will also react in a similar manner.

Most methods for detecting emergency situations using Twitter rely on identifying features within messages that contain domain-specific keywords. However, keyword-based methods require models to be trained on historical data of specific domains, in multiple languages, and for different types of events (e.g., earthquakes, floods, wildfires, etc.). In addition to being costly, these approaches may fail to detect previously unexpected situations, such as uncommon catastrophes or terrorist attacks.

Nevertheless, collective mentions of keywords are not the only type of self-organizing phenomena that may arise when a real-world extreme situation occurs. To leverage this information, we propose to use self-organized geolocation related activity to identify emergency situations. In our work we propose to detect such events by tracking the frequencies, and probability distributions of the interarrival time of the messages related to specific locations. Using an off-the-shelf classifier that is independent of domain-specific features, we study and describe emergency situations based solely on location-based features in messages.

In our work, we introduce the concept of the geographic spread to know if an event is or not an emergency situation. This term defines if an emergency event is focalized or diffused, which allows us to reduce the number of false positives related to non-emergency situations.

Our contribution is to propose a proof of concept that allows detecting crisis events training a model that does not use textual features to classify. Instead, the model just uses the frequency and distribution of messages among time-windows. Furthermore, our findings indicate that anomalies in location-related social media user activity indeed provide information for automatically detecting emergency situations independent of their domain. The results show that we can detect around 80% of crisis situations without using textual features.

Contents

1	Introduction	1
1.1	Research Problem	2
1.2	Research Questions	3
1.3	Objectives	3
1.3.1	General Objective	4
1.3.2	Specific Objectives	4
1.4	Methodology	4
1.5	Thesis Outline	4
1.6	Contributions	6
2	Related Work	7
2.1	Social Media Messages in Mass Emergency	7
2.1.1	Classification, Extraction and Summarization during Emergency Situations	8
2.1.2	Crisis-Related Social Media Monitoring	12
2.2	Event Detection Based on Locations	14
2.3	Location Extraction in Social Media	14
2.4	Summary	16
3	Theoretical Framework	17
3.1	Classification	17
3.1.1	Support Vector Machine Classifier	18
3.1.2	Decision Tree	19
3.1.3	Random Forest	19
3.2	Clustering	20
3.2.1	Partitional Models	21
3.2.2	Density Models	22
3.3	Normal Distribution	23
3.4	Evaluation Methods and Metrics	23
3.4.1	Confusion Matrix	24
3.4.2	Clustering Evaluation	25
4	Methodology	27
4.1	Data Pre-Processing	27
4.2	Signal Creation	29
4.2.1	Geographical Hierarchy	29

4.2.2	Location Extraction	30
4.3	Time-Window	31
4.3.1	Features Extraction	31
4.3.2	Determining Optimal Window Size	32
4.3.3	Features Normalization	33
4.4	Geographic Spread	33
5	Experimental Setup	35
5.1	Dataset Description	35
5.2	Ground Truth	36
5.3	Feature Selection	37
5.3.1	Remove Redundant Features	37
5.3.2	System's Historical Data	38
5.4	Labeled Emergency Situation Events	39
5.4.1	Under-Sampling	39
6	Supervised Experimental Analysis	43
6.1	Choosing a Machine Learning Classifier	43
6.1.1	Support Vector Machine	44
6.1.2	Random Forest	45
6.2	Summary of the Supervised Results	46
7	Evaluation	49
7.1	Ground Truth	49
7.1.1	Independent Analysis of Hierarchies	49
7.1.2	Dependent Analysis of Hierarchies	50
7.1.3	Geographic Spread Analysis	51
7.2	On-line Evaluation	54
8	Discussion and Conclusion	59
8.1	Unsupervised Experimental Analysis	61
8.1.1	Clustering Centroid Models	61
8.1.2	Clustering Density Models	62
8.1.3	Summary of the Unsupervised Results	62
8.2	Final Comments and Future Work	63
	Bibliography	64
	Appendices	70
	A Tweet Object	71
	B Text pre-processing	73
	C Features Extraction	74
	D Printing Details to Remove Redundant Features	78
	E Results of the Supervised Approach	79

List of Tables

3.1	Classification Confusion Matrix.	24
5.1	List of earthquakes studied as ground truth, sorted by date.	36
5.2	Number of messages by signal related to location for specific countries.	37
5.3	An example of the dataset generated by the creation of the signals and removing attributes after features selection. The table shows the time-windows metadata and the attributes for classification. Class true identifies an emergency situation and class negative does not.	40
6.1	The best performance found for country and state hierarchy using the Support Vector Machine classifier. The Precision (P), Recall (R) and F1-score (F1) are computed in order to know the performance of the models. More details of the each k-fold evaluation can be seen in Table E.3 and E.4 in Appendix E.	44
6.2	The best performance found for country and state hierarchy using the Random Forest classifier. The Precision (P), Recall (R) and F1-score (F1) were computed in order to know the performance of the models.	45
7.1	On-line evaluation of events occurred in England by time-windows (T-W) using Country(2)-State + G.S. method. The table shows the total number of detected time-windows, the number of detected time-windows before the beginning and after to the end of the event. The last two columns show the detection delay time with respect to the beginning of the event and the top 3 bigrams when the detection occurs.	55
7.2	On-line evaluation of events occurred in England by time-windows (T-W) using Country(3)-State + G.S. method. The table shows the total number of detected time-windows, the number of detected time-windows before the beginning and after to the end of the event. The last two columns show the detection delay time with respect to the beginning of the event and the top 3 bigrams when the detection occurs.	56
8.1	Example of messages that include similar locations names.	60
B.1	Examples of text preprocessing.	73
C.1	Features extraction by time-windows for each type of signal.	74
C.2	Features extraction by time-windows for each type of signal considering normalized features.	75

E.1	List of the parameters using for searching the best performance in the country hierarchy for the SVM classifier.	79
E.2	List of the parameters using for searching the best performance in the state hierarchy for the SVM classifier.	80
E.3	Division of earthquakes used in 5-fold cross validation.	81
E.4	Results of the Support Vector Machine classifier using 5-fold cross validation.	82
F.1	Results of the external validation for k-means algorithm. The <i>iter_max</i> and <i>n_start</i> represent the number of iteration for finishing and the number of points for creating the cluster respectively.	86
F.2	Results of the external validation for k-medoid algorithm. We also used the dissimilarity matrix between elements.	87
F.3	Results of the external validation for DBScan algorithm.	88

List of Figures

1.1	Key components of the proposed approach.	5
2.1	Distribution of shortlisted articles by publication year based on the work of Akter and Fosso [5]. This plot only considers works published until the beginning of 2017.	8
2.2	Distribution of shortlisted articles by subject areas based on the work of Akter and Fosso [5].	9
2.3	Geo-location occurrences as a percentage of on-topic messages based on the work of [54].	10
2.4	Caution & advice task: inter-annotator agreement based on the work of Imram et al. [23].	11
2.5	A visual summary of the Twicalli website. The visual interface showing an earthquake occurred on December 25th of 2016. (a) Heat map of the complete country. (b) Signal formed by the number of published tweets every 60 seconds. (c) Marker of detected event, on click, information related with the event is displayed. (d) Last published tweets with buttons to reorder. (e) World map with clustered markers, user can see here when an event identified in the signal is occurring in other country. (f) Buttons that filter the markers considering the source of location information, so users can choose messages in which they trust more because some location sources are less trustworthy than others [38].	13
2.6	The scale of the geographic information entered by 3,149 users who indicated that they lived in the United States based on the work of Hecht et al. [19]. .	15
2.7	A subset of the gazetteer hierarchy based on the work of Yin et al. [59]. . . .	16
3.1	General approach for building a classification model based on the work of Tan et al. [52].	18
3.2	Three different ways of clustering the same set of points based on the work of Tan et al. [52].	20
3.3	Examples of different density distribution for Skewness and Kurtosis.	24
4.1	Key components of the data processing module.	28
4.2	Example of gazetteer tree for Chile.	29
4.3	Example of location mentioned on the body of message.	30
4.4	Example of location mentioned in the user profile.	30
4.5	Example of location mentioned on the body of message and the user profile.	31
4.6	Example of signal creation using the frequency of each location and metadata level.	32

4.7	Example of a geographic spread. Left image (a) represents the administrative division for seven states of Chile. Right image (b) represents the adjacency matrix created for these states.	34
5.1	Matrices of features correlation.	38
5.2	Average variation in emergency situations between time-windows. Positive and negatives values in x-axis represent the following and previous time-windows from the beginning of the event respectively.	41
5.3	Relationship between features in country and state hierarchy. Red circles represent positive class (<i>detection</i>) and blue circles represent negative class (<i>nothing</i>).	42
6.1	The ROC curves for each fold in the SVM classifier for country hierarchy. The x-axis label (1-Specificity) represents the False Positive Rate (FPR).	47
6.2	The ROC curves for each fold in the SVM classifier for state hierarchy. The x-axis label (1-Specificity) represents the False Positive Rate (FPR).	48
7.1	Average performance of 5-fold cross-validation by hierarchy independently just using labels.	50
7.2	An example of evaluation for independent analysis of hierarchies.	51
7.3	Average performance of 5-fold cross-validation by hierarchy independently just using time-windows.	52
7.4	An example of evaluation for dependent analysis of hierarchies.	53
7.5	Average performance of 5-fold cross-validation by hierarchy dependently just using time-windows.	54
7.6	An example of evaluation for dependent analysis of hierarchies with geographic spread.	57
7.7	Average performance of 5-fold cross-validation by hierarchy dependently with geographic spread just using time-windows.	58
8.1	A micro-blog message related to Manchester City Football Club.	60
8.2	Relationship between delay time and number of locations in the first detection for diffused and focalized emergency situations. Earthquakes are labeled as <i>EQ</i> and terrorist attacks as <i>TA</i>	61
A.1	Tweet metadata retrieved from the Twitter Public Streaming API, as of April 2010[30]. Colors represent the different types of information. Some fields can be deprecated and the current fields can be found on Twitter's developed page [26]	72
D.1	Output for <code>findCorrelation</code> over frequency features.	78
D.2	Printing details for <code>findCorrelation</code> over inter-arrival features.	78
F.1	The total within-cluster sum of squares and the average silhouette width for country and state hierarchy. The k number of clusters is tested between $2 \leq k \leq 15$	83
F.2	The average silhouette width for country and state hierarchy using the Euclidean distance. The k number of clusters was tested between $2 \leq k \leq 15$	84

- F.3 The average silhouette width for country and state hierarchy using the Manhattan distance. The k number of clusters was tested between $2 \leq k \leq 15$. . 84
- F.4 The k-nearest neighbor distances computed for country hierarchy. The plot can be used to help find a suitable value for the eps neighborhood for DBScan. 85

Chapter 1

Introduction

“The world is witnessing levels of human suffering unseen in generations. More than 120 million women, men and children worldwide are in need of humanitarian assistance in 2016. Human suffering from the impacts of armed conflicts and natural disasters has reached staggering levels. Nearly 60 million people, half of them children, have been forced from their homes due to conflict and violence. Between 2008 and 2014, a total of 184 million people were displaced by natural disasters, an average of 26.4 million each year” [3].

The above paragraph is a short note of the President of the General Assembly of the United Nations published on March, 2016. In the letter, he explains the effects not only of natural disasters worldwide, but also human provoked disasters in the last years. In this sense, there is no country, or community, nor person that is immune to disasters.

According to the World Disaster Report created in 2015 [1], in the past 10 years, about 631 disasters occurred on average per year with 89,934 people killed, 193,558 people affected and estimated damages of 162,203 million US dollars. Natural disasters killed 76,420 people and technological disasters¹ caused the loss of 7,513 lives per year. Furthermore, the economic impact of the Queensland floods² between 2010 and 2011 were estimated to about *A\$6.8bn* in direct losses. The 2016 Japan’s Kyushu Island earthquake³ had economic losses estimated between *US\$25 billion* and *US\$30 billion*.

However, this situation is not new in world history. According to the book by Haddow et al. [17], this is a phenomenon that has affected the civilizations since the cavemen era. Researchers have found that early hieroglyphics depict cavemen trying to deal with disasters. In support of this argument, we can find the first efforts to prepare, respond and recover during the era of the Cold War in the 1950s. Here, the concept of “Emergency Management”

¹A technological disaster is a catastrophic event that is caused by either human error in controlling technology or a malfunction of a technology system. For example, structural collapses, such as bridges, mines and buildings, but also industrial accidents, such as chemical or nuclear explosions

²A series of floods hit Queensland, Australia, beginning in November 2010. The floods forced the evacuation of thousands of people from towns and cities. At least 90 towns and over 200,000 people were affected.

³The 2016 Kumamoto earthquakes were a series of earthquakes, including a magnitude 7.0 mainshock which struck at 01 : 25 JST on April 16, 2016 (16 : 25 UTC on April 15) beneath Kumamoto City of Kumamoto Prefecture in Kyushu Region, Japan.

took force as a new initiative due to the imminent risk of nuclear war and nuclear fallout.

According to the United Nations Department of Humanitarian Affairs [2], a *disaster* is defined as a serious disruption of the functioning of society, causing widespread human, material, or environmental losses which exceed the ability of affected society to cope, using only its own resources. On the other hand, they define an *emergency* as a sudden and usually unforeseen event that calls for immediate measures to minimize its adverse consequences. Hence, the biggest difference between both, disaster and emergency, is the damage and the consequences on people.

In addition, *emergency management* is defined as: the discipline dealing with risk and risk avoidance. Risk represents a broad of range of issues and includes equally diverse set of players. That supports the premise that emergency management is key to the integrity of people’s day-to-day lives and should be incorporated into decisions and not just called upon during times of disaster [17].

1.1 Research Problem

During emergency situations, traditional media may suffer infrastructure issues and real-time communications could be disrupted. Instead, microblogging has played a critical role over the last fifteen years allowing users to share real-time information from people local to the incident, such as status updates, casualties, damages and alerts [32, 23, 51, 48]. For this reason, researchers have studied user behavior during these events to detect, summarize and classify messages with the goal of helping authorities, and the general public, with situational awareness to provide fast and conscientious responses during crisis situations.

Twitter is a microblogging platform that allows users to share short messages (called *tweets*) and is currently used worldwide by over 300 million people⁴. About 80% of Twitter users access from mobile devices, which contributes to the immediacy of diffusion of information, especially during crisis situations [12].

One of the most important tasks during these situations is to timely detect incoming real-world events. This is because, as with most social media conversations, messages are often overridden with irrelevant and redundant noise. In current works [32, 10, 22, 38], these tasks are solved with methods that rely on keyword based filters on the *Twitter Public Streaming API*⁵. The problem with keyword-based methods is the need to train in specific domains for different type of events. For example, to detect earthquakes and terrorist attacks, we need to train both domains separately because the characteristics of each event are different with respect to the number of the affected people or the magnitude of the event.

Olteanu et al.[44] generated a set of keywords based on different datasets. However, this is not sufficient for cases in which specific and previously unseen terms arise for a particular event (e.g., *#eqnz* for Earthquakes in New Zealand, or *#pabloph* for Typhoon Pablo in Philippines) [47, 8, 28]. Furthermore, these sets of keywords are commonly obtained for a particular language and will not work in others. For instance, to track and detect earthquakes

⁴<http://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/>

⁵<https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data.html>

we need a lot of keywords in many languages: terremoto (Spanish), earthquake (English), terremoti (Italian), tremblements de terre (French), etc.

Additionally, several works related to crisis situations show a strong relationship between a type of event (*what*) and the spatiotemporal dimensions (*when* and *where*). For example, researchers have observed that the top trends during earthquakes are related to location mentions [40], also, that there is a strong relationship between the proximity to a hurricane path and hurricane-related social media activity [31]. Other works have addressed the extraction of locations and points of interest during floods [35], as well as mixing geographic information system (GIS) information with geo-tagged messages to improve disaster mapping and real-time event tracking [20].

In summary, an important issue in current work is to detect and track emergency situations in a manner that is independent of the type of event, its domain and language. In fact, most work has focused on textual features because the specific domain being tracked is known. Furthermore, researchers have shown that pre-trained classifiers significantly drop their classification accuracy when used on different but similar disasters [24].

The main motivation of this work is to find evidence that allows us to detect emergency events by solely using locations for a specific country. Our purpose is to conduct an exploratory study to allow to evaluate if there is evidence to support our hypothesis, which claims there is a relationship between social media messages posted during crises and the locations where the events took place. For this reason, we performed a case study of earthquakes, which included several events that occurred in two countries with different native languages. In the first part of this work, we present a methodology for the creation of signals based on the administrative division of each country. In general terms, our goal is to detect anomalies in social media activity of locations. Later, we validate our methodology using different approaches and choose the model that has the lowest number of false positive detections.

1.2 Research Questions

Based on our literature review, in which we identified the problems for detecting emergency situations, we propose to address the following research questions:

- **RQ1:** Is there evidence that an emergency situation can be detected based on anomalies in the messages related to specific locations?
- **RQ3:** Can we detect an emergency situation independently of its domain and language?
- **RQ2:** Can we characterize an emergency situation independently of its domain and language?

1.3 Objectives

In the following section, we present both this work’s main objective and its specific objectives.

1.3.1 General Objective

The main objective of this thesis is to detect emergency situations based on bursts of activity in social media related to particular geographical locations.

1.3.2 Specific Objectives

1. To create a ground truth dataset, based on social media data that includes emergency events and non-emergency events.
2. To create and validate a model for detecting emergency situations across different domains and languages.
3. To characterize emergency situations independently of their domain and language.
4. To perform a case study with different types of emergency situations and non-emergency events.

1.4 Methodology

We propose a method based on recurring references to country level locations in message metadata. For this task, we create a gazetteer tree based on the hierarchy of the specific country, divide the messages into fixed time-windows and compute the frequency and the probability distributions of the interarrival time of the messages for each geographical hierarchy. To detect an emergency situation, we train a SVM classifier for each hierarchy and apply a geographic spread to filter false positives detection considering that an emergency situation can be *focalized* or *diffused*.

In order to provide a complete coverage of location-based detection of emergency situations, we propose an approach with two main modules (depicted in Figure 1.1).

The *data processing* module describes the data extraction process from Twitter and its corresponding signal creation based on location extraction. After that, we divide the signal into time-windows and compute the features for each. Finally, we define the geographic spread for decreasing the number of false positives detections.

The data classification module describes the step in which the emergency situations candidates are obtained. To find these candidates, we classify the time-windows and filter the false positive detections using the proximity between country states. Then, we include the geographic spread definition where we identify emergency events that affect small areas (focalized), and large areas (diffused).

1.5 Thesis Outline

This thesis is organized as follows. In Chapter 2, we introduce an overview of relevant literature related to this work. In this way, we explain the most relevant work of the crisis informatics, event detection in social media and location extraction.

In Chapter 3, we define the data analysis tools used in this work. We introduce some of

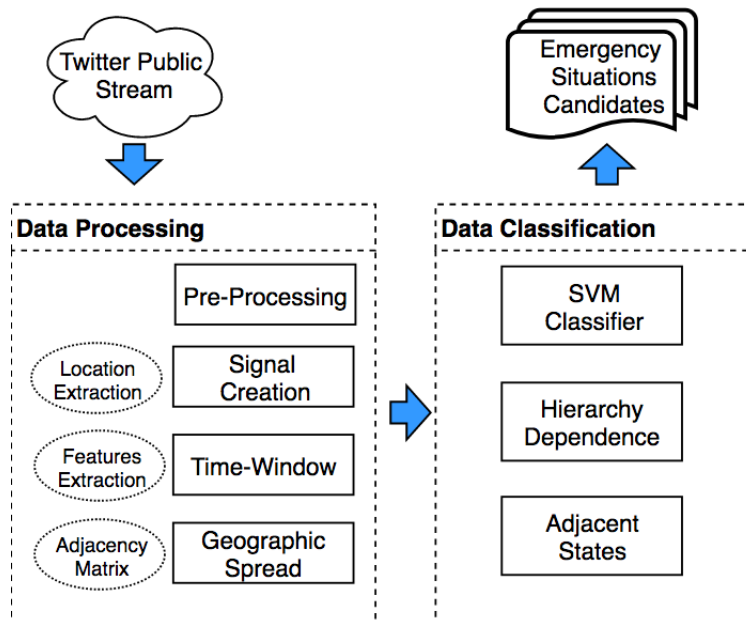


Figure 1.1: Key components of the proposed approach.

the most popular unsupervised and supervised machine learning approaches such as Support Vector Machine, Decision Tree, DBScan, among others. Furthermore, we explain some definitions about the probability distributions and some metrics to evaluate the effectiveness of the machine learning models.

Next, we present a complete description of our proposal in Chapter 4. We explain the process from the data extraction until the signal creation. Also, we introduce the concept of the geographic spread for decreasing the number of false positive detections.

In Chapter 5, we define our experimental setup where we present a full description of our dataset. Furthermore, we describe the process for obtaining the ground truth used for our models.

In Chapter 6, we use our ground truth and generate different solutions using unsupervised approaches. In this chapter, we demonstrate that is not possible generate groups of time-windows for dividing emergency situations and non-emergency events.

In Chapter 7, we change the approach and we use supervised learning. Here, we test with two different algorithms: Support Vector Machine and Random Forest. After several experiment with a lot of configurations, we find that the best model is Support Vector Machine.

In Chapter 8, we evaluate the best model found in the previous chapter. Here, we propose three evaluations, where we find out that the lowest number of false positive detections is when we use the concept of the geographic spread.

Finally, in Chapter 9 we deliver our discussion, conclusions and future work.

1.6 Contributions

Our main contribution is to create a methodology that detects instantaneous emergency situations just by using the location related information for a specific country. Furthermore, we train a classifier does not depend on language to detect a new event since they do not use textual features as input. Moreover, we characterize crisis situations that affect small or large geographical areas.

Additionally, our work has been presented in the following conferences (sorted by descending year):

- WebSci 2018, Amsterdam. Sarmiento, H., Poblete, B., & Campos, J. (2018, May). Domain-Independent Detection of Emergency Situations Based on Social Activity Related to Geolocations. In Proceedings of the 10th ACM Conference on Web Science (pp. 245-254). ACM.
- Alberto Mendelzon Workshop 2018, Calí. Sarmiento, H. & Poblete, B.(2018, May). Domain-Independent Detection of Emergency Situations Based on Social Activity Related to Geolocations. Not in Proceedings, just extended abstract.
- European Summer School in Information Retrieval 2017, Barcelona. Sarmiento, H. (2017, September). Detecting Emergency Situations by Inferring Locations in Twitter. In Proceeding of the Seventh BCS-IRSG Symposium on Future Directions in Information Access.

Chapter 2

Related Work

In this chapter, we give an introduction to the field of emergency situation management, event detection and location extraction. These topics are analyzed within the field of social media. First, we present prior work related to social media during mass emergencies, and we explain the most relevant works in social media monitoring and characterization of crisis situations. Then, we describe existing work related to event detection based on geolocations. The final section offers a short introduction of the area of location extraction from text by presenting the most relevant approaches for this thesis.

2.1 Social Media Messages in Mass Emergency

Twitter has been used extensively during emergency situations to extract and identify relevant information. However, social media exchanges during emergency situations have become so massive that it is necessary to sift through millions of data points to find useful information during an event [21].

In the past fifteen years, a large body of work related to the use of social media during emergency situations, also known as crisis informatics, has been published. In 2007, Palen and Liu [46] published one of the first papers on the subject explaining the relevance of collecting information in wikis about missing people after the attacks of 9/11 in 2001.

According to the systematic review published by Akter and Fosso in 2017 [5], the number of studies on the topic of crisis informatics the past seven years has increased. In their systematic review they considered results from SCOPUS databases, using the following search terms and their variants: (“disaster management” OR “emergency service” OR “disaster relief operations” OR “disaster resilience” OR “emergency management”) AND “big data”. Regarding the results, they included only journal articles retrieving 123 studies. Figure 2.1 shows the distribution of shortlisted articles by publication year, where the number of papers has increased since 2015.

Another interesting finding presented by Akter and Fosso [5] was the scattered spectrum of the subject areas with applications of Big Data in disaster contexts. These subject areas included Engineering, Computer Science, Social Science, Medicine, Environmental Science,

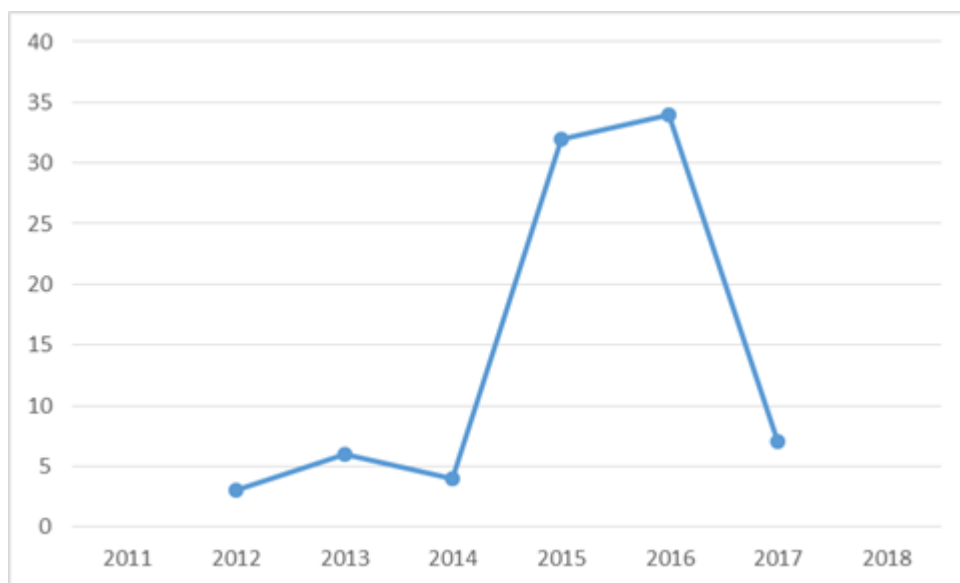


Figure 2.1: Distribution of shortlisted articles by publication year based on the work of Akter and Fosso [5]. This plot only considers works published until the beginning of 2017.

among others. Figure 2.2 displays results for 76 articles studied by the authors.

Imran et al. [21] also conducted a survey study in which they presented an extended summary of social media usage during emergency situations. This resulted in more than 150 papers related to these topics, where they included journals, full and short papers from different conferences and workshops. In fact, we can identify top conferences where the papers have been published, such as: The Web Conference (WWW), Special Interest Group on Information Retrieval (SIGIR), Conference on Information and Knowledge Management (CIKM) and Knowledge Discovery in Databases (KDD). Furthermore, other important ACM, AAAI and IEEE conferences have published several articles in this area, such as The International Conference on Weblogs and Social Media (ICWSM), Web Science Conference (WebSci), Intelligence and Security Informatics (ISI), European Conference on Information Retrieval (ECIR), Hypertext and Social Media, Visual Analytics Science and Technology (VAST), Conference on Human Information Interaction and Retrieval (CHIIR). In addition, we can identify a specific conference called Information Systems for Crisis Response and Management (ISCRAM).

As a consequence of the vast number of papers in the area, we focus our literature review on two main topics: *characterization of social messages during emergency situations* (classification, extraction and summarization) and *crisis-related social media monitoring*.

2.1.1 Classification, Extraction and Summarization during Emergency Situations

Most existing works classify social media messages during emergency situations into one or more categories such as *personal, informative, caution and advice, donations*, among others [24]. Extraction and summarization of the relevant/irrelevant information from the data is a difficult task. In fact, social media messages include irrelevant and redundant noise that

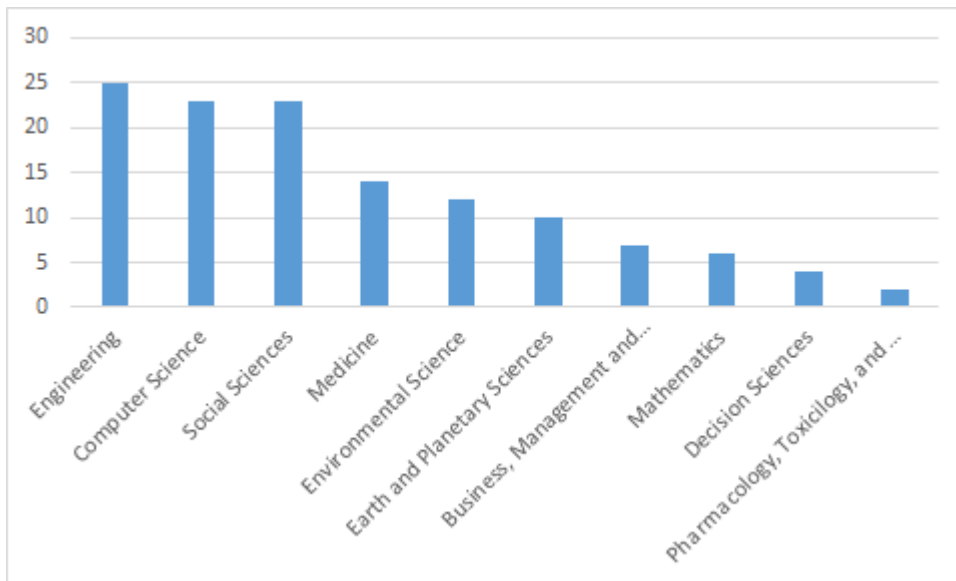


Figure 2.2: Distribution of shortlisted articles by subject areas based on the work of Akter and Fosso [5].

affects the effectiveness of useful information extraction using traditional techniques.

Next, we discuss work that explains the relationship among the emergency event, the type of event and the affected locations. We also include important qualitative findings that help understand the effect of a crisis situation in social media users.

CrisisLex [44] is a lexicon introduced to filter crisis-related messages from Twitter. To create this resource, the authors collected six disasters that affected several million people. They collected two data samples from Twitter: a keyword-based sample and location-based sample. For the keyword-based sample, the authors created two research teams who defined the terms that would be used to retrieve messages from Twitter. For the location-based sample, the data was partially collected using Topsy analytics. In addition, the authors collected data from 6 disasters that occurred in English-speaking countries (USA, Canada and Australia) which affected up to several million people. Based on these datasets, they used the most frequent terms in relevant messages to create their lexicon.

In relation to finding implicit relationships between the emergency situation and the affected locations, there are several works with interesting findings. Mendoza et al. [40] presented a study related to the 2010 Earthquake in Chile¹. The main objective of this work was to show how information propagated through the Twitter network, and to assess the reliability of Twitter as an information source under extreme circumstances. They studied the social phenomenon of the dissemination of the false rumors and confirmed news. The authors retrieved social media messages using the *Santiago* timezone, plus tweets which included a set of keywords. These keywords included hash-tags such as *#terremotochile* and the names of affected geographic locations.

¹The 2010 Chile earthquake occurred off the coast of central Chile on Saturday, 27 February at 03:34 local time (06:34 UTC), having a magnitude of 8.8 Mw.

As in the previously mentioned work, Vieweg et al. [54] studied two natural hazard events with the purpose of identifying information that may contribute to enhancing situational awareness. They covered the Red River Flood² (RR Flood) and the Oklahoma Fire³ (OK Fires) natural hazards, both occurring in 2009. Like the majority of prior work, this study also retrieved messages from Twitter using specific terms such as `red river` and `redriver`, for pulling Red River Flood tweets, and the terms `oklahoma`, `okfire`, `grass fire` and `grassfire`, for Oklahoma Grassfire tweets. One of the most relevant results for our thesis is the percentage of the on-topic (i.e., relevant to the emergency) messages with geolocation information. Figure 2.3 shows that the most named type of location in on-topic messages is the city hierarchy for both events (30% and 15% for “oklahoma fires” and the “red river flood” respectively). In contrast, location mentioning of country, place, and address have a lower frequency with less than 10% .

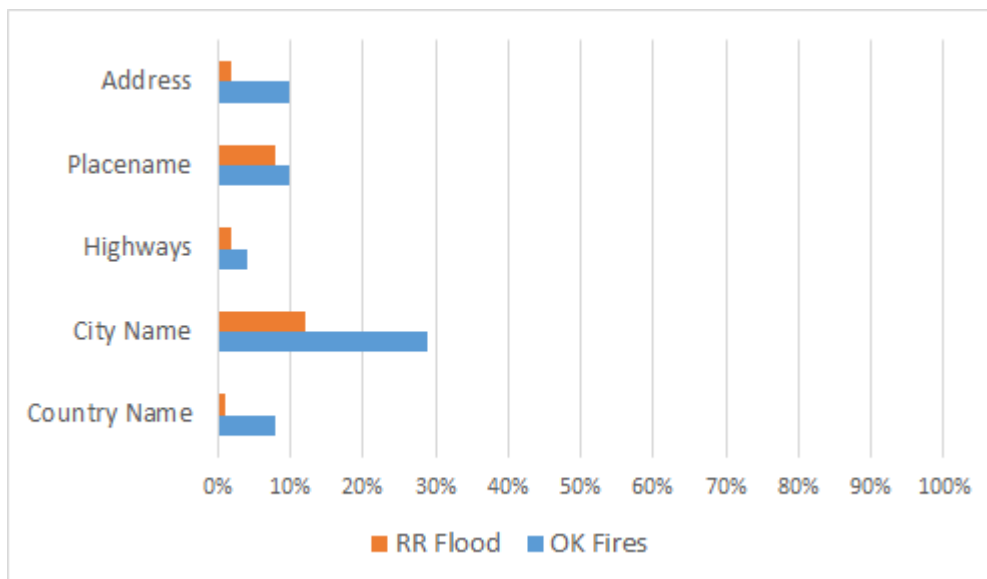


Figure 2.3: Geo-location occurrences as a percentage of on-topic messages based on the work of [54].

Another work related to emergency situations and location was presented by Kryvasheyev et al. [31]. In this paper, they found diverse relationships between the proximity of Hurricane Sandy⁴ and social media activity. For example, they found several phenomena between the pass of the hurricane along cities and its impact on social media activity. New York City, a city with severe damage during the event, had high social media activity, highly related to the proximity of the hurricane. In addition, they found an inverse relationship between the number of retweets and the level of activity, because affected locations produced more original content. Finally, they observed that the popularity of content was higher in directly affected areas than in others.

²The 2009 Red River flood along the Red River of the North in North Dakota and Minnesota in the United States and Manitoba in Canada brought record flood levels to the Fargo-Moorhead area. The flood was a result of saturated and frozen ground, Spring snowmelt exacerbated by additional rain and snow storms, and virtually flat terrain.

³The Oklahoma Fire occurred on April 9, 2009. High winds and dry conditions fueled numerous grassfires burning through central and southern Oklahoma and parts of northern Texas.

⁴Hurricane Sandy was the deadliest and most destructive hurricane of the 2012 Atlantic hurricane season.

Regarding the classification of messages during an emergency situation, Imran et al. [23] presented a study of the 2011 Joplin Tornado⁵, where they classified messages into six main categories. These divisions were based on the work of Vieweg et al. [55], which included an analysis of 32 specific types of information that contribute to situational awareness. In the work of Imran et al. [23], they considered the following topics: *caution and advice*, *information sources*, *donation*, *causalities and damage*, and *unknown*. For classifying messages, they used a Naïve Bayes classifiers with 10-fold cross validation. Several binary features were introduced such as, if the tweet contained an emoticon, a number, a hashtag, etc. The most important results in this work were related to the effectiveness of the model over the topics named above. In general terms, they found good precision, recall and f-measures values for the *caution* and *donation* but not for the topics *casualty* and *information source*. One important finding of this study related to the manual codification for the *caution and advice* task. As we can see in Figure 2.4, the annotators identified that a high percentage of topics related to the *what* and *where* dimensions (77.56 and 73.12 respectively).

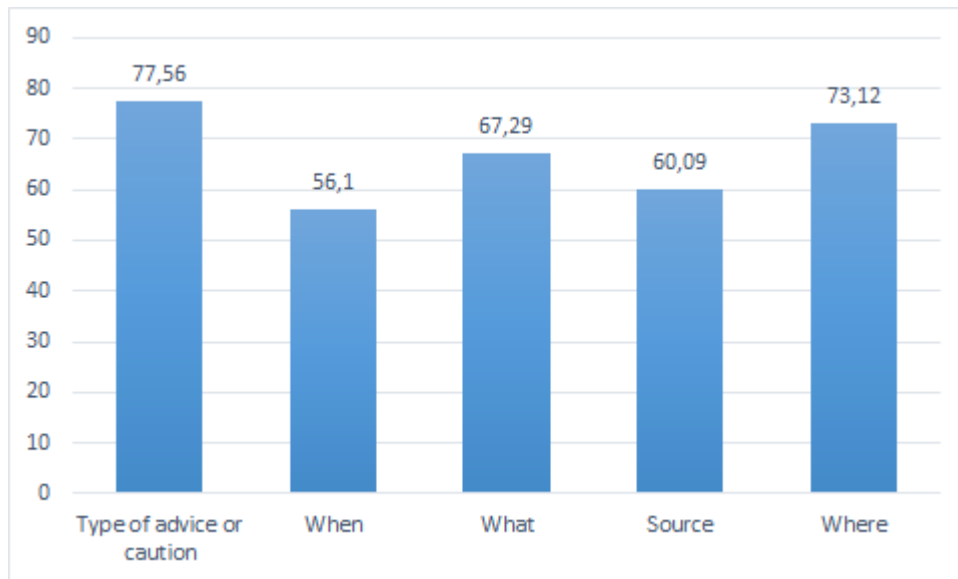


Figure 2.4: Caution & advice task: inter-annotator agreement based on the work of Imran et al. [23].

After the previous findings, Imran et al. [24] extended their work evaluating and classifying the same topics for two natural disasters: the 2011 Joplin Tornado and the 2012 Sandy Hurricane. They collected messages using the hashtags *#joplin*, and *#sandy* and *#nyc* for each event respectively. The main contributions in this paper was the methodology for evaluating both disasters. First, they trained and tested models for each disaster dependently, this means that they used, for example, a part of the Joplin dataset for training (66%) and the rest for testing (33%). In this experiment, recall and precision were good for most topics. However, when they evaluated across disasters, for example, training with the Joplin dataset and testing in the Sandy dataset, recall decreased considerable, but not precision. In conclusion, they demonstrated that generalizing between different emergency events is very difficult.

⁵The 2011 Joplin tornado was a catastrophic multiple-vortex tornado that struck Joplin, Missouri, late in the afternoon of Sunday, May 22, 2011.

Finally, Olteanu et al. [45] introduced an extensive systematic examination for 26 different crisis situations. In this paper they analyzed diverse types of events depending on the hazard category (natural or human-induced), development (instantaneous or progressive) and spread (diffused or focalized). The authors presented several findings, described next:

- With respect to the types and sources, messages from governments were often about caution and advice, such as tornado alerts. On the contrary, eyewitness tweets focused on affected individuals. Traditional news media and Internet media, on the other hand offered a variety of information including information about affected individuals, and messages of caution and advice.
- Regarding temporal aspects, they identified differences in the total volume of messages in each information type for instantaneous and progressive events. They additionally demonstrated that in instantaneous crises, outsiders, media and NGO (non-governmental organization) messages appeared early on, though, during progressive events, eyewitness and government messages appear early, mostly to warn and advise those in the affected areas, while NGO messages appear relatively late.

2.1.2 Crisis-Related Social Media Monitoring

One main task related to emergency situations is to detect a new real-crisis event in social media. Most existing event-detection methods described in the literature are based on keywords.

TweetTracker, presented by Kumar et al. [32], consists of a case study of tweets discussing the 2010s Haiti cholera outbreak⁶. The primary mechanism for monitoring tweets were through specific keywords and hashtag filters related to Haiti. To detect a new event, emerging trends was identified based on the analysis of older tweets. The system architecture of *TweetTracker* consists of four major components: the Twitter Stream Reader, where they retrieved messages based on user specified keywords, hashtags and geolocations. The *DataStore*, where data was constantly stored. And the *Visualization and Analysis Module*, where tweets was analyzed and filtered. Later, a map was included to focus on tweets of interest.

EMERSE, presented by Caragea et al. [10], used a set of keywords related to the Haiti earthquake and applied a SVM algorithm to classify messages. The main goal was to translate and classify messages for different languages. Furthermore, different sources were considered such as tweets and short message service (SMS) about Haiti disaster relief.

Like the *Twicalli* system [38], the authors introduced an unsupervised approach to detect earthquakes that only requires a general list of keywords. This work was based on the work of Guzman and Poblete [16], where the authors detected burst activity in social media using static time-windows for determining variation of the terms using the z-score value. In *Twicalli*, the main idea was to retrieve messages using specific keywords for earthquakes in different languages. Messages were later filtered by their geolocation and assigned by country. They next computed z-score variations between time-windows related to earthquake terms.

⁶The Haitian cholera outbreak was the first modern large scale outbreak of cholera, once considered a beaten back disease thanks to the invention of modern sanitation, yet now resurgent, having spread across Haiti from October 2010 to May 2017

Finally, they visualized earthquake detections and messages in a website⁷ as we can see in Figure 2.5.

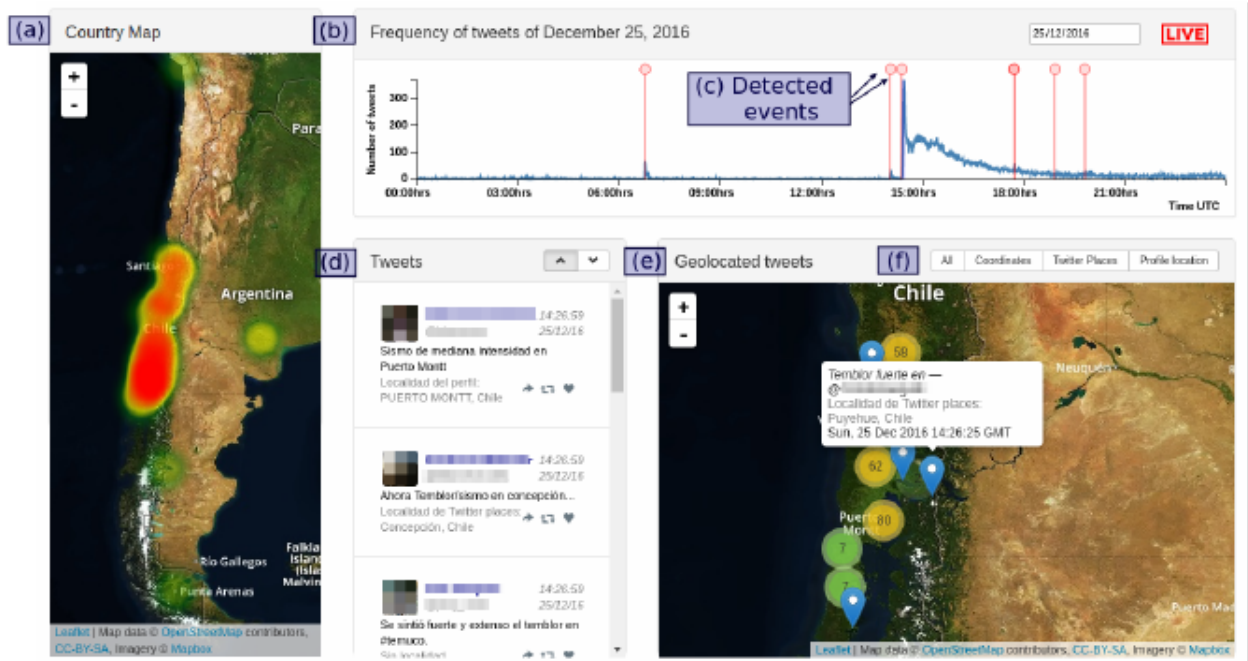


Figure 2.5: A visual summary of the Twicalli website. The visual interface showing an earthquake occurred on December 25th of 2016. (a) Heat map of the complete country. (b) Signal formed by the number of published tweets every 60 seconds. (c) Marker of detected event, on click, information related with the event is displayed. (d) Last published tweets with buttons to reorder. (e) World map with clustered markers, user can see here when an event identified in the signal is occurring in other country. (f) Buttons that filter the markers considering the source of location information, so users can choose messages in which they trust more because some location sources are less trustworthy than others [38].

Researchers at CSIRO Australia proposed *ESA* [9, 60], a system to detect disasters in Australia and New Zealand. This system was based on a probabilistic method that identifies bursty keywords, and historical data to build a language model of word occurrences. Alerts were identified if a term has a probability distribution that significantly deviates from the language model. After detecting an event, they applied clustering to get to the topics discussed for the targeted incident.

Similarly, the *Twitcident* [4] system detected incidents that rely on emergency broadcasting services, such as the police, the fire department and other public emergency services. The *Twitcident* framework translated the broadcasted message into an initial incident profile applied as a query to collect messages from Twitter, where an incident profile is a set of weighted attribute-value pairs that describe the characteristics related to the incident.

Finally, *AIDR* [22] is a platform that performs automatic classification of crisis-related microblog communications. The goal of the AIDR is to classify messages that people post

⁷Twicalli website <http://twicalli.cl/>

during disasters into a set of user-defined categories based on the works of Imran et al. 2013a [24] and Imran et al. 2013b [23]. The authors presented an architecture where they collected crisis-related messages from Twitter, asked a crowd to label a sub-set of those messages, and trained an automatic classifier based on the label. There are two important points in this work. First, they did not use pre-existing training data because it was not a satisfactory solution, given that crises had elements in common, and also had specific aspects which make domain adaptation difficult. The second point was that AIDR is not a system for continual event tracking. AIDR just tracks events when an instance is created.

2.2 Event Detection Based on Locations

In addition to crisis-related social media monitoring, there are also some unsupervised event detection approaches based on location information. The *Jasmine* system introduced by Watanabe et al. [57] detects local real-world events using geolocation information from microblog documents. To detect such events, they identified a group of messages that describe a particular theme, which were generated within a short time frame and a same geographic area.

Similarly, Unankard et al. [53] proposed an approach for early detection of emerging hotspot events in social networks with location sensitivity. In this study, the authors identified strong correlations between user locations and event locations when detecting emerging events using content similarity between clusters.

In the work introduced by Walther et al. [56], real-world events were detected on a small scale with messages from the New York metropolitan area. There, clusters were created for each candidate event and evaluated using cluster scores based on textual features (sentiment analysis, common themes, duplicates, etc) and other features (number of messages, unique coordinates, etc.).

In a supervised approach, the *TEDAS* [33] system detected, analyzed and identified events using refined rules (e.g., keywords, hashtags) and classified messages based on content and Twitter specific features such as URLs, hashtags and mentions. Location information was extracted using both explicit geographical coordinates and implicit geographical references in the content (i.e., locations in messages).

Conducted in a similar fashion, Becker et al. [7] proposed an on-line clustering technique, which continuously clusters similar tweets and then classifies the clusters using the Support Vector Machine algorithm. Events (clusters) were finally classified into real-world events or non-events.

2.3 Location Extraction in Social Media

Over recent years, researchers have focused on determining the exact geolocation of users in social media. In general, the goal is to know where the user is and to recommend specific topics as products, events or points of interest.

Bao et al. [6] presented an extensive systematic review of the location-based social net-

work (LBSN). They reviewed more than 50 papers published between 2003 and 2013. As for contributions, they identified a similar approach of recommendation systems for LBSN, considering the extraction and data sources of users information. In conclusion, they defined different methodologies based on the content, link analysis and collaborative filtering. And finally, they show the process of the recommendation for locations, users and activities.

Stock [50] presented a recently complete systematic review with 690 papers across 20 social media platforms. The results identified an extensive usage of Twitter (56%) as the main platform to extract information. Furthermore, the most used extraction methods were based on the message metadata and the user profile.

In order to explain the most relevant work used in this thesis, we offer a short explanation. In the work of Hecht et al. [19], authors found useful results about the location of the users in Twitter. They examined the location information provided in the users' profile. The findings suggested that 34% of users did not provide real location information and frequently incorporated fake locations or sarcastic comments that can fool traditional geographic tools (e.g., Google and Yahoo services). When users did input their location, they almost never specified it at a scale any more detailed than their city, as we see in Figure 2.6.

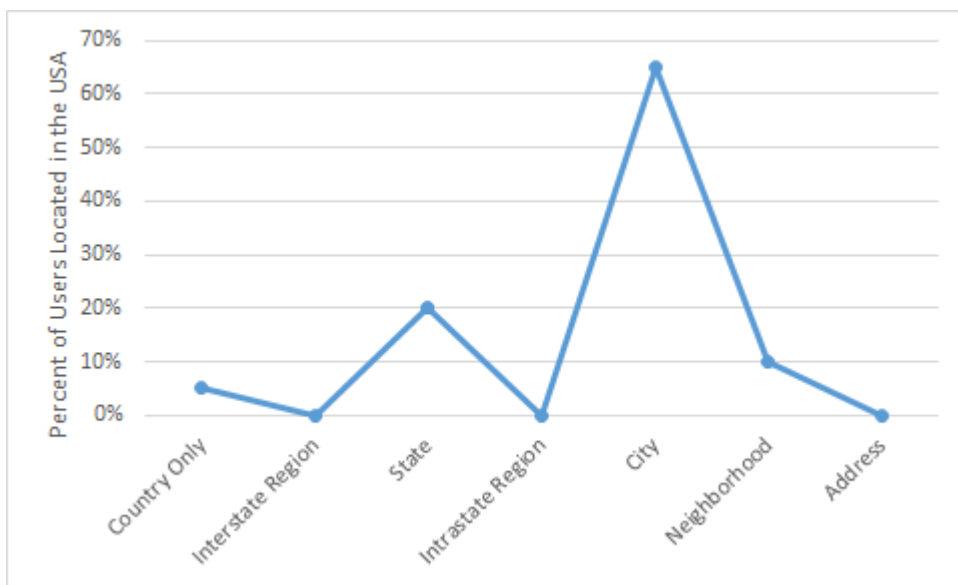


Figure 2.6: The scale of the geographic information entered by 3,149 users who indicated that they lived in the United States based on the work of Hecht et al. [19].

Graham et al. [15] collected a high number of tweets over a long period of time. They first set a bounding box for covering the whole planet. They retrieved 111 million tweets using Twitter's streaming API between November 10 and December 16, 2011. One goal in this paper was to study the reliability of key methods used to determine the language and location of content on Twitter. They compared three automated language identification packages to Twitter's user interface language setting and to a human coding of languages in order to identify common sources of disagreement. The paper also demonstrated that in many cases user-entered profile locations which differ from the physical locations from where users actually were sharing messages from.

Finally, Yin et al. [59] presented a novel algorithm that identifies all location mentions from three information sources: tweet texts, hashtags, and user profiles. They used a gazetteer database to infer the most probable locational focus of a tweet, which it can be represented in Figure 2.7, where each place was associated with a canonical taxonomy node. According to the authors, the algorithm had the ability to infer a locational focus that may not be explicitly mentioned in the message and determined its most appropriate granularity, e.g., city or country.

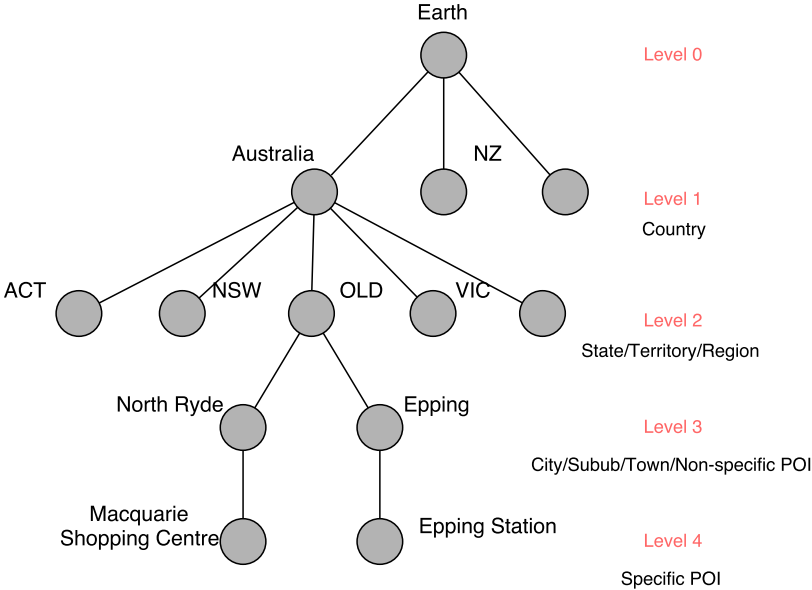


Figure 2.7: A subset of the gazetteer hierarchy based on the work of Yin et al. [59].

2.4 Summary

Most previous works on detecting an emergency situations rely on keywords using probabilistic temporal models for specific domains (e.g., keywords or location). In general terms, these approaches need knowledge of the event. Hence, events that are not like previously seen crisis situations cannot be identified using these methods. The works on event detection based on locations were designed to be applied to a small geographic area and based on historical data. Regardless of which approach was implemented (supervised or unsupervised), textual features were the most used attributes to characterize an event and on-line clustering was the most common technique to create candidate events.

Based on the works presented by Guzman et al. [16] and Maldonado et al. [38], we extend the ideas of bursty keywords and *z-score* variation between fixed time-windows and apply these proposals over locations to identify anomalies in social media activity. Furthermore, we do not use a set of keywords to detect specific types of events (e.g., earthquakes, floods or terrorist attacks). In contrast, our proposal focuses on detecting such events by tracking frequencies, and probability distributions of the interarrival time of the messages related to specific locations.

Chapter 3

Theoretical Framework

This thesis focuses on the fields Data Mining and Machine Learning. To analyze detections of emergency situations in social media, techniques from all these areas are required, as for instance, classification, clustering and hypothesis testing. Hence, the most important methods used in this thesis are described in this chapter.

The chapter is divided into three parts. First, we introduce the concept of *classification* along with the most common machine learning algorithms used for this task. Second, we explain the concept of *clustering* and common techniques for unsupervised classification. Finally, we introduce the evaluation measures for supervised and unsupervised learning.

3.1 Classification

Classification is the task of learning a target function f that maps each attribute set x to one of the predefined class labels y . The target function is also known as *classification model*. The input in this task is a collection of records, where each record also known as an instance or example. Additionally, one instance is characterized by a tuple (x, y) , where x is the attribute set and y is the class label (also known as category or target attribute)[52].

The general approach to solving a classification problem is to build classification models from an input dataset. In this way, building models with generalization capability is the main goal to predict the class labels of previously unknown records.

Figure 3.1 shows a general approach for solving classification problems. First, a training set is used to build a classification model. This set consists of records whose class labels are known. Hence, a testing set is created to evaluate classification model over unknown class labels.

The different classification learning algorithms to be considered in this thesis are explained in the following sections. For additional information about these algorithms please refer to [52, 41].

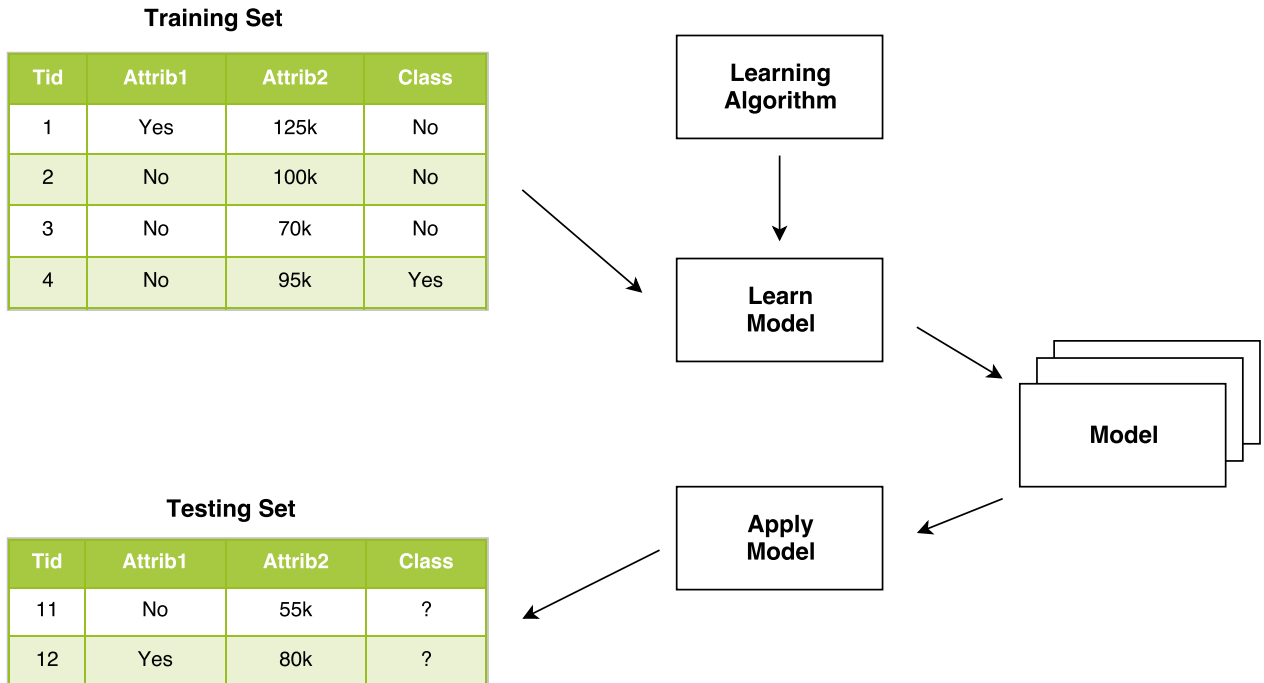


Figure 3.1: General approach for building a classification model based on the work of Tan et al. [52].

3.1.1 Support Vector Machine Classifier

The Support Vector Machine (SVM) is based on the concept of hyperplanes ($w^t \cdot x + b$) that define decision boundaries for binary classification problem $y_i \in -1, 1$ consisting of N training examples represented by x . The optimal hyperplane is the one that maximizes the margin between positives and negative observations in the training dataset [13]. The SVM algorithm can be formalized as the following optimization problem:

$$\min_{w,b} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi \quad (3.1)$$

$$\text{subject to } y_i(w^t x_i + b) \geq 1 - \xi_i \forall i \in \{1, \dots, N\}, \xi_i \geq 0 \forall i \in \{1, \dots, N\}$$

Where C is a user-specified parameter that represents the penalty of misclassifying the training instance. Hence, this parameter is referred to as the soft margin regularization and controls the sensitivity to possible outliers. The SVM classifier also has a user-specified parameter for controlling unbalanced data with respect to the number of instances for each class called class weights. There are other parameters for specific configurations of the kernels as the gamma, coefficient and degree.

The SVM formulations described above construct a linear decision boundary to separate the training examples into their respective classes. It is also possible to make SVMs find non-linear decision boundaries. A function $\phi(x)$ maps the feature space x into a high-dimensional

space is used. This high-dimensional space is called Hilbert space, where the dot product $\phi(x) \cdot \phi(x')$ is known as the kernel function $K(x, x')$. So the hyperplane is calculated in the high-dimensional space ($w^t \cdot \phi(x) + b$). Finally, we replace every dot product by a kernel function as is shown in the following expression:

$$\max_{\alpha} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \cdot K(x_i, x_j) \quad (3.2)$$

$$\text{subject to } \alpha_i \geq 0, \forall i \in \{1, \dots, N\}, \sum_{i=1}^N \alpha_i y_i = 0$$

Where the parameter $\alpha_i, i \in \{1, \dots, N\}$ corresponds to the *Lagrange multipliers* of the constrained optimization problem.

Many options for kernel function exist as following:

1. Linear function: $K(x_i, x_s) = x_i^T \cdot x_s$
2. Polynomial function: $K(x_i, x_s) = (x_i \cdot x_s + 1)^d$, where $d \in \mathbb{N}$ represents the polynomial degree.
3. Radial basis function (RBF): $K(x_i, x_s) = \exp(-\frac{\|x_i - x_s\|^2}{2\rho^2})$, where $\rho > 0$ represents the width of the kernel.

3.1.2 Decision Tree

A decision tree is simple yet widely used classification technique. The structure of the decision tree is like a flowchart in which each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label.

Greedy strategies are used to build a decision tree by making a series of locally optimum decisions on which attribute to use for partitioning the data. One such algorithm is *Hunt's algorithm*. In *Hunt's algorithm*, "a decision tree is grown in a recursive fashion by partitioning the training records D_t that are associated with node t and the class labels $y = \{y_1, y_2, \dots, y_c\}$ " [52]. The recursive definition of *Hunt's algorithm* is to select a partition of the records using an attribute test condition into smaller subsets when D_t contains records that belong to more than one class. A child node is created for each outcome of the test condition and the records in D_t are distributed to the children based on the outcomes. The algorithm is then recursively applied to each child node. The recursion termination is applied when all the records in D_t belong to the same class y_t . Then t is a leaf node labeled as y_t .

3.1.3 Random Forest

Random Forest is a class of ensemble methods that combines the predictions made by multiple decision trees. Each tree is created based on the values of an independent set of selected random vectors [52]. Bootstrap aggregation (also known as bagging) is used into the model-building process to choose N samples, with replacement, from the original training set. About one-third of the cases are left out of the bootstrap sample and not used in the construction of the tree. In this step, the Out-of-Bag (OOB) data is used to get a running unbiased estimate

of the classification error as trees are added to the forest. It is also used to get estimates of variable importance.

To perform prediction using the trained random forest, the algorithm uses the following steps:

1. Take the test features and use the rules of each randomly created decision tree to predict the outcome and store the predicted target.
2. Calculate the votes for each predicted target.
3. Consider the highest voted predicted target as the final prediction. This concept of voting is known as majority voting.

3.2 Clustering

A cluster is a set of similar objects based only on information found in the data that describe the objects and their relationships. The main goal of cluster analysis is for the object within a group to be similar (or related) to one another and different from (or unrelated to) the objects in other groups (Figure 3.2).

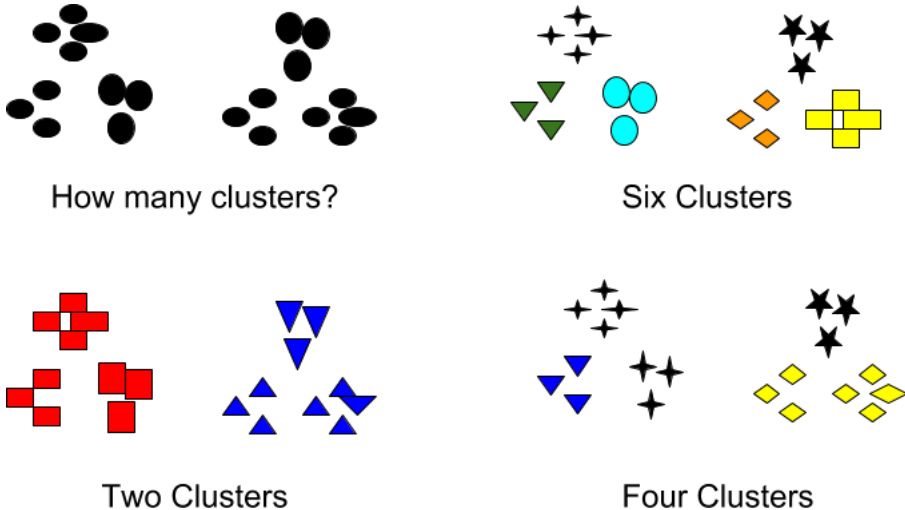


Figure 3.2: Three different ways of clustering the same set of points based on the work of Tan et al. [52].

One cluster can be differentiated from another using distance measure between their attributes. Some distance measures are explained as following:

1. Manhattan distance:

$$d_{man}(x, y) = \sum_{i=1}^n |x_i - y_i| \tag{3.3}$$

2. Euclidean distance:

$$d_{euc}(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \tag{3.4}$$

3. Minkowski distance:

$$d_{\text{mink}}(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p} \quad (3.5)$$

4. Pearson correlation distance:

$$d_{\text{cor}}(x, y) = 1 - \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (3.6)$$

The different unsupervised algorithms to be considered in this thesis are explained in the following sections. For additional information about these algorithms please refer to [52, 41].

3.2.1 Partitional Models

The main idea in this class of clustering algorithm is to create K clusters of the data, where the number K is a user-specified parameter. Each object in the data is assigned to the nearest cluster center, such that the squared distances from the clusters are minimized. Furthermore, the algorithms prefer clusters of approximately similar size, as they will always assign an object to the nearest centroid.

K-means

K-means is a *hill-climbing* algorithm, which guarantees convergence to a local optimum, but not necessarily a global optimum [34]. The main idea in this algorithm is to use the means to represent the clusters and use them as a guide to assign object to clusters.

Given an initial set of K means m_1, \dots, m_k , the algorithm proceeds by alternating between two steps [37]:

1. Assignment step: assign each object to the cluster whose mean has the least squared Euclidean distance such that:

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2 \forall j, 1 \leq j \leq k\} \quad (3.7)$$

where each x_p is assigned to exactly one $S^{(t)}$.

2. Update step: compute the new means to be the centroids of the observations in the new cluster. These centroids are calculated as following:

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \quad (3.8)$$

The algorithm converges when the assignments no longer change, so the convergence is satisfied. The most used criterion is the minimization of the squared error E of all the objects in the data:

$$E = \sum_{i=1}^K \sum_{o \in C_i} |o - \mu_i|^2 \quad (3.9)$$

where o is an object in the data that belongs to cluster C_i , μ_i is the mean of the cluster C_i and K is the number of clusters.

K-medoids

In contrast to K-means algorithm, K-medoids chooses data points as centers known as medoids. A medoid can be defined as the object of a cluster whose average dissimilarity to all of the objects in the cluster is minimal [29]. The general procedure for the algorithm is as follows:

1. Randomly choose k objects into data as the initial medoids.
2. Each one of the remaining objects is assigned to the cluster that has the closest medoid.
3. Randomly select a nonmedoid object in the current cluster, which will be referred to as $O_{nonmedoid}$.
4. Calculate the cost of replacing the medoid with $O_{nonmedoid}$. The cost is the difference in the square error if the current medoid is replaced by $O_{nonmedoid}$.

$$E = \sum_{i=1}^K \sum_{o \in C_i} |o - O_{medoid(i)}|^2 \quad (3.10)$$

If E is negative, then make $O_{nonmedoid}$ the medoid of the cluster.

Each step of the algorithm is repeated until there is no change.

3.2.2 Density Models

In this class of clustering algorithm, the main idea is to keep growing clusters as long as their density is above a certain threshold. Clusters are defined as areas and the objects in these sparse areas are usually considered to be noise and border points. In contrast to partitional clustering where algorithms detect only a cluster of a convex shape, density models detect clusters of arbitrary shape.

DBSCAN

In the DBSCAN algorithm, given a set of points in some space, it groups together points that are closely packed together. The main idea is to create clusters that have a high enough density, high enough being specified by the user. Unlike the K-means and K-medoids, the number of clusters are not specified in DBSCAN. In this algorithm, specified-user parameter are: (1) ε (known as *eps*) represents the maximum radius of the neighborhood from the core point p ; (2) *minPts*, the number of points reached by the core point p ; (3) distance function can be chosen by the user, and has a major impact on the results [52, 41].

The DBSCAN algorithm can be abstracted into the following steps:

1. Find the ε neighbors of every point, and identify the core points with more than *minPts* neighbors.
2. Find the connected components of core points on the neighbor graph, ignoring all non-core points.

3. Assign each non-core point to a nearby cluster if the cluster is an ε neighbor, otherwise assign it to noise.

3.3 Normal Distribution

The probability distribution is a description of a random phenomenon in terms of the probabilities of events. One of the most common distribution is the Normal Distribution (known as Gaussian or Gauss Distribution). It is a distribution where the values are plots in a symmetrical fashion, and most of the results are situated around the probability mean. Normal distribution can be formalized as the following expression:

$$f(x | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.11)$$

where μ is the mean of the distribution, σ is the standard deviation and σ^2 is the variance.

To evaluate some behavior and metrics over the Normal Distribution, we describe the following measures:

- Standard Score: more commonly referred to as *z-score*, is the number of standard deviations from the mean a data point is.

$$z\text{-score} = \frac{x_i - \mu}{\sigma} \quad (3.12)$$

- Skewness: is a measure of the lack of symmetry. A distribution is symmetric if it looks the same on the left and right of the center point. For univariate data x_1, x_2, \dots, x_N the formula for Skewness is:

$$Skewness = \frac{\sum_{i=1}^N (x_i - \mu)^3 / N}{\sigma^3} \quad (3.13)$$

- Kurtosis: is a measure of the “tailedness” of the probability distribution. Likewise to the concept of Skewness, Kurtosis is a description of the shape of the data distribution. For univariate data, the formula is:

$$Kurtosis = \frac{\sum_{i=1}^N (x_i - \mu)^4 / N}{\sigma^4} \quad (3.14)$$

Examples of different shapes according to the values of Skewness and Kurtosis are shown in Figure 3.3.

3.4 Evaluation Methods and Metrics

The following sections present some concepts, tools and techniques used in our supervised and unsupervised experiments.

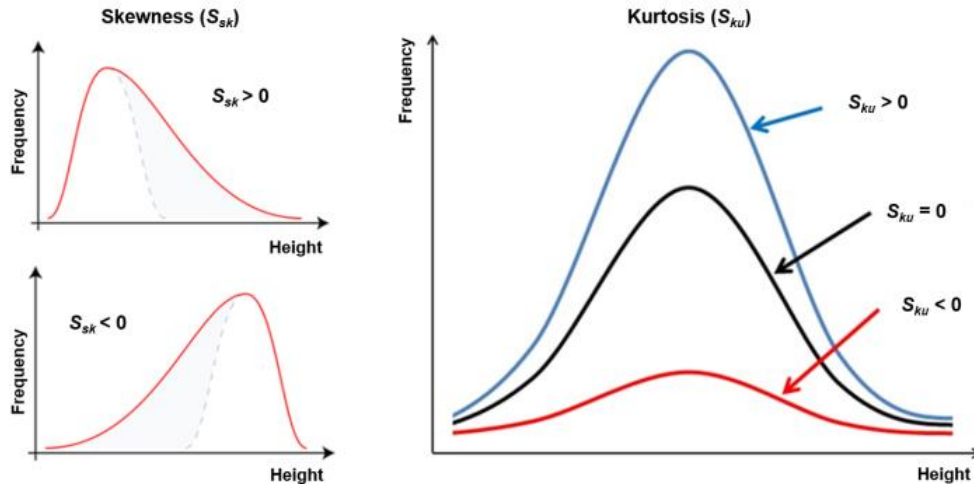


Figure 3.3: Examples of different density distribution for Skewness and Kurtosis.

3.4.1 Confusion Matrix

In classification tasks, the evaluation of the performance is based on the counts of test record correctly and incorrectly predicted by the model. The predicted outputs are compared with their corresponding real values from the testing dataset. Using this approach for a binary classification problem, four possible outputs can be obtained as is shown in Table 3.1, known also as Confusion Matrix.

The first outcome, *True Positive (TP)*, represents the object O , belongs to class C and is classified as such. Secondly, *True Negative (TN)* represents the object O , does not belong to class C and it is not classified as a member of class C . Unlike the TP and TN , which describe correct classification of the object O for class C , *False Positive (FP)* and *False Negative (FN)* represent objects misclassified. On the one hand, FP describes that although object O does not belong to class C , it is classified as member of class C . On the other hand, FN represents that object O belongs to class C , and is not classified as a member of class C .

Table 3.1: Classification Confusion Matrix.

	Actual Value: Positive	Actual Value: Negative
Prediction Outcome: Positive	True Positive (TP)	False Negative (FN)
Prediction Outcome: Negative	False Positive (FP)	True Negative (TN)

According the different outputs explained above, the following measures can be computed:

- Precision (P): the proportion of correctly classified positive observations over all the observations classified as positive.

$$Precision = \frac{TP}{TP + FP} \quad (3.15)$$

- Recall (R): the proportion of positive classified observations over all the actual positive

observations.

$$Recall = \frac{TP}{TP + FN} \quad (3.16)$$

- F_1 -score: the harmonic mean of precision and recall.

$$F_1\text{-score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (3.17)$$

- False Positive Rate (FPR): the proportion of the false positives over all the negative observations.

$$\text{False Positive Rate} = \frac{FP}{FP + TN} \quad (3.18)$$

3.4.2 Clustering Evaluation

Internal Criteria

Also known as *unsupervised evaluation* or *internal indexes*, the *internal criteria* is the clustering evaluation, where the evaluation of the clustering is compared with result itself and only using information present in the data set. Unsupervised measures are often divided into two classes: measures of cluster cohesion (compactness, tightness) and measures of cluster separation (isolation). For additional information about the internal criteria, please refer to [52, 18].

External Criteria

The External Criteria (also known as Supervised Evaluation or External Indexes) is the clustering evaluation that uses information not present in the dataset (e.g., class labels). Here, the clustering result is compared with the ground truth, and if the result is somehow similar to the reference, this final output is considered as a "good" clustering. Some measures used in this thesis are as follows:

- Purity: the purity measure focuses on the representative class, i.e., the class with majority object, within each cluster. Formally, purity can be computed with the following expression:

$$Purity(\Omega, C) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j| \quad (3.19)$$

where $\Omega = \{\omega_1, \omega_2, \dots, \omega_k\}$ is the set of clusters and $C = \{c_1, c_2, \dots, c_j\}$ is the set of classes.

- Entropy: the entropy measure is the expected amount of uncertainty in a cluster. It can also be represented as a measure of disorder in the cluster. The entropy measure can be computed as follows:

$$Entropy(\Omega) = - \sum_k \frac{P(\omega_k)}{N} \log \frac{P(\omega_k)}{N} \quad (3.20)$$

where $P(\omega_k)$ is the probability of an element being in cluster ω_k and N is the number of points in the dataset.

- Normalized Mutual Information (NMI): is a measure that allows for trading off quality of the clustering against the number of clusters. The NMI measure can be computed as follows:

$$NMI(\Omega, C) = \frac{I(\Omega; C)}{[H(\Omega) + H(C)]/2} \quad (3.21)$$

and I is mutual information computed as follows:

$$I(\Omega; C) = \sum_k \sum_j P(\omega_k \cap c_j) \log \frac{P(\omega_k \cap c_j)}{P(\omega_k) \cdot P(c_j)} \quad (3.22)$$

where $H(\Omega)$ and $H(C)$ are the entropies of the Ω and C respectively, and $P(\omega_k \cap c_j)$ is the probability of a element being in the intersection of ω_k and c_j .

- Variation of Information (VI): it is highly related to the mutual information, which measures the amount of information that is lost or gained in changing from the class set to the cluster set. By a random variable view similar to the previous case, we can compute the variation of information as follows:

$$VI(C, C') = 2H(C, C') - H(C) - H(C') \quad (3.23)$$

where $H(C)$ is the entropy associated with clustering C and $H(C')$ is the entropy associated with clustering C' .

Chapter 4

Methodology

Our focus in this thesis is to detect an emergency situation based on identifying anomalies in social media activity related to locations. In this way, the main task is to extract locations from messages by reducing the noise and irrelevant information.

The *data processing* module describes the data extraction process from Twitter (depicted in Figure 4.1). Key components are divided in four tasks: (1) We pre-processed data to allow a better analysis because social media messages are often noisy and redundant. (2) We created discrete signals based on location extraction using a geographical dictionary (also known as *gazetteer* [14]) to create a geographic hierarchy for a specific country. In addition, we created signals in different levels of the geographic hierarchy and the tweet metadata. (3) We divided signals into fixed time-windows and computed non textual features for each. (4) In order to discard those detections that occur in isolated and non connected locations, we created a geographic spread based on the proximity between locations by using an adjacency matrix M , which an element M_{ij} represents whether a location i is directly connected with a location j .

4.1 Data Pre-Processing

Our data consist of messages published by users in the Twitter microblogging platform. Twitter is an on-line social media platform oriented sharing short messages. We explain in more detail this social network by defining terminologies used in this thesis [25].

A *tweet* is a message published by a user in Twitter. The tweet is usually a text that must not exceed 140 (or 280 depending of the user) characters in length. Furthermore, the tweet can contain images, videos, geolocation (also known as geotagging) and links to internal or external contents. The action of publishing a message also is called *tweet*. In the Twitter platform, similar messages can be organized by specific topic or keyword. A *hashtag* is any word or phrase immediately preceded by the $\#$ symbol that allow this action.

Users in the platform can be identified by a username, which is always preceded by the $@$ symbol. Users can be mentioned in the tweet's text, called a mention. In addition a user can *follow* another user, which means that the user is subscribed to another Twitter

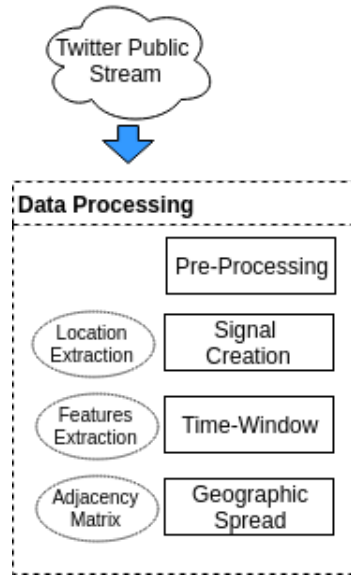


Figure 4.1: Key components of the data processing module.

account to see its tweets. In this action, the user who follows another user is called a *follower*. Furthermore, users can share a tweet published by another user: this message (and the action as well) is called *retweet*.

We use Twitter as social media source for several reasons. First, Twitter is currently used worldwide by over 300 million people¹ and 80% of their users access from mobile devices. Hence this contributes to the immediacy of diffusion information in crisis situations. Secondly, there are many types of sources in this social network as people (e.g., common or politicians), news media (e.g., newspapers and radio stations), ONG's and public services. Thirdly, Twitter provides a *Public Streaming API*² to retrieve the most recent tweets filtering by specific keywords or "drawing" a geographic real-place to get messages (also known as bounding box). Forth, Twitter provides a semi-structured data as we see in Appendix A.1.

Since we consider users as citizen sensors, we filter messages depending of native language of each country using the attribute *lang* retrieved from the tweet metadata. For example, if we analyze Italy, we just consider messages in Italian. Furthermore, we remove user mentions, URLs, special characters and apply text tokenization. Also we do not remove hashtags or stopwords³ (we only remove the # symbol). An example of this pre-processing is shown in Table B.1 in Appendix B.

Our text processing method was developed in Python using some libraries as *NLTK*⁴ and a tweet preprocessor tool called *Preprocessor*⁵. In Table B.1, we perform examples in different languages (e.g., Spanish, English and Italian) and metadata level (e.g., tweet text and user location). In the case of language as Spanish and Italian, we remove accent marks from the

¹<http://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/>

²Application programming interface

³Stopwords usually refers to the most common words in a language, there is no single universal list of stop words used by all natural language processing tools

⁴<http://www.nltk.org/>

⁵<http://github.com/s/preprocessor>

raw text.

4.2 Signal Creation

We create a set of discrete-time signals for each location, which indicates each time that a message related to a specific location was posted. In order to explain the effect of an emergency situation within its local and national scope, we use the lowest possible geographical hierarchy level available with the aim of comparing the impact in the highest level. Furthermore, we study the anomalies in different metadata levels to understand how locations are shared in Twitter, for instance, either based in the locations set by users in their profile or sharing location in their messages.

4.2.1 Geographical Hierarchy

Using the idea of “*gazetteer as a tree*” presented by Yin et al. [59] (Figure 2.7), we construct our gazetteer tree based on Geonames⁶ and Wikipedia⁷ for each country to analyze. However, in the study of Yin et al. [59], the gazetteer hierarchy presents four levels where the lowest level represents a specific point of interest.

In our approach, we use a subset of the gazetteer hierarchy with three levels, in which the lowest level is represented by a city since a large amount of users specify their location down to this level [19]. For example, if we have the *city:Manchester*, we associate this location with *region-state:North West* and also with *country:England*. Furthermore, the name of locations are considered just in the native language of the country. For instance, in the case of Italy, we consider *Roma* and not *Rome*. These modifications the original method can be seen in Figure 4.2. Here, we consider the country Chile, and for this reason we just retrieve messages in Spanish. We also extract the geographic hierarchy from the resources mentioned above to construct the gazetteer tree.

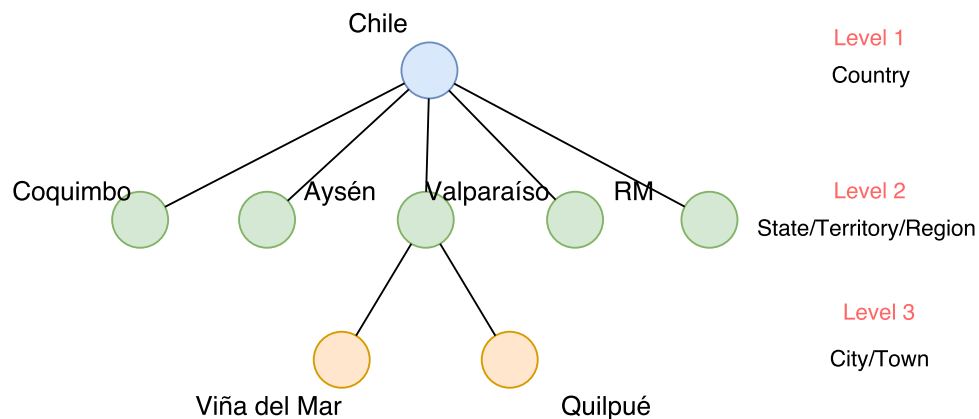


Figure 4.2: Example of gazetteer tree for Chile.

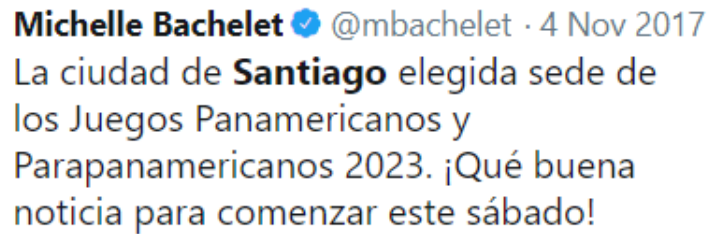
⁶<http://download.geonames.org/export/dump/>

⁷<http://www.wikipedia.org/>

4.2.2 Location Extraction

The structure of the tweet metadata contains information about the message and the user. Considering the aforementioned geographical hierarchy, we extract locations from different parts of the metadata, creating 3 signals for each location:

- Tweet Text: location is mentioned in the attribute `text` of tweet object, that is, on the body of the message. Figure 4.3 has an example of where the location *Santiago* is mentioned in the message.



Michelle Bachelet ✓ @mbachelet · 4 Nov 2017
La ciudad de **Santiago** elegida sede de los Juegos Panamericanos y Parapanamericanos 2023. ¡Qué buena noticia para comenzar este sábado!

Figure 4.3: Example of location mentioned on the body of message.

- User Location: location is mentioned in the attribute `location` inside the `user` object, that is, the location set by the user in the profile. An example is in Figure 4.4 where the location *Santiago* is mentioned in the user profile.



Figure 4.4: Example of location mentioned in the user profile.

- Tweet Text - User Location: location is mentioned in the attribute `text` of tweet object and also location is mentioned in the attribute `location` inside the `user` object. This means that the location is mentioned in the body of the message and the user who shares the message has the same location in the profile. In this case, tweet text and user location can be different in the smallest hierarchy, but in the highest level can be the same location. An example can be shown in Figure 4.5 where the location *Santiago* is mentioned in the message and the user who shares a message has settled in her profile the user location in *Santiago*.



Michelle Bachelet ✓ @mbachelet · 4 Nov 2017
La ciudad de **Santiago** elegida sede de los Juegos Panamericanos y Parapanamericanos 2023. ¡Qué buena noticia para comenzar este sábado!

Figure 4.5: Example of location mentioned on the body of message and the user profile.

Given a small portion of users sharing their current location using GPS coordinates [15], we do not consider this level of the tweet metadata in this work.

Mixing geographical hierarchy and locations in microblog metadata, we create $N \times M$ signals where N is the number of locations obtained by gazetteer tree and M is the number of metadata-levels extracted from the tweet object. For instance, we create a signal for *city:Valparaíso* and we find this hierarchy in *metadata:Tweet Text* and also in *metadata:User Location*. This means that we track the mention of *city:Valparaíso* at the level of the body of message and at the level of the user profile location individually. Furthermore, we create signals in the highest levels of the tree. Here, we create signals for *state:Valparaíso* and *country:Chile* at the level of the body of the message and the user profile respectively. An example of these signals is shown in Figure 4.6.

4.3 Time-Window

A time-window (or a window) is a certain time interval which is split in order to compute aggregated characteristics in data streams. Furthermore, these time-windows allow us to analyze discrete signals instead of continuous signals. In addition, we can compare adjacent (or a set of adjacent) time-windows and find variations among them related to features in each. Hence, in this stage we address the problems of how to divide and determine the time-window size to detect a new emergency situation and what features by the time-window allow it.

4.3.1 Features Extraction

Our main focus in this work is to detect an emergency situation without computing text features over the messages. In this way, our features are based in the number of messages by time-windows (*the frequency*), the time difference between the messages in a time-window (*the interarrival time*) and their probabilistic distribution (*the skewness and kurtosis*). Specifi-

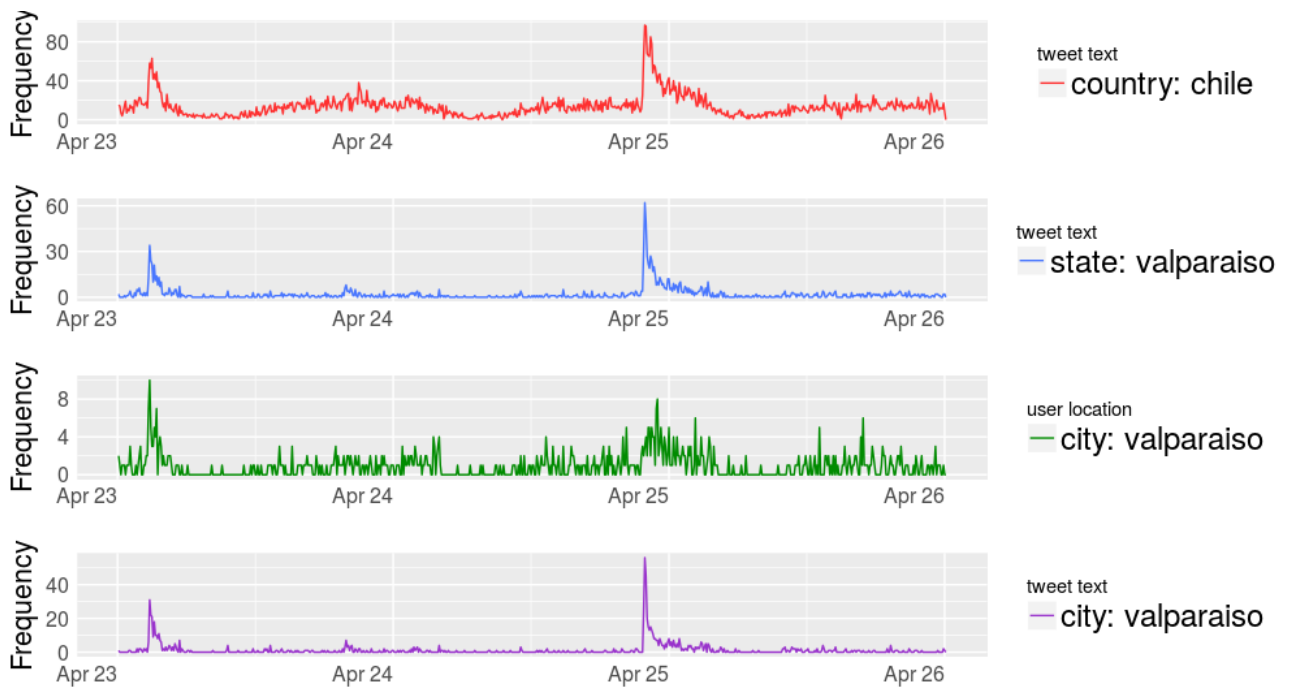


Figure 4.6: Example of signal creation using the frequency of each location and metadata level.

cally, we will explain some definitions about the interarrival time and the probabilistic distribution. After that we will summarize briefly each kind of features.

Interarrival Time

To characterize the urgency of the messages within a certain time-window, we compute the *interarrival time* which is defined as $d_i = t_{i+1} - t_i$, where d_i denotes the time difference between two consecutive social media messages i and $i+1$ that arrived in moments t_i and t_{i+1} respectively. Using this definition and according the work of [27], high-activity events have a high-frequency in the first bins represented by values $d_i \approx 0$ seconds. We used our previous results which explain different patterns in social media activity related to the interarrival time when an emergency happens [49]. In fact, our previous work revealed that a crisis event had a lesser difference between messages than non-emergency situations.

To quantify a high-frequency in very small values of d_i , we compute *skewness* (equation 3.13) and *kurtosis* (equation 3.14) values, which represent the asymmetry and the tailedness of the shape of probability distribution respectively [39]. Using the above explanation about the interarrival time between messages and their probabilistic distribution, we summarize our features as show Table C.1 in Appendix C.

4.3.2 Determining Optimal Window Size

The window size is defined as the time length (in seconds) which our method splits signals and compute every feature explained above. According to Guzman and Poblete [16]: “If the window size is too small, the occurrence of empty windows for a term increases, making the

noise rate increase and frequency rate tend towards zero. On the other hand, if the window size is too large, the stability of the signal becomes constant and bursty keyword detection is delayed". Using this definition and according to the empirical results presented by Maldonado et al. [38], we divide our signals into windows of six minutes because it divides a 24-hour day exactly, making the analysis easier to understand and to compare from different days.

4.3.3 Features Normalization

A normalization method according to Tan et al. [52], "is to make an entire set of values have a particular property. If different variables are to be combined in some way, then such a transformation is often necessary to avoid having a variable with large values dominate the results of the calculation". In order to allow the comparison of the computed features explained above, we use the *z-score* normalization presented in the equation 3.12. The main reasons for using this type of normalization are: we do not know every value of the population in our dataset (as in the case of min-max normalization); and we need to compute a score based on the past values to know variations between time-windows. Moreover, using the expression presented in the equation 3.12, we compute the *z-score* as follows:

$$zscore = \frac{x_i - \mu_k}{\sigma_k} \quad (4.1)$$

where x_i is the value i of the analyzed metric in the current time-window, μ_k and σ_k are mean and standard deviation of the previous k time-windows respectively.

Using the above explanation about the data normalization, the interarrival time between messages and the probability distribution, we finally compute sixteen features divided in two main groups: the frequency of the messages by time-windows and the inter-arrival time between social media posts. Table C.2 in Appendix C shows a description of each.

4.4 Geographic Spread

An emergency situation that affects and mobilizes response in a small area is defined as *focalized*, while a disaster with a large geographic impact is defined as *diffused* [45]. Using this definition, we extend this concept to represent neighborhoods between locations obtained from section 4.2.1. For that purpose, we create an *adjacency matrix* M , where $M_{i,j} = 1$, represents if two locations are geographically connected and $M_{i,j} = 0$ if they are not connected. For instance, if an event is diffused (e.g., an earthquake), the detection should be in adjacent-locations independently of metadata-level. On the other hand, if an event is focalized (e.g., a terrorist attack), just one location should be detected but in different metadata-levels simultaneously (e.g., at the level of user location and tweet text for the same location).

For example, using the administrative division (of a part) of Chile, we can construct the adjacency matrix based on the direct proximity between two locations in the country. The values of the main diagonal in the adjacency matrix are equal to 1 because the same location is connected to itself. As we explain above, $M_{i,j} = 1$ represents if two locations are geographi-

cally connected. If we look at the *Valparaíso* state on the map (Figure 4.7a), this location has three neighbors: *Coquimbo*, *RM* and *O'Higgins* state. Then, in our adjacency matrix (Figure 4.7b), we set the values $M_{valparaiso,coquimbo} = 1$, $M_{valparaiso,rm} = 1$ and $M_{valparaiso,ohiggins} = 1$ in those connected states. Otherwise, we set the value $M_{i,j} = 0$ in those non-connected states such as $M_{valparaiso,maule} = 0$, $M_{valparaiso,biobio} = 0$ and $M_{valparaiso,araucania} = 0$.

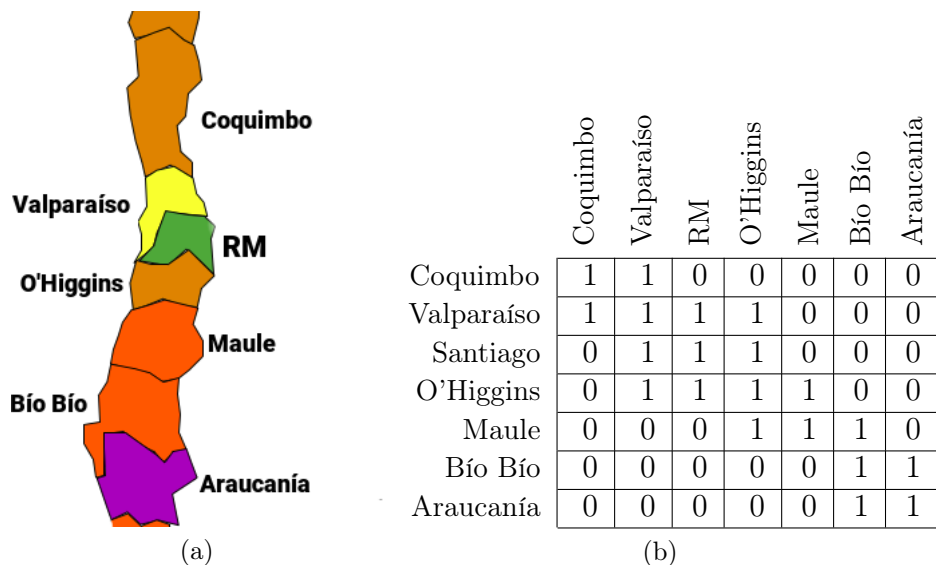


Figure 4.7: Example of a geographic spread. Left image (a) represents the administrative division for seven states of Chile. Right image (b) represents the adjacency matrix created for these states.

In our proposal, the geographic spread is very relevant for removing false positive detections since an emergency situation does not have isolated locations when an event occurs. For example, using the Figure 4.7, we can see that the Coquimbo and Valparaíso states are directly connected but Araucania is not with them. If we find detections just in these states, we then discard it an emergency situation because generally, natural or human-induced disaster reach a common area between neighboring locations. For instance, earthquakes or nuclear blasts impact several neighboring locations in short or medium time and in this case we do not find isolated locations. Unlike the disasters, other types of events such as soccer matches, music festivals o political elections, can generate detections in several states but the scope can generate isolated or non-connected locations.

Chapter 5

Experimental Setup

Our experiments aim to find empirical evidence that supports the effectiveness of our method for detecting emergency situations. In this way, we prepare our experimental dataset in accordance with the methodology presented in Section 4.

First, we construct our ground truth based on two publicly available earthquake catalogs where these natural disasters occur in two countries. Second, we extract the administrative division to create gazetteer trees. For each analyzed country, we compute signals in different levels of the geographic hierarchy and the tweet metadata. Third and using the same administrative division, we create an adjacency matrix to apply the geographic spread. Fourth, we apply features selection over our sixteen features divided into two groups: frequency and interarrival time.

Finally, we divide and label our dataset as: training and testing. We also apply under-sampling because our data present a high imbalance between positive and negative labels.

5.1 Dataset Description

Our hypothesis is to find empirical evidence that we can identify an emergency situation without specific domain keywords over the twitter stream. Hence, we needed to retrieve random messages about any topic and any place in the world. Additionally and strictly, a portion of the messages must contain information about an emergency situation.

Generally, several works use public datasets to improve and compare techniques. For example, the most common available catalog is *TREC (Text REtrieval Conference)*¹. In this resource, we find topics as confusion track, query track, question answering track, microblog track and others. In the last mentioned track, the goal is to explore technologies for monitoring a stream of social media posts with respect to a user's interest profile. However, the identified interest profiles do not represent an evaluation that allow to evaluate emergency events. Then, we could not use TREC for our evaluation methodology.

¹<https://trec.nist.gov/>

In contrast, we generated our own dataset based on the messages retrieved from Twitter. For this work we collected data from Twitter Public Streaming API², which allows access to subsets equal to 1% of public status descriptions in real-time. With this tool we could retrieve either messages using a set of keywords or messages from specific locations setting a “bounding box”. In our approach, we got entire subsets of messages without using keywords or specific locations. In addition, this subset of public status descriptions represent a good sample of the full status published in Twitter for high-impact real-world events [42]. Hence, we retrieved messages related to any topic, written in any language and posted anywhere in the world in this micro-blog service.

5.2 Ground Truth

To construct our ground truth, we first identified *instantaneous-diffused* crises according to the definition presented by Carr [11]. These events are characterized by the fact that no one could do anything to prevent them and their effects were felt by an entire community. This definition is relevant because an unexpected (or instantaneous) event generates an anomaly in the frequency of the social media activity since it disrupts users’ normal life. Furthermore, diffused events affect a large portion of users simultaneously, generating a collective reaction in neighboring affected locations. Based on the aforementioned definitions, we chose earthquakes as crises to study unexpected events that affect thousand and million people at the same time.

We analyzed five earthquakes (considered as instantaneous-diffused crises) with magnitudes between $5.5Mw$ and $7.6Mw$ ³, occurred in Italy and a Spanish-speaking country (such as Chile) between October 2016 and April 2017 (Table 5.1). For that purpose, we collected 20 million messages 12-hours before and after the emergency situation events related to any topic posted in Twitter.

Table 5.1: List of earthquakes studied as ground truth, sorted by date.

Country	Datetime (UTC)	Magnitude (Mw)	Language
Italy	2016-10-26 17:10:36	5.5	Italian
Italy	2016-10-30 06:40:17	6.6	Italian
Chile	2016-12-25 14:22:26	7.6	Spanish
Chile	2017-04-23 02:36:06	5.9	Spanish
Chile	2017-04-24 21:38:28	6.9	Spanish

Following our proposed methodology, we created both the gazetteer hierarchies⁴ for each country and constructed the signals based on each hierarchy and metadata-level. In this step, we filtered messages based on the locations of each country and discarded the others messages. As a result of the number of messages of each signal (Table 5.2), we did not consider all signals related to city hierarchy since a great amount of small cities have zero

²<https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data.html>

³Mw: the moment magnitude scale

⁴<http://users.dcc.uchile.cl/~hsarmien/gazetteer.html>

frequency in a normal situation unlike to capital or metropolitan cities. However, we kept these frequencies for the aggregate signals at the level of the state and country.

Table 5.2: Number of messages by signal related to location for specific countries.

Hierarchy	Metadata-level	Messages
All	All	87,291
Country	Tweet Text	11,584
	User Location	25,313
	Tweet Text - User Location	1,417
State	Tweet Text	4,110
	User Location	13,352
	Tweet Text - User Location	86
City	Tweet Text	1,415
	User Location	8,971
	Tweet Text - User Location	20

5.3 Feature Selection

After the features extraction where we computed sixteen features divided into two main groups, we selected the most relevant features based on the following criteria.

5.3.1 Remove Redundant Features

Data can contain attributes that are highly correlated to each other, which can affect the performance of learning and mining algorithms. In Figure 5.1, we present two correlation matrices for frequency and interarrival time features. For our analysis, we consider a high-correlation between attributes when the value is about 0.75. Furthermore, we needed to remove redundant features not just using pair-to-pair correlation but by computing the correlation with other features. In this sense, we estimated the absolute values of pair-wise correlations. If two variables have a high correlation, the method looks at the mean absolute correlation of each variable and removes the variable with the largest mean absolute correlation.

Based on the above explanation with a cutoff equal to 0.75, the results show that the attributes *freq_rollz10*, *freq_rollz15* and *freq_rollz20* have a high correlation with the other attributes with values 0.79, 0.98 and 0.77 respectively (see Appendix D.1). Hence, we removed these attributes from our features. Similarly, we applied the same method to the interarrival time features. First, we computed the correlation matrix and later used the method with a cutoff equal to 0.75. In this group of features, the results show the attributes *Q1*, *Q2* and *Q3* have a high correlation with the other attributes. (see Appendix D.2). In the same way as the above analysis, we also removed these variables from our attributes.

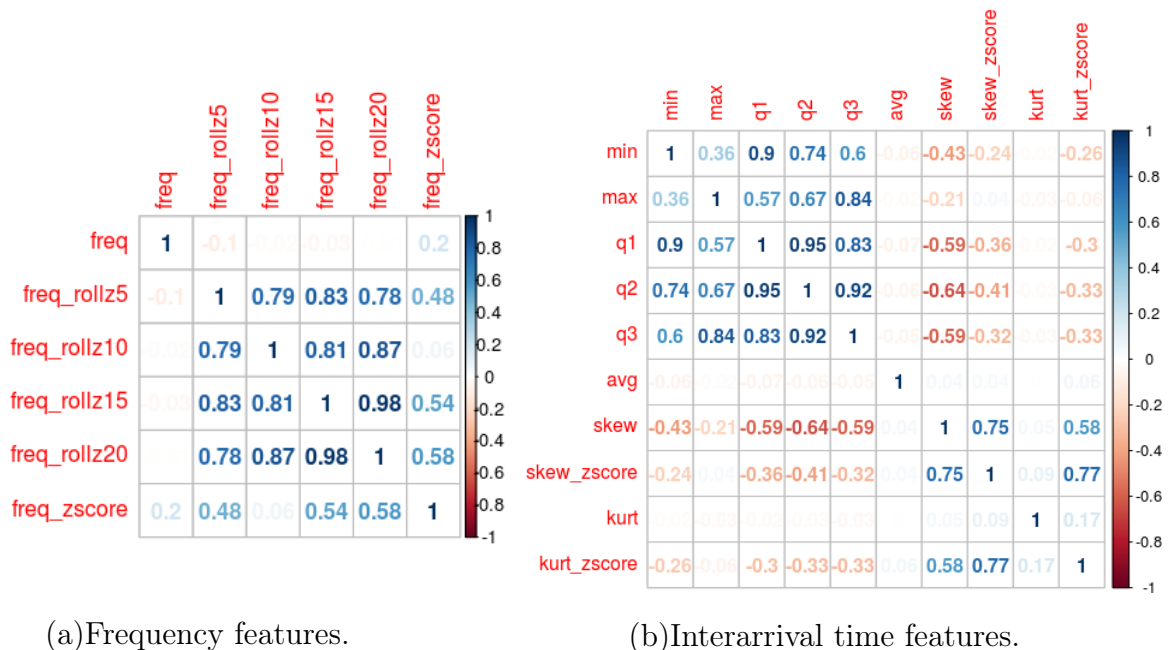


Figure 5.1: Matrices of features correlation.

5.3.2 System’s Historical Data

Analyzing and comparing new values with the historical data is very important to detect anomalies in the normal activity. However, obtaining historical social network data is difficult because there are limitations for retrieving messages in terms of queries per time, specific domains, disk space and time spent collecting data. In this sense, we did not have historical data, but we used our system’s historical data to compute and analyze variations with the new values. In fact, if we had not considered previous values and just used a screenshot of the current time-windows, it would have been an incorrect evaluation. This is because we would consider a new data as an isolated value without knowing the anomalies and variations with respect to the past values.

Based on the previous analysis, the attributes without knowledge of the system’s historical data were discarded such as the number of messages in each time-window ($freq$), quartiles ($Q1$, $Q2$, $Q3$), the shape distribution ($skewness$ and $kurtosis$) and so forth. Furthermore, if we considered attributes with a short knowledge of previous values, we found a lot of rows (in our dataset) with null values. Indeed, this is an important issue in our work because null values did not allow for computing $zscore$ normalization. For example, our attribute $freq_rollz5$ just considers the previous five past time-windows to compute $zscore$. But, according to Equation 4.1, $zscore$ has a denominator that contains the standard deviation of the population. Then, if the previous time-windows are short, we can often find null values and the standard deviation will be zero, and then the fraction will be undefined. The above explanation occurs because a lot of locations are small and without continuous social media activity. We did not also have the social media activity for each location used in this work, therefore, we did not compare the current activity with the historical activity. Finally, by removing the features based on

redundant and the previous values, we keep the next attributes in our dataset: *freq_zscore*, *skew_zscore* and *kurt_zscore*.

5.4 Labeled Emergency Situation Events

According to the methodology proposed in Section 4 and applying the features selection presented in Section 5.3, we generated different signals and removed several attributes to obtain our experimental dataset. Table 5.3 highlights an example of this dataset corresponding to the Chilean earthquake of December 2016.

Our filtering task can be seen as binary task, where the positive class (*detection label*) corresponds to messages related to instantaneous emergency situations, while the negative class (*nothing label*) corresponds to the remaining or non-related to crisis situations. However and as we explained in our methodology (Section 4), we actually work with bags of tweets divided into time-windows with different features. We then labeled time-windows instead of tweets or messages.

Each row of our dataset is labeled as positive class when the event date occurs inside of the current time-window. The exact event date and time was obtained from the National Seismology Agency in Chile⁵ and the National Institute of Geophysics and Volcanology in Italy⁶.

We defined that a certain time-window contained an *event detection* if it had a positive variation in frequency, skewness and kurtosis with respect to the normalization of the previous values. Moreover and according to (Figure 5.2), we included the three following time-windows after the event to compensate for the imbalance between classes, given that after these number of time-windows, the variation in the features decrease. Otherwise, the rows were labeled as negative class, meaning that they were not emergency events.

One observation in this step was that those time-windows corresponding the state hierarchy with label *class=True*, in the majority of the cases corresponding to the most affected locations by the emergency situations.

5.4.1 Under-Sampling

Given that an emergency situation is not a common event, we had a highly imbalanced dataset in terms of class labels. The number of time-windows analyzed for country and state hierarchy were equal to 2235 and 2889 respectively. The positive class (corresponding to *detection*) had $1\% \approx 2\%$ of time-windows about the total of our dataset. As a result of the analyzed data scattering (Figure 5.3), we used the country and state hierarchy separately, because the relationship of the features in each hierarchy were different.

To adjust the class distribution of the dataset, we applied the *under-sampling* technique. In this way, we used the functions provided for *ROSE* [36] to deal with binary classification problems in the imbalanced classes. This *under-sampling* technique was applied over

⁵<http://www.sismologia.cl/>

⁶<http://www.ingv.it/it/>

Time-window metadata				Attributes for classification				
Ti	Tf	Hierarchy	Location	Metadata	Freq zscore	Skew zscore	Kurt zscore	Class
14:24:00	14:30:00	country	chile	tweet text	1.9969	0.4603	0.1252	True
14:24:00	14:30:00	country	chile	user location	1.3472	-0.06795	-0.3868	False
14:24:00	14:30:00	state	biobio	user location	0.6022	-0.3482	-0.9066	False
14:24:00	14:30:00	state	los lagos	user location	6.0913	2.7235	1.5000	True
14:24:00	14:30:00	state	metropolitana	user location	1.5681	-0.17024	-0.5626	False
14:30:00	14:36:00	country	chile	tweet text	4.1259	0.7296	0.6863	True
14:30:00	14:36:00	country	chile	user location	1.9969	0.4603	0.1252	True
14:30:00	14:36:00	country	chile	ttext-ulocation	14.1338	2.6002	3.6949	True
14:30:00	14:36:00	state	biobio	tweet text	0.4795	-0.4153	-0.6542	False
14:30:00	14:36:00	state	la araucania	user location	6.3866	8.3375	12.03857	True
14:36:00	14:42:00	country	chile	tweet text	6.3233	3.3888	6.5373	True
14:36:00	14:42:00	country	chile	user location	3.7502	-0.1372	-0.1893	False
14:36:00	14:42:00	country	chile	ttext-ulocation	9.0203	1.5187	1.2517	True
14:36:00	14:42:00	state	los rios	tweet text	14.6172	1.3849	2.7308	True
14:36:00	14:42:00	state	los lagos	user location	13.5506	1.7388	0.5650	True

Table 5.3: An example of the dataset generated by the creation of the signals and removing attributes after features selection. The table shows the time-windows metadata and the attributes for classification. Class true identifies an emergency situation and class negative does not.

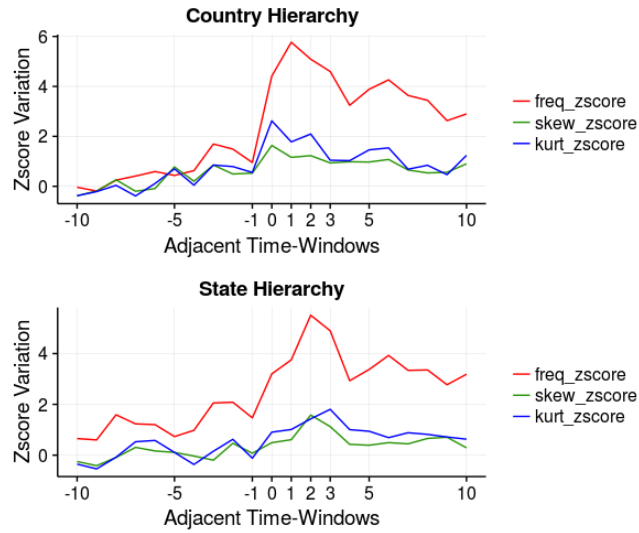


Figure 5.2: Average variation in emergency situations between time-windows. Positive and negatives values in x-axis represent the following and previous time-windows from the beginning of the event respectively.

country and state datasets separately. Then, we decreased our negative class which allowed us increased our positive class to $15 \approx 18\%$ in both cases. Finally, the number of analyzed time-windows for country and state hierarchy were equal to 200 and 500 respectively.



Figure 5.3: Relationship between features in country and state hierarchy. Red circles represent positive class (*detection*) and blue circles represent negative class (*nothing*).

Chapter 6

Supervised Experimental Analysis

According to the “data classification” module (Figure 1.1), we first trained a classifier to identify emergency events. Our filtering task can be seen as a binary classification task. The positive class (*detection label*) corresponds to time-windows related to instantaneous emergency situations, while the negative class (*nothing label*) corresponds to the remaining or non-related to crisis situations.

To classify messages, we employed traditional binary classifier Support Vector Machine (*SVM*) and Random Forest. We separated country and state in different datasets and set classification parameters independently.

Finally, we presented the results for the best classifier and chose the one with the best performance. Our first priority was to find the best recall value and later (as second priority) the best precision value.

6.1 Choosing a Machine Learning Classifier

One of the main tasks when using the machine learning techniques was to select the best attributes for representing data and chose the classifier for our dataset. In fact, this is a big open problem because there are currently a lot of algorithms with different configurations and parameters. For this reason, our efforts focused on choosing that classifier with the best performance for unbalanced data, small number of instances and a binary classification task.

Our experiments with supervised approaches were conducted using two classifiers: Support Vector Machine (*SVM*) and Random Forest. For each classifier, we tested different parameters and found the best balance between a high number of detections and a low number of false positives. Furthermore, we validated our results using the k-fold cross-validation, where $k = 5$, given that we had five earthquakes in our dataset. In this way, we exchanged between different earthquakes, evaluated the generated model and tried to identify the best same configuration between different folds. For each fold, we used four earthquakes for training and one for testing. Finally, we used the precision, recall and f1-score to evaluate each fold and later computed the average between folds for each measure.

6.1.1 Support Vector Machine

As explained in Subsection 3.1.1, there are a lot of configurations for setting the SVM classifier. For this reason, we performed a grid search for tuning SVM classifier. In our case, the grid search did not get good results when we applied the parameters computed by the function. However, this method helped us to know what parameters to use and what we could discard. For example, the grid search delivered us that the best kernel was polynomial in the case of the country hierarchy, but the other parameters were omitted for the most part.

According to Figure 5.3 where we analyzed the relationship among features for country and state hierarchy, we divided these hierarchies into two different experimental settings. For each fold in the cross-validation evaluation, we tested 23 different configurations for each hierarchy, which we explain briefly below:

- We used the four available kernels: linear, polynomial, radial and sigmoid.
- We used four values for the cost: 1, 5, 20, 50.
- We set both hierarchies with different values for the class.weights. On the one hand, the country hierarchy demonstrated a high difference between the ratio of the positive and the negative class (1 : 50 respectively). On the other hand, the state hierarchy had a small difference between them (1 : 5).

The above explanation is important because we needed to avoid that any time-windows in the country level were emergency events. Indeed, little events must not affect the social country activity with respect to the number of messages by time-windows. For this reason, we set a strong ratio between positive and negative class to avoid this behavior. More details on each configuration are shown in Table E.1 and Table E.2 in the Appendix E. After applying the 5-fold cross-validation, we got the following best configuration as shown in Table 6.1:

Table 6.1: The best performance found for country and state hierarchy using the Support Vector Machine classifier. The Precision (P), Recall (R) and F1-score (F1) are computed in order to know the performance of the models. More details of the each k-fold evaluation can be seen in Table E.3 and E.4 in Appendix E.

kernel	C	gamma	coef0	degree	weights	P	R	F1
<i>Country Hierarchy</i>								
polynomial	20	2	2	3	1:50	0.308	0.855	0.450
<i>State Hierarchy</i>								
linear	1	-	-	-	1:5	0.358	0.860	0.474

As explained above, we separated the country and state hierarchy and trained each model independently. In fact, the results in Table 6.1 show that each hierarchy level had different configurations. On the one hand, country hierarchy was based on a polynomial kernel, where the degree was 3. On the other hand, state hierarchy just considered a simple lineal kernel to divide and separate both types of classes in our dataset. Figure 5.3 explains this behavior where the values in the country hierarchy were more scattered than in the state hierarchy.

Also, the shape distribution in both hierarchies were different and for this reason we evaluated them with many kernels in our grid search.

6.1.2 Random Forest

The Random Forest method also is used for classification and regression. This algorithm works on a multitude of decision trees for training and outputting the class that is the mode of the classes between the trees. In this way, Random Forest’s parameter describes the number of trees to grow. Other parameters used for this algorithm are the following: the number of variables randomly sampled as candidates at each split (also known as *mtry*); and the cutoff, which represents a vector of length equal to number of classes. The winning class for an observation is the one with the maximum ratio of proportion of votes to cutoff.

Unlike the SVM classifier, we did not use a function for tuning the parameters because this method just tuning the *mtry* parameter. However, and as done with the previous classifier, we divided the country and state hierarchy, but we used the same parameters for both. The main reason was that there were not many settings to configure. We therefore set the parameters as we explained here:

- We tested with eight different values for the number of trees between 10 and 10,000.
- For the number of variables randomly sampled as candidates at each split (*mtry*), we set the value $mtry = \sqrt{p}$, where p is the number of variables in the dataset. In our case $p = 3$.
- For the cutoff value, we used $1/k$ as the default value set by the function for classification, where k is the number of classes. In our case $k = 2$.

After applying the 5-fold cross-validation, we got the following best configuration, as seen in Table 6.2:

Table 6.2: The best performance found for country and state hierarchy using the Random Forest classifier. The Precision (P), Recall (R) and F1-score (F1) were computed in order to know the performance of the models.

trees	OBB error	P	R	F1
<i>Country Hierarchy</i>				
50	8.5	0.75	0.301	0.429
<i>State Hierarchy</i>				
10	9.1	0.268	0.614	0.378

Table 6.2 also includes the average of the OBB error (out-of-bag) computed in the k -fold cross-validation. This value represents the average error for each z_i calculated using predictions from the trees that do not contain z_i in their respective bootstrap sample. This allows the Random Forest classifier to be fit and validated whilst being trained.

In the same way as the SVM classifier, we also found a simpler configuration in the state hierarchy than the country hierarchy. In the first model named (for state hierarchy), we

found the best performance when we set 10 trees, because in highest values our model had overfitting given that the values of the OBB, precision, recall and f1-score were the same between different configurations. In the second model named (for country hierarchy), our best performance was when we set 50 trees for the forest creation. For the state hierarchy, here we also had overfitting when we used highest values for the number of trees.

6.2 Summary of the Supervised Results

As a result of our evaluations in two different classifiers, we needed to choose the best performance between them. Our first priority was to find that model with a high number of detections based on the real number of detections: in other words, a high recall value. However, we also needed a low value of the false positive detections. In fact, this is important because we were able to identify non-emergency events as crisis situations (e.g., to detect a soccer match or a TV show as an emergency situation). As a second priority, we also needed a high precision value with the goal of decreasing the number of false positive detections.

According to the results presented in Table 6.1 and Table 6.2, we found out that the best model was the SVM classifier. Indeed, this choice was for country and state hierarchy because in both cases we found a good balance between the precision and recall values, explained below:

- For country hierarchy, we found the best recall value in the SVM classifier. However, in the Random Forest classifier we found the best precision value. As we explained above, our first priority was a good value for recall and as second priority a good value for precision. For this reason we chose the SVM classifier for this hierarchy. In the next chapter, we will explain a methodology evaluation for increasing the precision value using the geographic spread.
- For state hierarchy, we also chose the SVM classifier because both precision and recall values were much better than the Random Forest classifier.

We additionally computed the Sensitivity and Specificity values to visualize the ROC curve. Using our classifier previously selected, we plotted the ROC curve for each fold and the country hierarchy. As we see in Figure 6.1, every ROC curve had an AUC value about 0.85 that indicates a good performance between the True Positive Rate (Sensitivity) and the False Positive Rate (1-Specificity). Furthermore, Figure 6.1 shows the most balance when the False Positive rate was 0.2 and the True Positive Rate was 0.8. In this case, this balance indicates that we had 20% approximately of false positives.

In the same way that the country hierarchy, we also computed the ROC curve for the state hierarchy. As we see in Figure 6.2, our results were slightly better than the country hierarchy because the AUC values were about 0.9. We also estimated the balance between Sensitivity and False Positive Rate. In this case, Figure 6.2 shows the most balance when the False Positive Rate was 0.1 and True Positive Rate was 0.9.

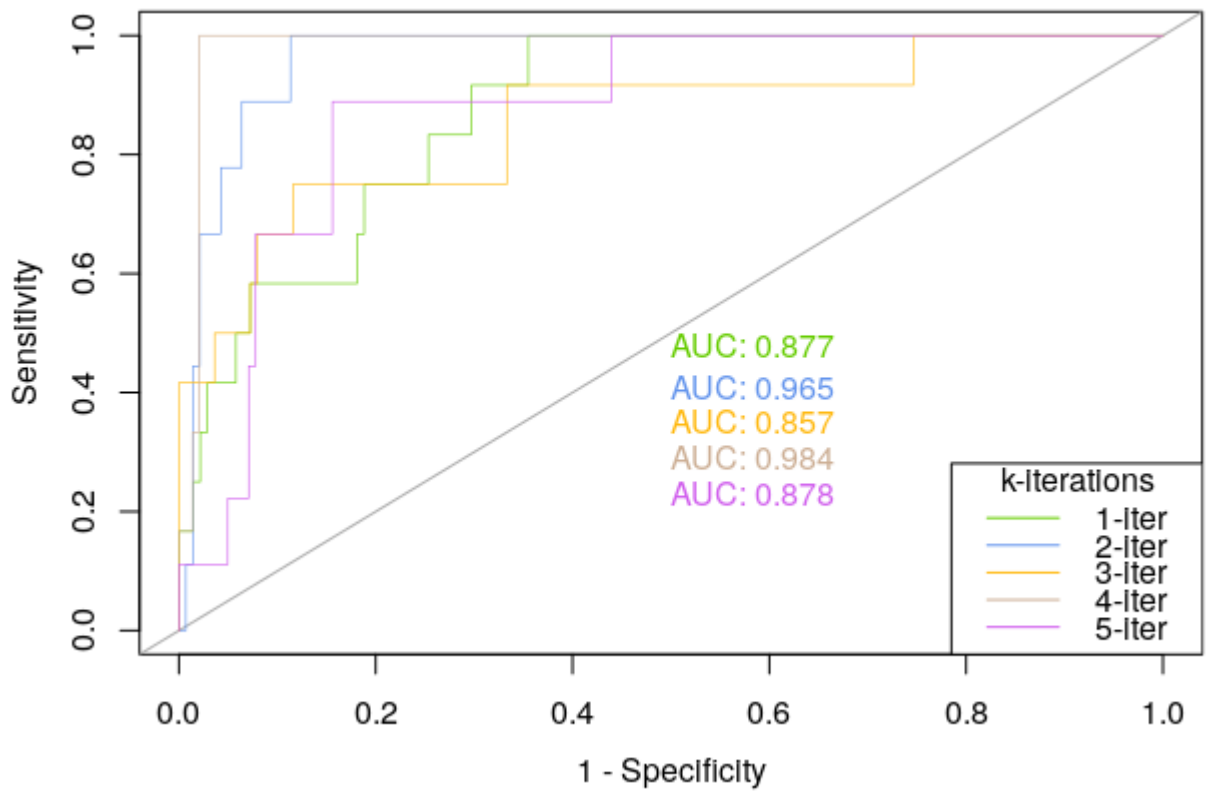


Figure 6.1: The ROC curves for each fold in the SVM classifier for country hierarchy. The x-axis label (1-Specificity) represents the False Positive Rate (FPR).

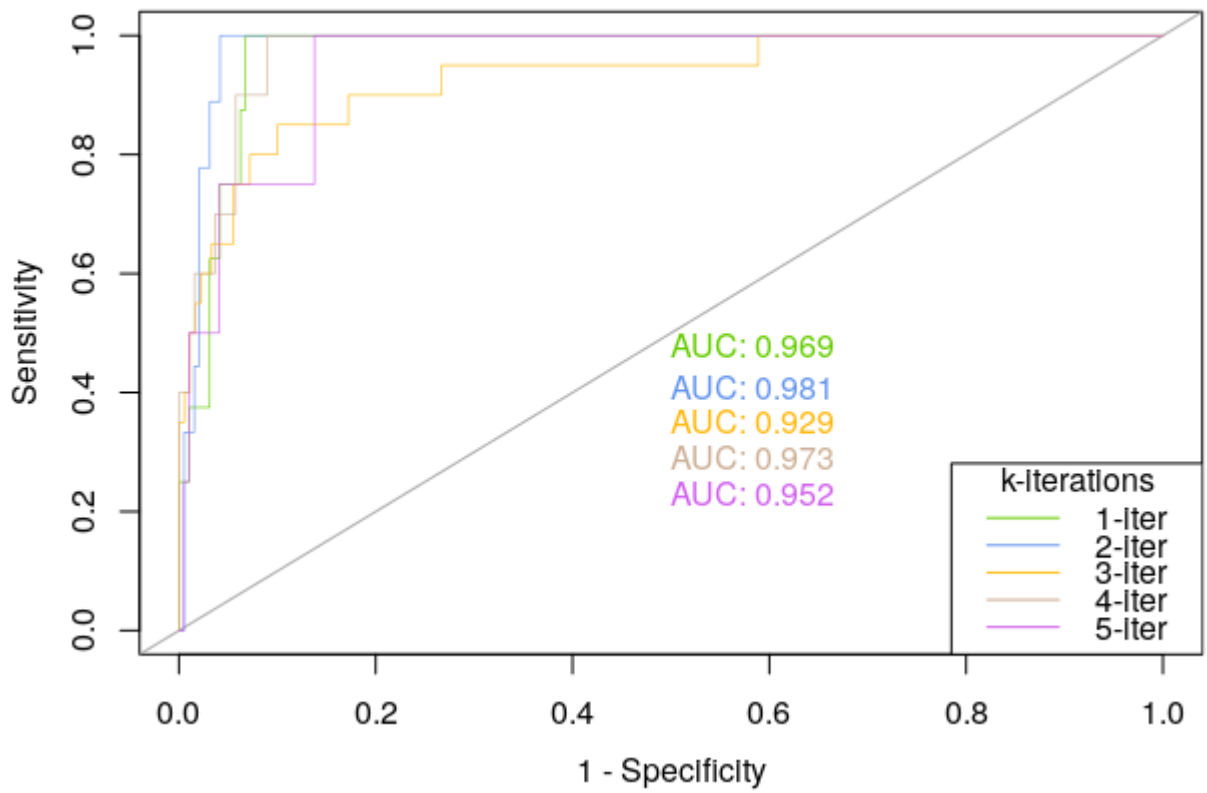


Figure 6.2: The ROC curves for each fold in the SVM classifier for state hierarchy. The x-axis label (1-Specificity) represents the False Positive Rate (FPR).

Chapter 7

Evaluation

In this chapter, we present an extensive evaluation using the model that had the best performance in the previous section. We validated our model using three types of evaluations after applying our classifier presented in the above chapter. First, we consider that an emergency situation is an isolated event between different levels of the hierarchies. Second, we add the dependencies between the country and state level when an emergency event occurs. Third, we include the geographic spread in order to know the proximity of the affected locations and for decreasing the number of false positives detection. And fourth, with the best performance found in the ground truth evaluation, we also evaluated our model in other kinds of crisis events occurred in England.

7.1 Ground Truth

7.1.1 Independent Analysis of Hierarchies

Our first analysis just considered the hierarchies as isolated detections. Figure 7.1 shows the results considering only the prediction over each instance in our datasets. As we noted in Table 5.3, each instance (or row) in our dataset is one specific hierarchy and metadata level with its corresponding attributes for classification.

As noted above, the assignation from the lowest level (city) to the highest (country) in the gazetteer hierarchy generated high frequency of messages, which caused multiple *bursts* in our country signal for non emergency situations. This concept can explain the values of Precision (P) and FPR in Figure 7.1.

In addition to the analysis of the number of detections by labels, we also studied the number of detections by time-window. For this analysis we aggregated the hierarchies by time-window and computed whether or not all instances were positives in the current time-window. This means that for each hierarchy in one specific time-window, we analyzed whether or not the classes were positives for each metadata-level. If all instances were positives in the time-window, the time-window was correctly assigned as positive. An example of this evaluation is shown in Figure 7.2, where we had four instances for country and state hierarchy and for different metadata-levels.

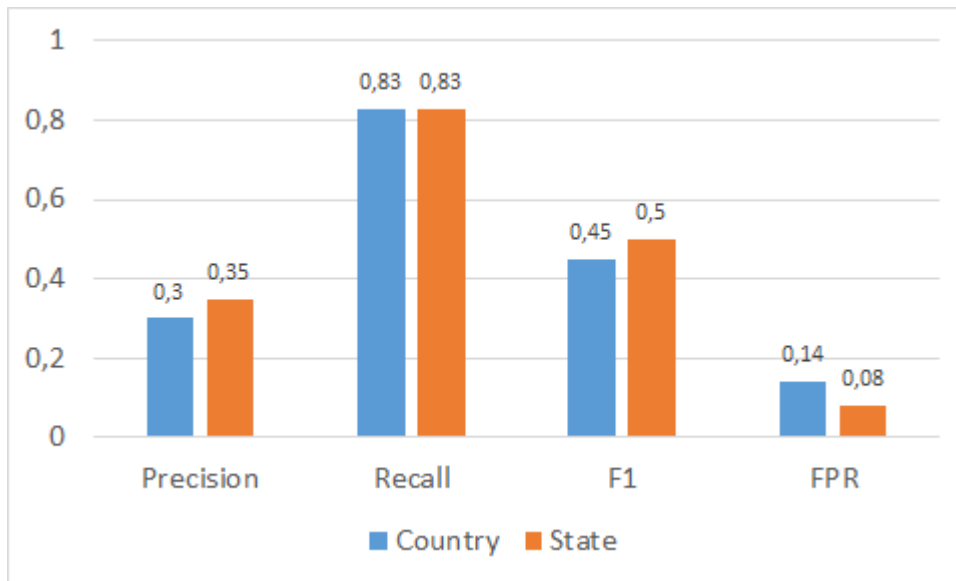


Figure 7.1: Average performance of 5-fold cross-validation by hierarchy independently just using labels.

According to the results shown in Figure 7.3, when we analyzed country and state independently the values of Precision, F1 and FPR had worse values than the analysis by label. These results can be explained because we considered every location in the state hierarchy and aggregated them counting the number of positive classes. However, an emergency could not affect all locations in this hierarchy. For example, an emergency could occur in Valparaiso, but not in Maule state (Figure 4.7).

7.1.2 Dependent Analysis of Hierarchies

Our second analysis considered the hierarchies as non-isolated detections. In the results explained above, we considered country and state hierarchy independently, which was not a correct analysis because an emergency situation affects states and country at the same time. For this reason, we inspected the time-windows where all metadata-level for country and state hierarchy had a correct detection simultaneously. An example of this situation is shown in Figure 7.4. As well as the independent analysis presented above, we aggregated the hierarchies by time-window and computed whether all instance are positives in the time-window. However, after classifying the hierarchies as a positive or negative class, in this evaluation we compared whether or not both hierarchies were positives simultaneously. We then determined whether the time-window corresponded to one detection. In our example, the state hierarchy had a false class and the country hierarchy had a true class. For this reason, the time-window was labeled as a false value.

The results are shown in Figure 7.5. In contrast to the independent analysis of country and state, we improved the Precision, F1 and FPR values as a consequence of a smaller amount of the time-windows related to non-emergency situations were assigned as detection. However, when we see the value obtained for FPR ($FPR = 0.03$), this rate represents an incorrect number of time-windows assigned as detection equal to 23. This means that we had 23 new emergency situations detected by our method.

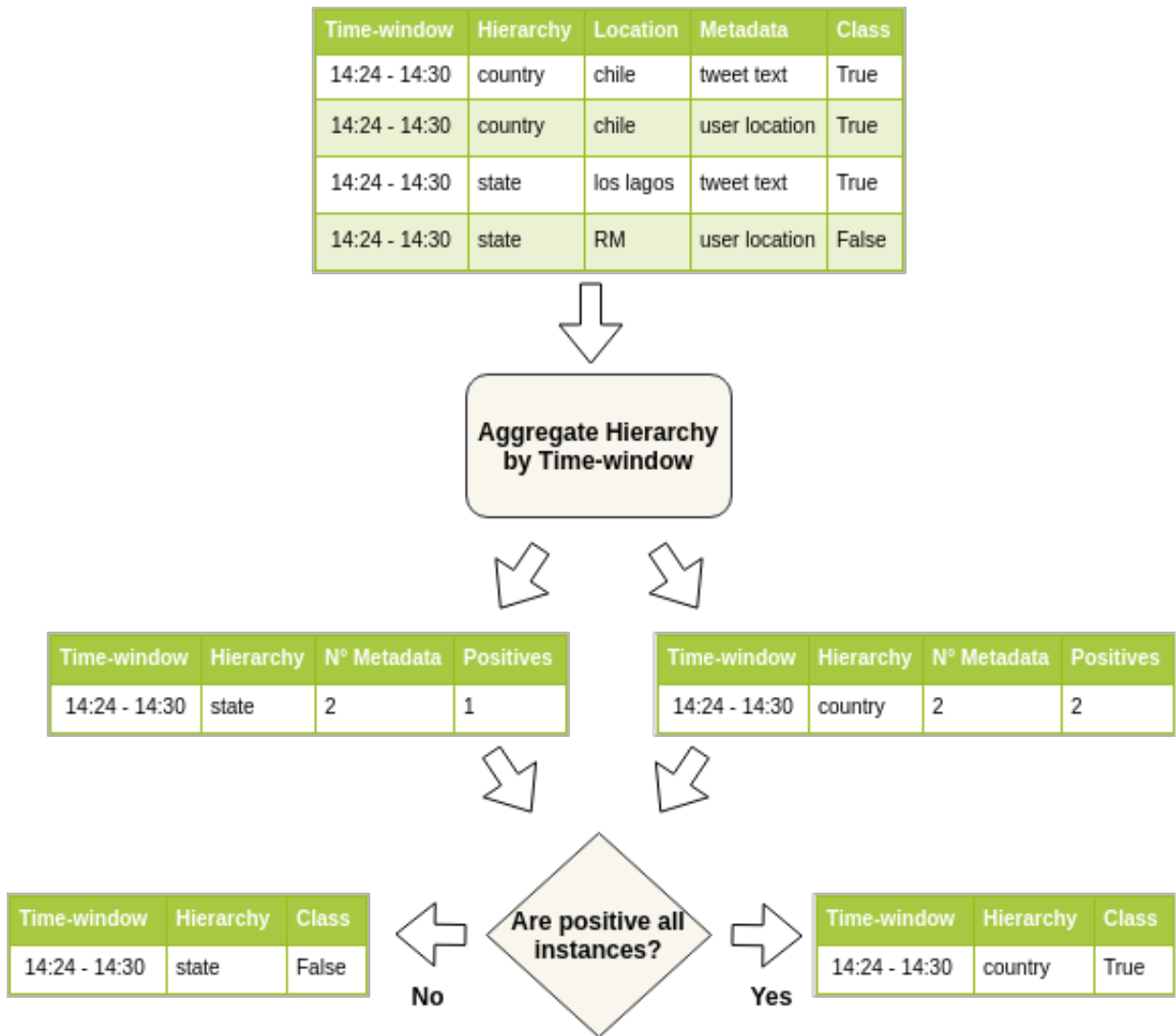


Figure 7.2: An example of evaluation for independent analysis of hierarchies.

7.1.3 Geographic Spread Analysis

In addition to the results of the dependency analysis explained above, we saw that a large amount of time-windows for country hierarchy ($\approx 82\%$) had more than one metadata-level when exist a correct detection. This can be explained since an emergency situation produces a collective reaction on the level of body of the message (*tweet text*), users sharing any messages with profile location in a specific country (*user location*) or mixing both concepts (*tweet text - user location*). For this reason, our third analysis considered the hierarchies as non-isolated detections and applies the Geographic Spread (G.S.). Using the *Adjacency Matrix* to represent neighborhoods between regions/states, we considered as a correct detection those time-windows where the state/s classified as detection were defined as *Focalized* or *Diffused* and exist dependency between hierarchies.

Figure 7.6 represents an example of this evaluation. As well as the previous analyses, we grouped the hierarchies by time-window. However, before determining if all instances were positives, we applied two kinds of filters for state and country. For state hierarchy, we

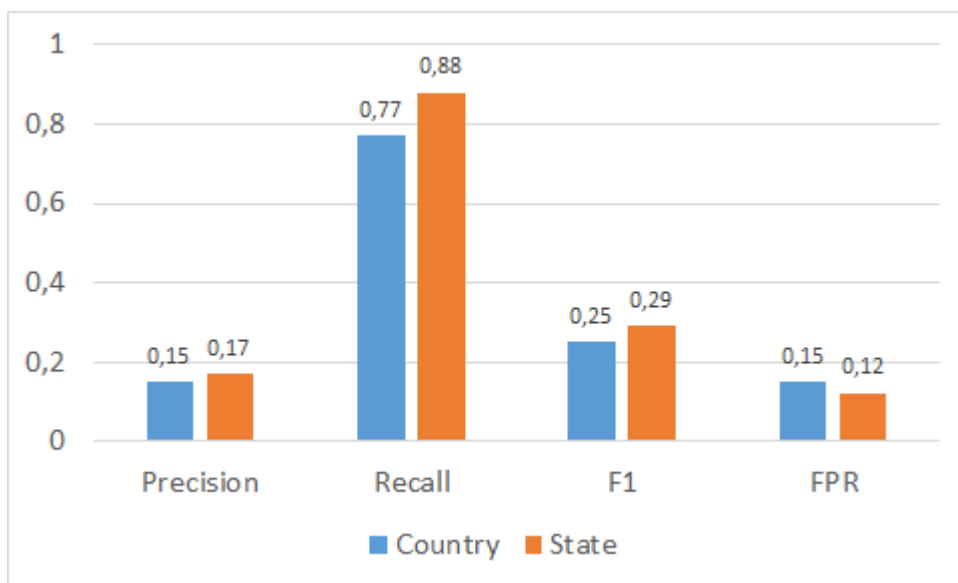


Figure 7.3: Average performance of 5-fold cross-validation by hierarchy independently just using time-windows.

applied the geographic spread to filter non-isolated states when a detection was identified. Here, our filter determined if an event could be focalized or diffused. For country hierarchy, we considered a strict (two levels) or soft (three levels) evaluation related to the number of metadata-levels identified by time-window. The following steps were similar to previous evaluations.

Considering the geographic spread by states and the number of metadata-levels by country hierarchy, we analyzed the results shown Figure 7.7. On the one hand, the *Country(2) + State + G.S.* represents the detection when we considered at least two metadata-levels (soft evaluation) for the country hierarchy and the geographic spread for states. In contrast to the previous analyses, we improved the values of the Precision, F1 and FPR. The last metric was very important because there were no time-windows incorrectly assigned as emergency situations. Consequently, the Recall values decreased which means that our method removed some time-windows classified as detection. Beside the percent of emergency situations detected was equal to 100% with an average delay equal to 10.4 minutes ($min = 6$, $max = 14$) from the impact of the event to the first detection.

On the other hand, the *Country(3) + State + G.S.* (strict evaluation) represents the detection when we considered three metadata-levels for country hierarchy and the geographic spread for states. Similar to *Country(2) + State + G.S.*, we improved the values of Precision, F1 and FPR but our recall decreased from $R = 0.64$ to $R = 0.47$, detecting 80% of the emergency situations with an average delay equal to 11.5 minutes ($min = 8$, $max = 14$) from the impact of the event to the first detection.

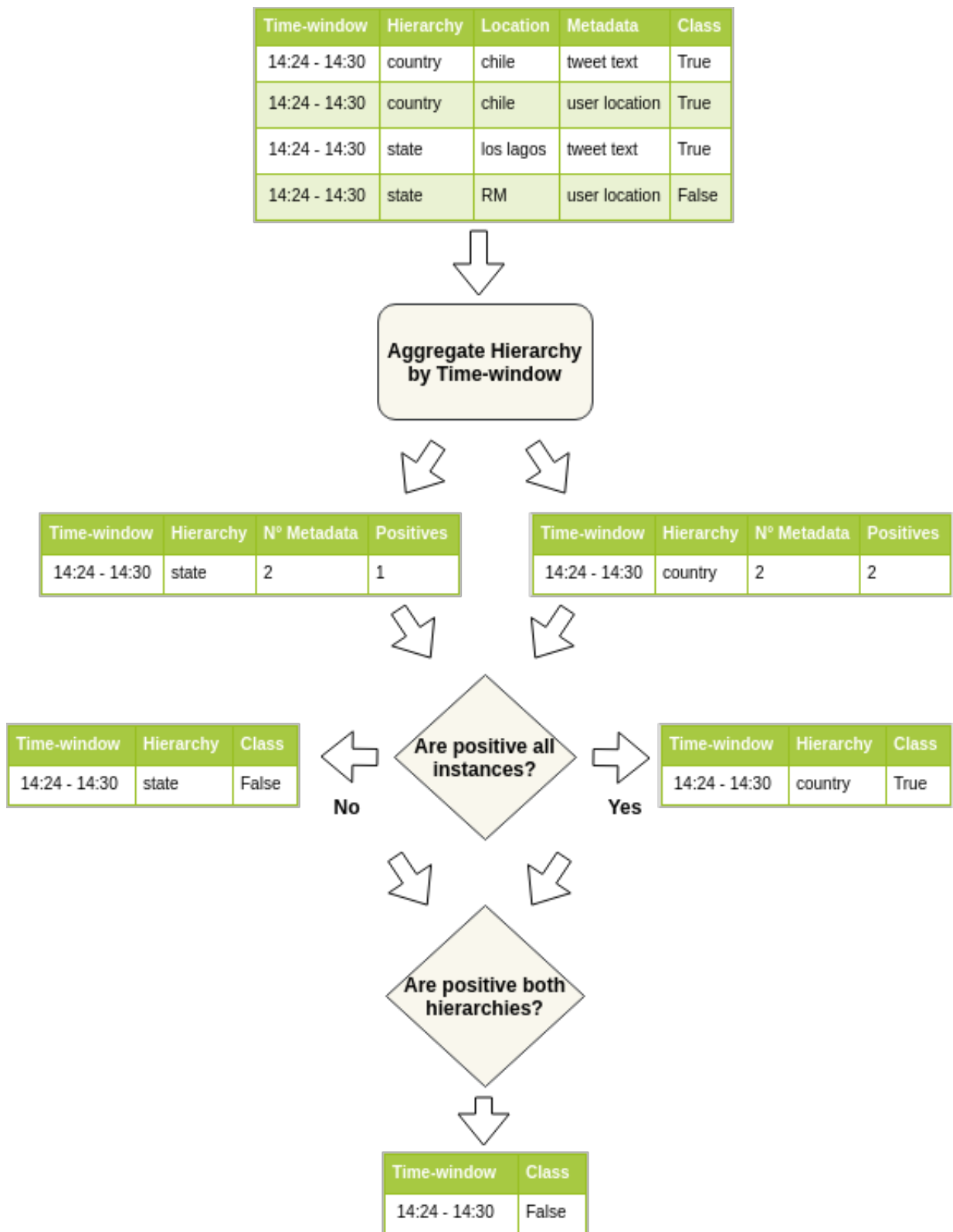


Figure 7.4: An example of evaluation for dependent analysis of hierarchies.

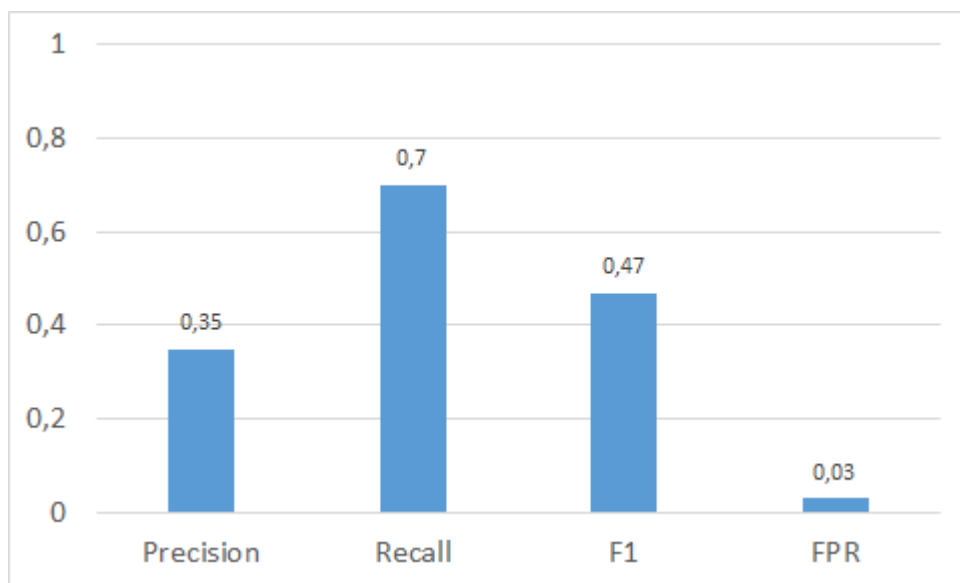


Figure 7.5: Average performance of 5-fold cross-validation by hierarchy dependently just using time-windows.

7.2 On-line Evaluation

For our evaluation in the Twitter Public Stream, we trained a classifier with five earthquakes identified in our ground truth. Furthermore, our on-line evaluation dataset is formed by eight different events that occurred in England between December 2016 and October 2017. For each event we considered the full-day in which they occurred. The main goals of this evaluation was to know the capacity of our method to detect emergency situations and discard those non-related to emergency events that involve location references. The geographic spread analysis was used to evaluate our method because it decreases the number of false positives detection. In the same way of the experiments in Section 7.1.3, we compared the results using the two presented methods with respect to the number of metadata-levels by country hierarchy.

As can be noted on Table 7.1 and Table 7.2, we studied three terrorist attacks and five high-impact real-world events related to soccer matches, music concerts and political elections. In the case of the terrorist attacks, we studied these types of events since that according to Carr [11], these crises were identified as *instantaneous-focalized* events, where an unexpected event affects the community, but in a reduced area. Unlike to earthquakes (identified as *instantaneous-diffused* events), we studied the capacity of our classifier to detect another type of event where the number of affected people is smaller than earthquakes, tsunami, and others *instantaneous-focalized* events.

For *Premier League Soccer Matches* and *U.K Elections*, we can not identify the beginning of the event, since in the first one there are many soccer matches during the analyzed day and in the second one there is no a specific start time. In order to know the topics when our method detects an event, we computed the Top 3 Bigrams in the detected time-windows. Also, we calculated the delay time for emergency events since the beginning of the event until the first detection.

Table 7.1: On-line evaluation of events occurred in England by time-windows (T-W) using Country(2)-State + G.S. method. The table shows the total number of detected time-windows, the number of detected time-windows before the beginning and after to the end of the event. The last two columns show the detection delay time with respect to the beginning of the event and the top 3 bigrams when the detection occurs.

Event	Detected	Before	After	Delay	Top 3 Bigrams
Premier League Soccer Matches	2	-	-	-	(man, utd), (new, year), (happy, new)
Westminster Terrorist Attack	13	0	13	32	(stay, safe), (terror, attack), (safe, everyone)
Manchester Terrorist Attack	12	1	11	23	(ariana, grande), (incident, arena), (grande, concert)
London Terrorist Attack	14	7	7	36	(stay, safe), (incident, bridge), (borough, market)
U.K. Elections	5	-	-	-	(theresa, may), (vote, labour), (van, dijk)
Adele Live in Wembley	9	7	2	-	(elland, road), (new, times), (phil, jackson)
England vs Slovenia Soccer Match	4	4	0	-	(simon, brodikin), (join, us), (theresa, may)
Metallica Live in London	4	4	0	-	(always, said), (chance, win), (carabao, cup)

On the one hand, the first evaluation *Country(2)-State + G.S.* had full detection of the terrorist attacks with average delay time equal to 30.3 minutes. These detections were related to the event given that the bigrams represent terms associated with crisis situations. However, the *London Terrorist Attack* has 50% of the detected time-windows after the event, which means that there were seven time-windows non-related to emergency situations. Besides the crisis situations analysis, we also studied the number of detected time-windows in non-related to emergency situation events. In the same way, we had a large amount of misclassified time-windows that do not represent crisis situations as we can see in the Top 3 Bigrams for each non-related to event.

On the other hand, the second evaluation *Country(3)-State + G.S.* decreased the number non-related to emergency situations events detected as crisis situations. We saw three time-windows in two events detected as emergency situations (*England vs Slovenia*, and *Metallica Live in London*). In these cases, the time-windows were detected before the event and corresponded to non-emergency situations according to the bigrams. Furthermore, when we

Table 7.2: On-line evaluation of events occurred in England by time-windows (T-W) using Country(3)-State + G.S. method. The table shows the total number of detected time-windows, the number of detected time-windows before the beginning and after to the end of the event. The last two columns show the detection delay time with respect to the beginning of the event and the top 3 bigrams when the detection occurs.

Event	Detected	Before	After	Delay	Top 3 Bigrams
Premier League Soccer Matches	0	-	-	-	
Westminster Terrorist Attack	4	0	4	32	(terror, attack), (stay, safe), (terrorist, attack)
Manchester Terrorist Attack	2	0	2	23	(ariana, grande), (praying, everyone), (everyone, affected)
London Terrorist Attack	1	1	0	-	(ariana, grande), (around, world), (lady, gaga)
U.K. Elections	0	-	-	-	
Adele Live in Wembley	0	0	0	-	
England vs Slovenia Soccer Match	1	1	0	-	(per, day), (menswear, sample), (closed, roads)
Metallica Live in London	2	2	0	-	(happy, birthday), (chance, win), (always, said)

analyzed the number of the detected emergency situations, two-thirds (66%) of the events were detected correctly with average delay time equal to 30.3 minutes. In the case of *London Terrorist Attack*, our method detects one time-window before the event but the bigrams described that the detections do not correspond to crisis situations.

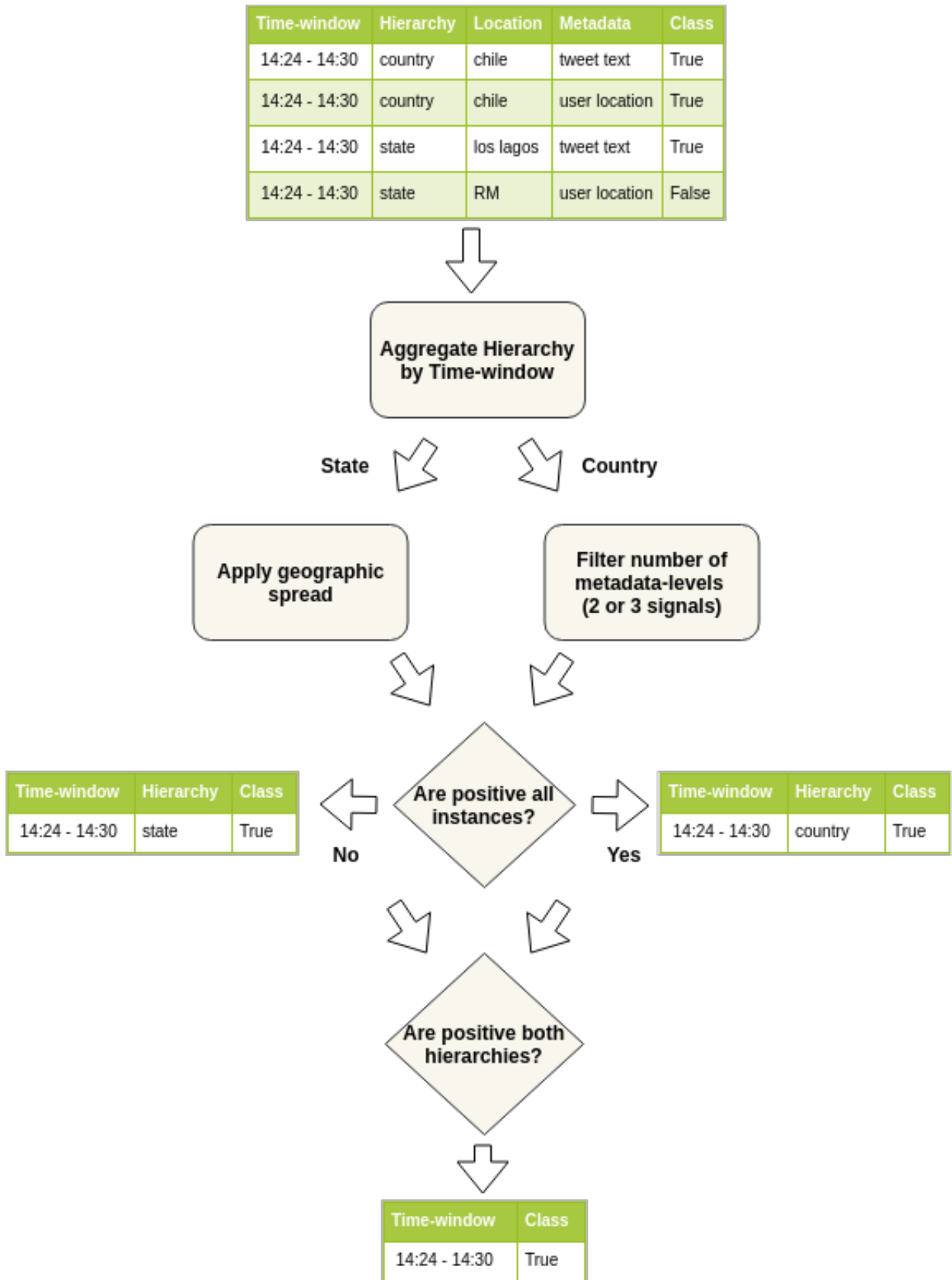


Figure 7.6: An example of evaluation for dependent analysis of hierarchies with geographic spread.

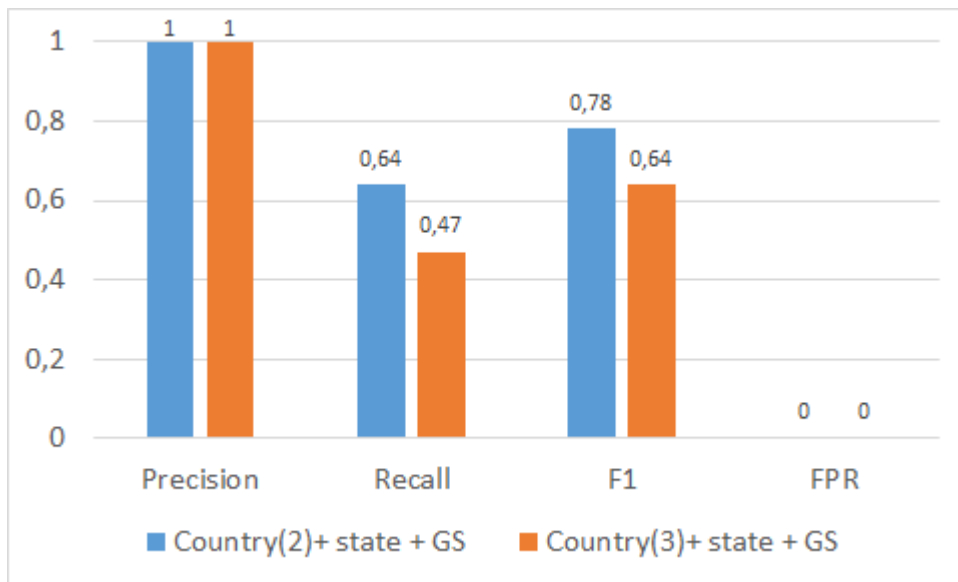


Figure 7.7: Average performance of 5-fold cross-validation by hierarchy dependently with geographic spread just using time-windows.

Chapter 8

Discussion and Conclusion

Our findings show that we can now detect a candidate emergency situation based on anomalies in the frequency of location mentions. Our method also detected 80% of events related to emergency situations as we demonstrated in our ground truth experiments. Furthermore, our approach is independent of textual features because we applied the model over different languages such as Spanish, Italian and English. We tested our model in different types of crisis events such as earthquakes (EQ) and terrorist attacks (TA), where these are identified in the literature as *instantaneous-diffused* and *instantaneous-focalized* events respectively. We also applied our methodology on different magnitudes (in the case of earthquakes) and number of affected people (e.g., *Manchester Terrorist Attack* and *Westminster Terrorist Attack*).

Number of active users per country

Countries have different number of active users depending of several factors may bias on-line data: (1) Geopolitical issues with respect to prohibitions against Twitter in several countries such as China and Iran. (2) There are another micro-blog platforms more popular than Twitter. For example, Sina Weibo in China. (3) There are socioeconomic issues raise barriers to Twitter (and the larger Internet) in rural areas [43].

In our experiment with United Kingdom¹, the number of active users in this country could affect detections because there is a high daily level of social media activity. Therefore, in order to detect bursty activities in microblog messages, we require a high variation with respect to the average daily social media activity.

Locations with similar names

There are locations with similar names in other countries. For example, the city *York* in England and the city *New York* in United States 8.1. So, if we inspect a micro-blog message with an incorrect country location, our method could detect anomalies in the wrong place. However, using the idea for detecting anomalies in different levels of the hierarchy, we reduced these kinds of false positives.

¹<http://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/> visited on January 2018

Table 8.1: Example of messages that include similar locations names.

Raw text	After preprocessing
God Bless All Huracan Sandy Victims In New York And New Jersey, Thousands Of people Are Without Power Right Now! God Protects Those People!	['god', 'bless', 'huracan', 'sandy', 'new', ' york ', 'new', 'jersey', 'thousand', 'people', 'power', 'right', 'protects']
Why are some universities (York) super hard to get into	['why', 'are', 'some', 'universities', ' york ', 'super', 'hard', 'get']

Soccer team that include city names

Mostly soccer teams (and other kinds of sports) have names that include city names ². In fact, “club names may reflect the geographical, cultural, religious or political affiliations” ³. In Europe, many clubs are named after their towns or cities such as *Liverpool F.C* or *Hamburger SV*. In our study, we evaluated the method in England, where there are vastly soccer teams with location names. In this sense, if two soccer teams play, social networks react with a huge number of messages during the game (Figure 8.1). In fact, only did we find collective reactions in the text, but also collective reactions from English users.



Figure 8.1: A micro-blog message related to Manchester City Football Club.

Regarding the geographic spread where we defined an emergency situation as diffused or focalized, we found evidence that differentiates each type of event. In the case of diffused events, the delay time of the our first detection was less than 12 minutes. In focalized events it was greater than 30 minutes (Figure 8.2). This explains that in diffused events such as earthquakes, a high number of people are affected (thousands or millions) at the same time by an event which generates a collective reaction in social media in the locations where the event happened. In Figure 8.2, we see that earthquakes have at least two detected locations in the first detection (except Italy EQ2). In contrast, focalized events have less eyewitnesses (hundreds or thousands), then when the users share messages in social media, where the frequency does not affect the average daily message of the country in the first minutes. This can be explained in Figure 8.2 where the terrorist attacks have just one detected location in the first detection. Additionally, the delay time can be different for many reasons: datetime of the event (for example, during the early hours), few differences with the end of the current time-window, type of the affected locations (rural or urban cities) and the number of active users by locations.

²<https://blog.oxforddictionaries.com/2016/01/15/football-team-names/>

³https://en.wikipedia.org/wiki/Association_football_club_names

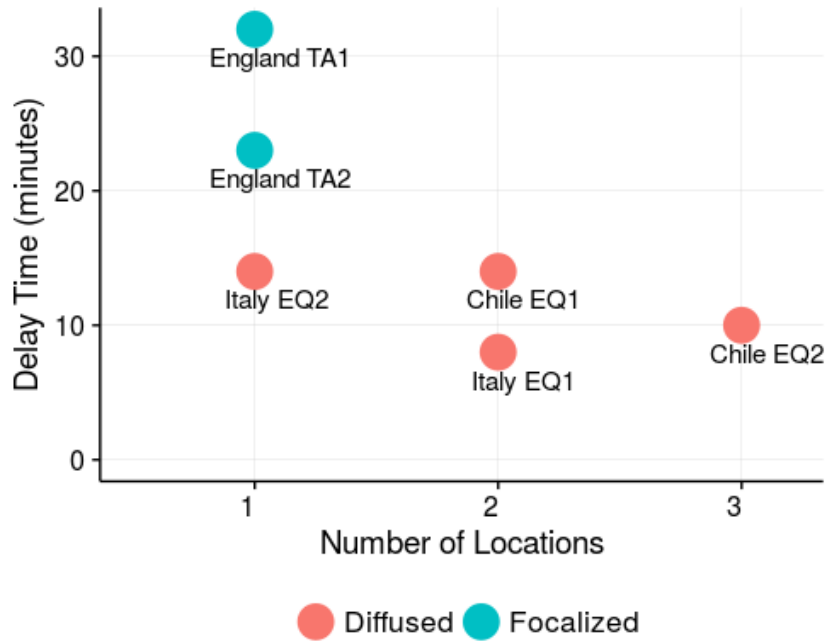


Figure 8.2: Relationship between delay time and number of locations in the first detection for diffused and focalized emergency situations. Earthquakes are labeled as *EQ* and terrorist attacks as *TA*.

8.1 Unsupervised Experimental Analysis

Unlike that presented in Section 6, where we used a supervised approach to detect emergency situation candidates, we also applied methods based on clustering to generate time-windows groups related to emergency and non-emergency events.

We used two types of unsupervised algorithms to apply clustering in our dataset. First, we employed the clustering centroid models as *k-means* and *k-medoids* algorithms. These types of models are characterized mainly to set the k number of generated clusters. Second, we employed clustering density models as *DBScan*. These models, instead of setting the number of generated clusters, are based on density of points and use mainly two parameters: *epsilon*, which specifies how close points should be to each other to be considered a part of a cluster; and *minPts*, which specifies how many neighbors a point should have to be included in a cluster. Furthermore, we validated the quality of the generated clusters using external criteria because we had labeled data to use as ground truth.

8.1.1 Clustering Centroid Models

The main task in using these models in the data set involves finding and setting the number of k clusters. In our case, given that we had just two classes, our k number of clusters was reduced to $k = 2$. With this type of model, we expected to generate two clusters, where each cluster has the positive and negative class respectively.

K-means

To ensure the correct selection of the k number of clusters, we estimated this value using the most common methods for it. We then computed the total within-cluster sum of squares and the average silhouette width, where several values of k between $2 \leq k \leq 15$ were tested. Figure F.1, in Appendix F shows the elbow method (for computing the sum of squares) and the average silhouette width for country and state hierarchy. In both cases, the best value of the number of clusters is $k = 2$.

K-medoids

Unlike the previous method, in the k-medoid method we can set the different distance functions to apply over the elements in the dataset. Some of these distance functions are Manhattan, Euclidean, Minkowski, etc. We also estimated the k number of clusters, but just based on the average silhouette width. Furthermore, we again used the same values of k between $2 \leq k \leq 15$. Figure F.1 in Appendix F shows the average silhouette width for country and state hierarchy.

In general terms and for different distance functions, our estimations show that the best value was $k = 2$. Figure F.2 and F.3 in Appendix F explain this result for Euclidean and Manhattan distance respectively.

8.1.2 Clustering Density Models

As we explained at the beginning of this section, we use density models to generate groups of points. Unlike the centroid models, we do not need the k number of clusters as parameter. These types of models are very useful when we need to find outliers and clusters without a specific shape. For this reason we considered an emergency situation as a not common event that occurs in social media.

DBScan

DBScan algorithm is mainly based on two parameters: the *epsilon* (also known as **eps**) which specifies how close points should be to each other to be considered a part of a cluster; and the *minPts*, which specifies how many neighbors a point should have to be included in a cluster.

To determine the epsilon parameter, we applied the knee method which computes the distance between points using the number of k nearest neighbors. The k value was set using the number of features plus one, hence our experiment yielded $k = 4$.

Similar to the k-medoids method, we computed different measures of distance as Euclidean, Manhattan and Minkowski, and computed the knee method. Figure F.4 in Appendix F shows an example of this estimation.

8.1.3 Summary of the Unsupervised Results

In this section, we defined and applied the most common methods for generating the number of clusters. We also used different distance measures and parameters in each algorithm.

Afterwards, we evaluated the performance of each based on the external criteria given that we had labeled time-windows in our dataset.

The main measures that we used to evaluate the external criteria were the following: purity, entropy, normalized mutual information (also known as NMI) and normalized variation of information (NVI). To explain results, we separated each algorithm by hierarchy because each model has different parameters. The results of these evaluations are in Table F.1, Table F.2 and Table F.3 in the Appendix F.

Before interpreting the results, we needed to identify the relevance of our measures to evaluate the quality of the groups after clustering. According to the findings presented by Wu et al. [58], the most commonly used external measure to evaluate clusters such as entropy, purity and mutual information do not work well for different data distributions. In our case we had a high unbalanced data with respect to the number of instances for each class (positive and negative classes). Therefore, the entropy and purity did not represent good measures, even if they represent good values in our results. Hence, we just considered the NMI and NVI to evaluate the quality of the clusters according to the same work presented by Wu et al. [58] where they demonstrated that NMI and NVI work fine with unbalanced data.

For example, we saw good values for entropy in the algorithms evaluations, where a good value was near zero. In fact, if we considered only this measure, we expected good results. But if we inspected the results for NMI and NVI, we had bad results in the quality of the clusters because they were near zero and one respectively. In the same way, if we also inspected the purity measure in our results, we saw good values (near one). Likewise, the values for NMI and NVI were bad and the purity measure did not work well as a good quality measure.

As a result of the different evaluations with three algorithms, parameters and configurations, we did not identify good quality values in the clusters generated in this process. The main reason was the low amount of the positive class (in our case, detection label) that generated an unbalanced data. Therefore, we discarded the use of the unsupervised approach to find and separate between time-windows related to emergency situations and non-emergency events.

8.2 Final Comments and Future Work

To conclude, in this work we presented a methodology for detecting emergency situations based on locations for a specific country. We showed that the users have collective and self-organized patterns in the affected locations when an emergency situation occurs. Furthermore, this approach is independent of the textual features in the classifier and can be used in different types of events and languages. Unlike the previous studies in the literature, our approach does not require training of a classifier for each language, domain or type of event. Then, if an unexpected event happens in a country, our method would detect it because the characteristics of the event are similar to the events in the training dataset. Nevertheless, the results also showed that there were 30% of unrelated events that were incorrectly detected as crisis situations. Hence, a human validation is necessary to confirm each detection similar to a decision support system where a expert confirms the final outcome.

As mentioned above, the approach had a good performance in different languages such as English, Spanish and Italy. Nonetheless, we just developed our experiments in countries with one first native language. For example, countries such as India, Switzerland or South Africa have multiples official languages. In this case, our approach would filter the locations in one specific language, though we would not detect a new crisis, because users share messages in several languages in these countries. However, we would improve this situation creating multiples gazetteer trees for each official language.

We also presented an analysis of the geographic spread for different types of events such as earthquakes and terrorist attacks. As mentioned in the Subsection 4.4, the adjacency matrix was constructed manually, which could affect the analysis in regions with many neighbor locations. Furthermore, we did not consider the distance between locations to compute closeness. For this reason, our future work will include the use of the distance to calculate, for example, spatial autocorrelations with the goal of determining whether two observations are similar, based on their spatial location degrees. With respect to the number of crisis events, this was a proof of concept because we considered just a small portion of emergency situations, which are not representative for all types of crises according to either the hazard type (natural or human-induced), temporal development (instantaneous or progressive) or geographic spread (diffused or focalized). For this reason, our future work will expand our dataset, including a vast number of emergency and non-emergency events. Additionally, this could contribute to developing a journal paper with our results.

In this thesis, the general objective was to detect emergency events, but we also developed a brief characterization of crisis events as illustrated in Figure 8.1. We showed different patterns between two types of instantaneous events: earthquakes and terrorist attacks. These differences were related to the relationship between the delay time to detect the event and the number of locations detected by our method. Based on these promising results, we will extend the characterization of crisis events in a language independent manner. We will therefore explore transfer learning and domain adaptation techniques, and other non-textual features. Furthermore, we could conduct a study of the sentiment analysis to understand what the emotions and feelings of the social media users are during emergency events. In addition, we could compare this level of sentiment with other non-crisis events.

We could also improve our detection results in different ways. We will add points of interest to our gazetteer tree to increase the frequency by time-windows in each hierarchy. Furthermore, we will add more non-textual features such as number of retweets and tweets, unique detected locations and special locations. We also plan to study the relevance of the different metadata-levels, assign weights for each, and create a web application to visualize events in real-time. Finally, if we consider that users have collective patterns in social media, we could compare anomalies between social media activities and real physics sensors. For example, we could address the following question: are there relationships between social media bursty activities, seismometers, and phone calls when a crisis happens?

Bibliography

- [1] IFRCworld disaster report 2015: Focus on local actors, the key to humanitarian effectiveness. http://ifrc-media.org/interactive/wp-content/uploads/2015/09/1293600-World-Disasters-Report-2015_en.pdf. Accessed: 2018-06-15.
- [2] United Nationsdepartment of humanitarian affairs. <https://reliefweb.int/sites/reliefweb.int/files/resources/004DFD3E15B69A67C1256C4C006225C2-dha-glossary-1992.pdf>. Accessed: 2018-06-15.
- [3] United Nationsthe president of the general assembly. https://www.un.org/pga/70/wp-content/uploads/sites/10/2015/08/3-Mar-Humanitarian-Response-in-Africa_The-Urgency-to-Act-30-March-2016.pdf. Accessed: 2018-06-15.
- [4] Fabian Abel, Claudia Hauff, Geert-Jan Houben, Richard Stronkman, and Ke Tao. Twitcident: fighting fire with information from social web streams. In *Proceedings of the 21st International Conference on World Wide Web*, pages 305–308. ACM, 2012.
- [5] Shahriar Akter and Samuel Fosso Wamba. Big data and disaster management: a systematic review and agenda for future research. *Annals of Operations Research*, pages 1–21, 2017.
- [6] Jie Bao, Yu Zheng, David Wilkie, and Mohamed F Mokbel. A survey on recommendations in location-based social networks. *ACM Transaction on Intelligent Systems and Technology*, 2013.
- [7] Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on twitter. *ICWSM*, 11(2011):438–441, 2011.
- [8] Axel Bruns and Jean E Burgess. Local and global responses to disaster:# eqnz and the christchurch earthquake. In *Disaster and emergency management conference, conference proceedings*, volume 2012, pages 86–103. AST Management Pty Ltd, 2012.
- [9] Mark A Cameron, Robert Power, Bella Robinson, and Jie Yin. Emergency situation awareness from twitter for crisis management. In *Proceedings of the 21st International Conference on World Wide Web*, pages 695–698. ACM, 2012.

- [10] Cornelia Caragea, Nathan McNeese, Anuj Jaiswal, Greg Traylor, Hyun-Woo Kim, Prasenjit Mitra, Dinghao Wu, Andrea H Tapia, Lee Giles, Bernard J Jansen, et al. Classifying text messages for the haiti earthquake. In *Proceedings of the 8th international conference on information systems for crisis response and management (ISCRAM2011)*. Citeseer, 2011.
- [11] Lowell Juilliard Carr. Disaster and the sequence-pattern concept of social change. *American Journal of Sociology*, 38(2):207–218, 1932.
- [12] Carlos Castillo. *Big crisis data: social media in disasters and time-critical situations*. Cambridge University Press, 2016.
- [13] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [14] Laurence Echard. *The Gazetteer’s: Or Newsman’s Interpreter: Being a Geographical Index. Of All the Considerable Cities, Patriachships,... in Europe.... The Seventh Edition, Corrected and Very Much Enlarged with the Addition of a Table of Births, Marriages, &c. of All Kings,... of Europe. By Laurence Eachard,...* John Nicholson, and Samuel Ballard, 1704.
- [15] Mark Graham, Scott A Hale, and Devin Gaffney. Where in the world are you? geolocation and language identification in twitter. *The Professional Geographer*, 66(4):568–578, 2014.
- [16] Jheser Guzman and Barbara Poblete. On-line relevant anomaly detection in the twitter stream: an efficient bursty keyword detection model. In *Proceedings of the acm sigkdd workshop on outlier detection and description*, pages 31–39. ACM, 2013.
- [17] George Haddow, Jane Bullock, and Damon P Coppola. *Introduction to emergency management*. Butterworth-Heinemann, 2017.
- [18] Marwan Hassani and Thomas Seidl. Using internal evaluation measures to validate the quality of diverse stream clustering algorithms. *Vietnam Journal of Computer Science*, 4(3):171–183, 2017.
- [19] Brent Hecht, Lichan Hong, Bongwon Suh, and Ed H Chi. Tweets from justin beiber’s heart: the dynamics of the location field in user profiles. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 237–246. ACM, 2011.
- [20] Qunying Huang, Guido Cervone, Duangyang Jing, and Chaoyi Chang. Disastermapper: A cybergis framework for disaster management using social media data. In *Proceedings of the 4th International ACM SIGSPATIAL Workshop on Analytics for Big Geospatial Data*, pages 1–6. ACM, 2015.
- [21] Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. Processing social media messages in mass emergency: A survey. *ACM Computing Surveys (CSUR)*, 47(4):67, 2015.

- [22] Muhammad Imran, Carlos Castillo, Ji Lucas, Patrick Meier, and Sarah Vieweg. Aidr: Artificial intelligence for disaster response. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 159–162. International World Wide Web Conferences Steering Committee, 2014.
- [23] Muhammad Imran, Shady Elbassuoni, Carlos Castillo, Fernando Diaz, and Patrick Meier. Extracting information nuggets from disaster-related messages in social media. In *ISCRAM*, 2013.
- [24] Muhammad Imran, Shady Elbassuoni, Carlos Castillo, Fernando Diaz, and Patrick Meier. Practical extraction of disaster-relevant information from social media. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1021–1024. ACM, 2013.
- [25] Twitter Inc. The Twitter Glosary. <http://help.twitter.com/en/glossary>. Accessed: 2018-02-12.
- [26] Twitter Inc. Twitter data dictionaries. <http://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object>. Accessed: 2018-02-12.
- [27] Janani Kalyanam, Mauricio Quezada, Barbara Poblete, and Gert Lanckriet. Prediction and characterization of high-activity events in social media triggered by real-world news. *PloS one*, 11(12):e0166694, 2016.
- [28] Sarvnaz Karimi, Jie Yin, and Cecile Paris. Classifying microblogs for disasters. In *Proceedings of the 18th Australasian Document Computing Symposium*, pages 26–33. ACM, 2013.
- [29] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.
- [30] Raffi Krikorian. Map of Twitter Status Object. <http://online.wsj.com/public/resources/documents/TweetMetadata.pdf>. Accessed: 2018-02-12.
- [31] Yury Kryvasheyev, Haohui Chen, Nick Obradovich, Esteban Moro, Pascal Van Hentenryck, James Fowler, and Manuel Cebrian. Rapid assessment of disaster damage using social media activity. *Science advances*, 2(3):e1500779, 2016.
- [32] Shamanth Kumar, Geoffrey Barbier, Mohammad Ali Abbasi, and Huan Liu. Tweet-tracker: An analysis tool for humanitarian and disaster relief. In *ICWSM*, 2011.
- [33] Rui Li, Kin Hou Lei, Ravi Khadiwala, and Kevin Chen-Chuan Chang. Tedas: A twitter-based event detection and analysis system. In *Data engineering (icde), 2012 ieee 28th international conference on*, pages 1273–1276. IEEE, 2012.
- [34] Jessica Lin, Michail Vlachos, Eamonn Keogh, and Dimitrios Gunopulos. Iterative incremental clustering of time series. In *International Conference on Extending Database Technology*, pages 106–122. Springer, 2004.

- [35] John Lingad, Sarvnaz Karimi, and Jie Yin. Location extraction from disaster-related microblogs. In *Proceedings of the 22nd international conference on world wide web*, pages 1017–1020. ACM, 2013.
- [36] Nicola Lunardon, Giovanna Menardi, and Nicola Torelli. Rose: A package for binary imbalanced learning. *R Journal*, 6(1), 2014.
- [37] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.
- [38] Jazmine Maldonado, Jheser Guzman, and Barbara Poblete. A lightweight and real-time worldwide earthquake detection and monitoring system based on citizen sensors. In *Proceedings of the Fifth Conference of Human Computation and Crowdsourcing*, pages 137–146. AAAI, 2017.
- [39] Kanti V Mardia. Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57(3):519–530, 1970.
- [40] Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. Twitter under crisis: Can we trust what we rt? In *Proceedings of the first workshop on social media analytics*, pages 71–79. ACM, 2010.
- [41] Theophano Mitsa. *Temporal data mining*. CRC Press, 2010.
- [42] Fred Morstatter, Jürgen Pfeffer, Huan Liu, and Kathleen M Carley. Is the sample good enough? comparing data from twitter’s streaming api with twitter’s firehose. In *ICWSM*, 2013.
- [43] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kiciman. Social data: Biases, methodological pitfalls, and ethical boundaries. 2016.
- [44] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In *ICWSM*, 2014.
- [45] Alexandra Olteanu, Sarah Vieweg, and Carlos Castillo. What to expect when the unexpected happens: Social media communications across crises. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 994–1009. ACM, 2015.
- [46] Leysia Palen and Sophia B Liu. Citizen communications in crisis: anticipating a future of ict-supported public participation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 727–736. ACM, 2007.
- [47] Liza Potts, Joyce Seitzinger, Dave Jones, and Angela Harrison. Tweeting disaster: hashtag constructions and collisions. In *Proceedings of the 29th ACM international conference on Design of communication*, pages 235–240. ACM, 2011.

- [48] Christian Reuter and Marc-André Kaufhold. Fifteen years of social media in emergencies: a retrospective review and future directions for crisis informatics. *Journal of Contingencies and Crisis Management*.
- [49] Hernan Sarmiento. Detecting emergency situations by inferring locations in twitter. In *Seventh BCS-IRSG Symposium on Future Directions in Information Access, FDIA 2017, 5 September 2017, Barcelona, Spain, 2017*.
- [50] Kristin Stock. Mining location from social media: A systematic review. *Computers, Environment and Urban Systems*, 2018.
- [51] Kevin Stowe, Michael J Paul, Martha Palmer, Leysia Palen, and Kenneth Anderson. Identifying and categorizing disaster-related tweets. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*, pages 1–6, 2016.
- [52] Pang-Ning Tan et al. *Introduction to data mining*. Pearson Education India, 2006.
- [53] Sayan Unankard, Xue Li, and Mohamed A Sharaf. Emerging event detection in social networks with location sensitivity. *World Wide Web*, 18(5):1393–1417, 2015.
- [54] Sarah Vieweg, Amanda L Hughes, Kate Starbird, and Leysia Palen. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1079–1088. ACM, 2010.
- [55] Sarah Elizabeth Vieweg. *Situational awareness in mass emergency: A behavioral and linguistic analysis of microblogged communications*. PhD thesis, University of Colorado at Boulder, 2012.
- [56] Maximilian Walther and Michael Kaiser. Geo-spatial event detection in the twitter stream. In *ECIR*, pages 356–367. Springer, 2013.
- [57] Kazufumi Watanabe, Masanao Ochi, Makoto Okabe, and Rikio Onai. Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2541–2544. ACM, 2011.
- [58] Junjie Wu, Jian Chen, Hui Xiong, and Ming Xie. External validation measures for k-means clustering: A data distribution perspective. *Expert Systems with Applications*, 36(3):6050–6061, 2009.
- [59] Jie Yin, Sarvnaz Karimi, and John Lingad. Pinpointing locational focus in microblogs. In *Proceedings of the 2014 Australasian Document Computing Symposium*, page 66. ACM, 2014.
- [60] Jie Yin, Sarvnaz Karimi, Bella Robinson, and Mark Cameron. Esa: emergency situation awareness via microbloggers. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2701–2703. ACM, 2012.

Appendices

Appendix A

Tweet Object

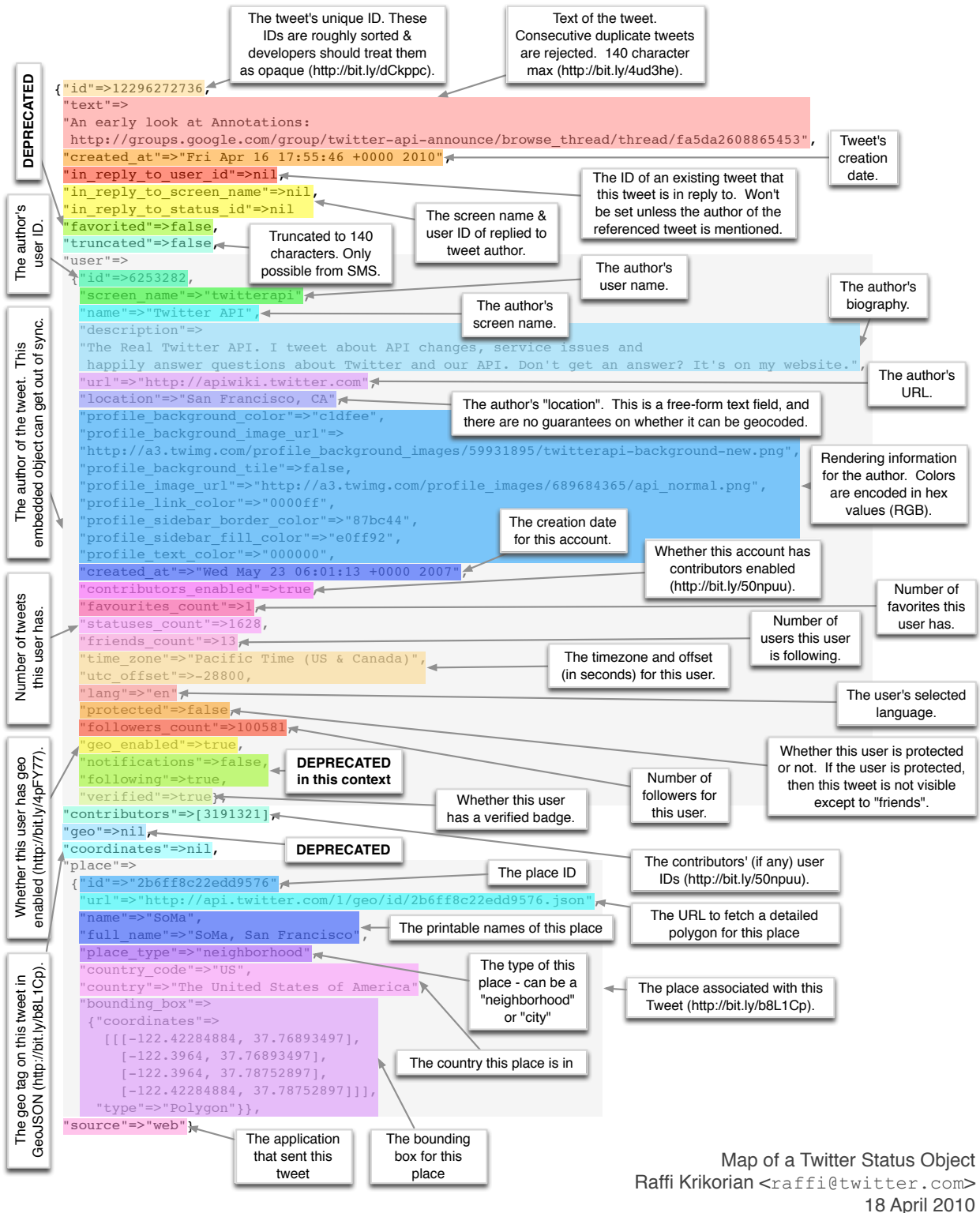


Figure A.1: Tweet metadata retrieved from the Twitter Public Streaming API, as of April 2010[30]. Colors represent the different types of information. Some fields can be deprecated and the current fields can be found on Twitter’s developed page [26]

Appendix B

Text pre-processing

Table B.1: Examples of text preprocessing.

Raw text	After preprocessing
RT @onemichile: Sismo de mayor intensidad entre las regiones de Coquimbo y Biobío. Más información en https://t.co/OFmIDav6yZ	['sismo', 'de', 'mayor', 'intensidad', 'entre', 'las', 'regiones', 'de', 'coquimbo', 'y', 'biobio', 'mas', 'informacion', 'en']
Urgente #Onemi informa EVACUACION DE BORDE COSTERO #VALPARAISO #Chile #Terremoto	['urgente', 'onemi', 'informa', 'evacuacion', 'de', 'borde', 'costero', 'valparaiso', 'chile', 'terremoto']
Leeds / London	['leeds', 'london']
RT @Matt_in_London: Sounds like something major's going down at the Ariana show in Manchester. Loud bangs and people being evacuated.	['sounds', 'like', 'something', 'major', 's', 'going', 'down', 'at', 'the', 'ariana', 'show', 'in', 'manchester', 'loud', 'bangs', 'and', 'people', 'being', 'evacuated']
@giornaleprociv: ++ #TERREMOTO FORTE SCOSSA CENTRO ITALIA / ROMA ore 7.41 prime valutazioni dicono M 6.6 ma.dato non ancora certo	['terremoto', 'forte', 'scossa', 'centro', 'italia', 'roma', 'ore', '7', '41', 'prime', 'valutazioni', 'dicono', 'm', '6', '6', 'ma', 'dato', 'non', 'ancora', 'certo']

Appendix C

Features Extraction

Table C.1: Features extraction by time-windows for each type of signal.

	Feature	Description
	Frequency (freq)	the number of messages
	Min	the minimum time difference between two consecutive messages (found in the all messages) in the current time-window
	Max	the maximum time difference between two consecutive messages (found in the all messages) in the current time-window
	The first quartile (Q1)	the middle number between the smallest number and the median computed for the time difference between two consecutive messages in the current time-window
	The second quartile (Q2)	the computed median for the time difference between two consecutive messages in the current time-window
	The third quartile (Q3)	the middle value between the median and the highest value computed for the time difference between two consecutive messages in the current time-window

Continued on next page

Table C.1 – *Continued from previous page*

	Feature	Description
	Average (avg)	the average difference between two consecutive messages (found in the all messages) in the current time-window
	Skewness (skew)	the asymmetry of the normal distribution shape computed between two consecutive messages (found in the all messages) in the current time-window
	Kurtosis (kurt)	the tailedness of the normal distribution shape computed between two consecutive messages (found in the all messages) in the current time-window

Table C.2: Features extraction by time-windows for each type of signal considering normalized features.

	Feature	Description
Number of messages	Frequency (freq)	the number of messages
	Frequency zscore (freq_zscore)	the normalized number of messages with respect to all previous values
	Frequency rolling-5 zscore (freq_rollz5)	the normalized number of messages with respect to the last five previous time-windows
	Frequency rolling-10 zscore (freq_rollz10)	the normalized number of messages with respect to the last ten previous time-windows
	Frequency rolling-15 zscore (freq_rollz15)	the normalized number of messages with respect to the last fifteen previous time-windows
	Frequency rolling-20 zscore (freq_rollz20)	the normalized number of messages with respect to the last twenty previous time-windows

Continued on next page

Table C.2 – *Continued from previous page*

	Feature	Description
Interarrival Time	Min	the minimum difference between two consecutive messages (found in the all messages) in the current time-window
	Max	the maximum difference between two consecutive messages (found in the all messages) in the current time-window
	The first quartile (Q1)	the middle number between the smallest number and the median computed for the difference between two consecutive messages in the current time-window
	The second quartile (Q2)	the computed median for the difference between two consecutive messages in the current time-window
	The third quartile (Q3)	the middle value between the median and the highest value computed for the difference between two consecutive messages in the current time-window
	Average (avg)	the average difference between two consecutive messages (found in the all messages) in the current time-window
	Skewness (skew)	the asymmetry of the normal distribution shape computed between two consecutive messages (found in the all messages) in the current time-window
	Skewness zscore (skew_zscore)	the normalized asymmetry of the normal distribution shape respect to previous values for the difference between two consecutive messages (found in the all messages) in the current time-window
	Kurtosis (kurt)	the tailedness of the normal distribution shape computed between two consecutive messages (found in the all messages) in the current time-window

Continued on next page

Table C.2 – *Continued from previous page*

	Feature	Description
	Kurtosis zscore (kurt_zscore)	the normalized tailedness of the normal distribution shape respect to previous values for the difference between two consecutive messages (found in the all messages) in the current time-window

Appendix D

Printing Details to Remove Redundant Features

```
Compare row 3 and column 5 with corr 0.79
Means: 0.58 vs 0.279 so flagging column 3

Compare row 4 and column 5 with corr 0.98
Means: 0.504 vs 0.293 so flagging column 4

Compare row 5 and column 2 with corr 0.779
Means: 0.373 vs 0.211 so flagging column 5

All correlations <= 0.75
```

Figure D.1: Output for `findCorrelation` over frequency features.

```
Compare row 4 and column 3 with corr 0.947
Means: 0.527 vs 0.324 so flagging column 4

Compare row 3 and column 5 with corr 0.827
Means: 0.454 vs 0.278 so flagging column 3

Compare row 5 and column 2 with corr 0.845
Means: 0.396 vs 0.256 so flagging column 5

All correlations <= 0.75
```

Figure D.2: Printing details for `findCorrelation` over inter-arrival features.

Appendix E

Results of the Supervised Approach

Table E.1: List of the parameters using for searching the best performance in the country hierarchy for the SVM classifier.

kernel	C	gamma	coef0	degree	class.weights
linear	20				1:50
linear	10				1:50
linear	1				1:50
linear	50				1:50
polynomial	5	0.1	0	3	1:50
polynomial	20	2	2	2	1:50
polynomial	20	2	2	3	1:50
polynomial	20	2	2	4	1:50
polynomial	20	2	2	5	1:50
polynomial	20	3	3	3	1:50
polynomial	20	3	3	4	1:50
polynomial	20	3	3	5	1:50
polynomial	50	10	2	3	1:50
polynomial	50	2	2	5	1:50
radial	20	2			1:50
radial	20	0.1			1:50
radial	20	5			1:50
radial	50	2			1:50
radial	50	0.1			1:50
radial	50	5			1:50
sigmoid	20	2	2		1:50
sigmoid	20	10	2		1:50
sigmoid	20	30	2		1:50
sigmoid	20	0.5	2		1:50

Table E.2: List of the parameters using for searching the best performance in the state hierarchy for the SVM classifier.

kernel	C	gamma	coef0	degree	class.weights
linear	20				1:5
linear	10				1:5
linear	1				1:5
linear	50				1:5
polynomial	5	0.1	0	3	1:5
polynomial	20	2	2	2	1:5
polynomial	20	2	2	3	1:5
polynomial	20	2	2	4	1:5
polynomial	20	2	2	5	1:5
polynomial	20	3	3	3	1:5
polynomial	20	3	3	4	1:5
polynomial	20	3	3	5	1:5
polynomial	50	10	2	3	1:5
polynomial	50	2	2	5	1:5
radial	20	2			1:5
radial	20	0.1			1:5
radial	20	5			1:5
radial	50	2			1:5
radial	50	0.1			1:5
radial	50	5			1:5
sigmoid	20	2	2		1:5
sigmoid	20	10	2		1:5
sigmoid	20	30	2		1:5
sigmoid	20	0.5	2		1:5

Table E.3: Division of earthquakes used in 5-fold cross validation.

fold	Training set	Testing set
1	Italy EQ 6.6 <i>Mw</i> Chile EQ 7.6 <i>Mw</i> Chile EQ 5.9 <i>Mw</i> Chile EQ 6.9 <i>Mw</i>	Italy EQ 5.5 <i>Mw</i>
2	Italy EQ 6.6 <i>Mw</i> Italy EQ 5.5 <i>Mw</i> Chile EQ 5.9 <i>Mw</i> Chile EQ 6.9 <i>Mw</i>	Chile EQ 7.6 <i>Mw</i>
3	Italy EQ 6.6 <i>Mw</i> Italy EQ 5.5 <i>Mw</i> Chile EQ 6.9 <i>Mw</i> Chile EQ 7.6 <i>Mw</i>	Chile EQ 5.9 <i>Mw</i>
4	Italy EQ 6.6 <i>Mw</i> Italy EQ 5.5 <i>Mw</i> Chile EQ 5.9 <i>Mw</i> Chile EQ 7.6 <i>Mw</i>	Chile EQ 6.9 <i>Mw</i>
5	Italy EQ 5.5 <i>Mw</i> Chile EQ 7.6 <i>Mw</i> Chile EQ 5.9 <i>Mw</i> Chile EQ 6.9 <i>Mw</i>	Italy EQ 6.6 <i>Mw</i>

Table E.4: Results of the Support Vector Machine classifier using 5-fold cross validation.

fold	kernel	C	gamma	coef0	degree	weights	P	R	F1
<i>Country</i>									
1	polynomial	20	2	2	3	1:50	0.275	0.666	0.390
2	polynomial	20	2	2	3	1:50	0.321	1	0.486
3	polynomial	20	2	2	3	1:50	0.357	0.833	0.500
4	polynomial	20	2	2	3	1:50	0.400	1	0.571
5	polynomial	20	2	2	3	1:50	0.189	0.777	0.304
Average							0.308	0.855	0.450
Stdev							0.081	0.145	0.104
<i>State</i>									
1	linear	1	-	-	-	1:5	0.285	1	0.444
2	linear	1	-	-	-	1:5	0.236	1	0.383
3	linear	1	-	-	-	1:5	0.629	0.850	0.720
4	linear	1	-	-	-	1:5	0.500	0.700	0.583
5	linear	1	-	-	-	1:5	0.142	0.750	0.240
Average							0.358	0.86	0.474
Stdev							0.200	0.138	0.184

Appendix F

Results of the Unsupervised Approach

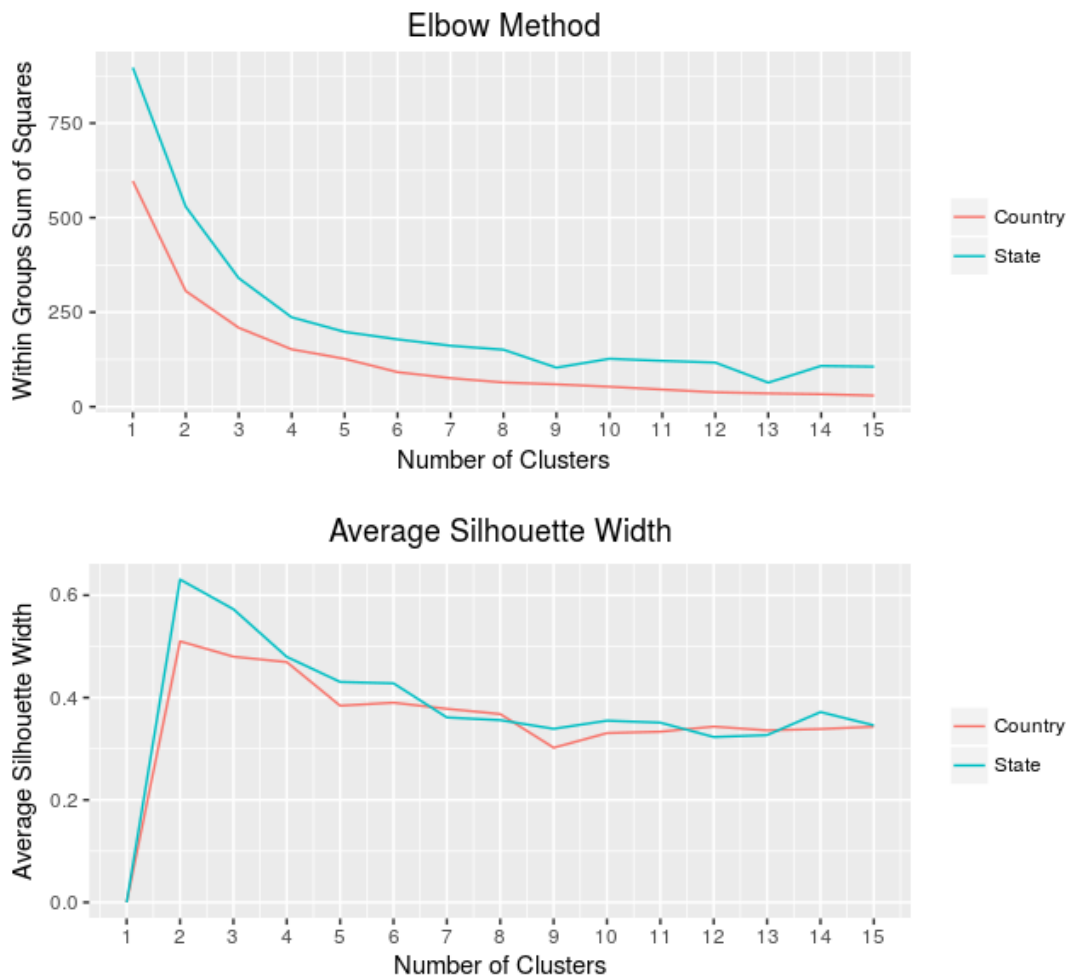


Figure F.1: The total within-cluster sum of squares and the average silhouette width for country and state hierarchy. The k number of clusters is tested between $2 \leq k \leq 15$.

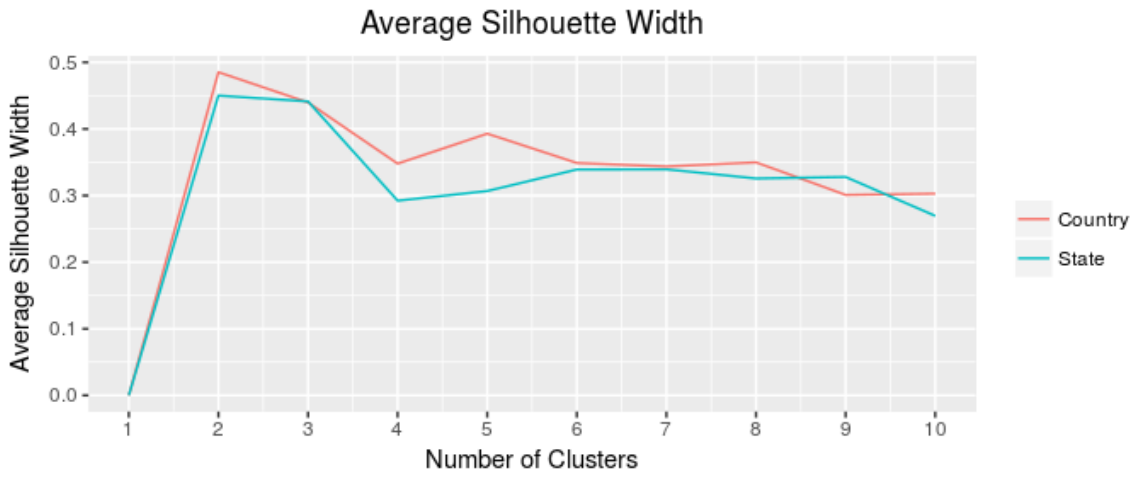


Figure F.2: The average silhouette width for country and state hierarchy using the Euclidean distance. The k number of clusters was tested between $2 \leq k \leq 15$.

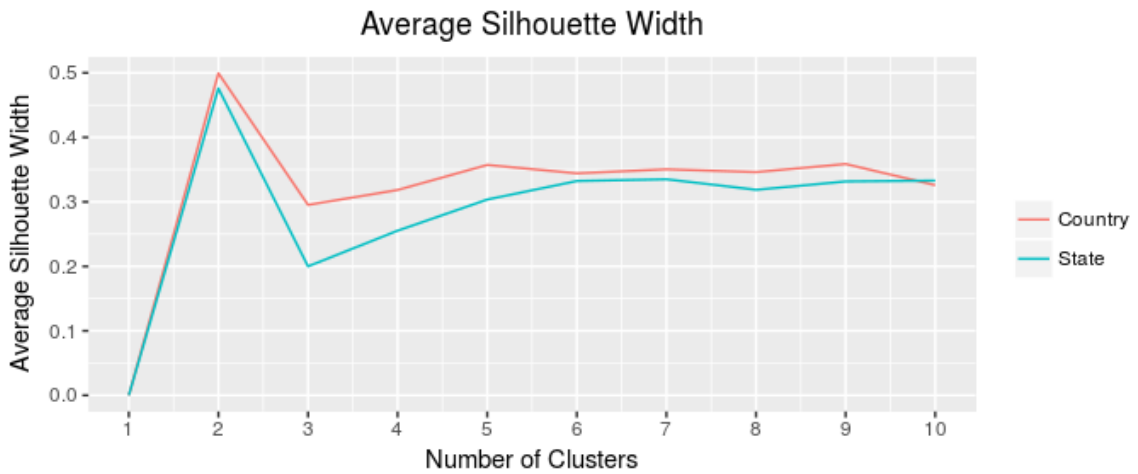


Figure F.3: The average silhouette width for country and state hierarchy using the Manhattan distance. The k number of clusters was tested between $2 \leq k \leq 15$.

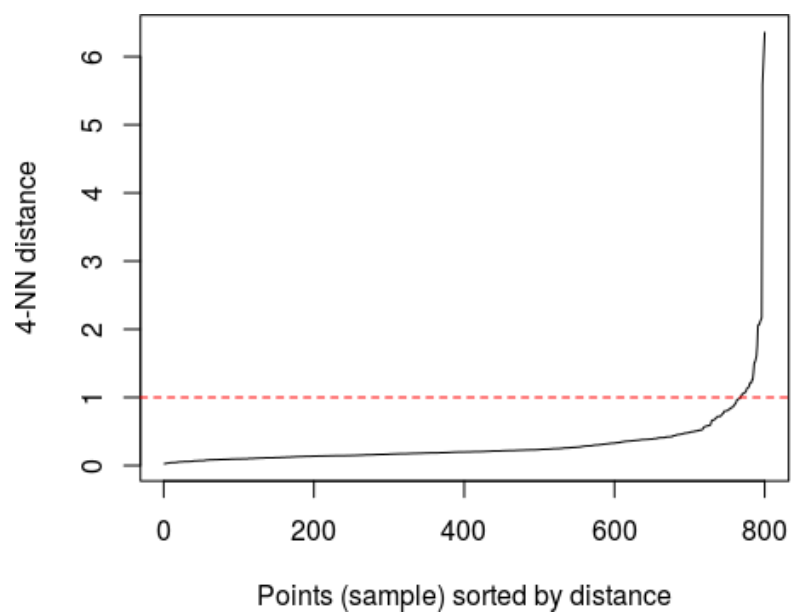


Figure F.4: The k-nearest neighbor distances computed for country hierarchy. The plot can be used to help find a suitable value for the eps neighborhood for DBScan.

k	iter_max	n_start	cluster1	cluster2	avg silhouette	purity	entropy	N.M.I	N.V.I
<i>Country Hierarchy</i>									
2	1000	20	137	63	0.510	0.825	0.667	0.273	0.847
2	1000	10	137	63	0.510	0.825	0.667	0.273	0.847
2	1000	5	137	63	0.510	0.825	0.667	0.273	0.847
2	1000	1	135	65	0.510	0.835	0.647	0.309	0.817
2	100	20	137	63	0.510	0.825	0.667	0.273	0.847
2	100	20	137	63	0.510	0.825	0.667	0.273	0.847
2	100	10	137	63	0.510	0.825	0.667	0.273	0.847
2	100	5	137	63	0.510	0.825	0.667	0.273	0.847
2	100	1	137	63	0.510	0.825	0.667	0.273	0.847
<i>State Hierarchy</i>									
2	1000	20	44	256	0.630	0.903	0.376	0.358	0.781
2	1000	10	44	256	0.630	0.903	0.376	0.358	0.781
2	1000	5	44	256	0.630	0.903	0.376	0.358	0.781
2	1000	1	44	256	0.630	0.903	0.376	0.358	0.781
2	100	20	44	256	0.630	0.903	0.376	0.358	0.781
2	100	10	44	256	0.630	0.903	0.376	0.358	0.781
2	100	5	44	256	0.630	0.903	0.376	0.358	0.781
2	100	1	44	256	0.630	0.903	0.376	0.358	0.781

Table F.1: Results of the external validation for k-means algorithm. The *iter_max* and *n_start* represent the number of iteration for finishing and the number of points for creating the cluster respectively.

k	distance	cluster1	cluster2	avg silhouette	purity	entropy	N.M.I	N.V.I
<i>Country Hierarchy</i>								
2	euclidean	122	78	0.485	0.810	0.682	0.321	0.808
2	manhattan	120	80	0.499	0.81	0.672	0.337	0.796
2	minkowski 3	117	83	0.461	0.815	0.635	0.387	0.759
2	minkowski 4	117	83	0.455	0.815	0.635	0.387	0.759
<i>State Hierarchy</i>								
2	euclidean	111	189	0.45	0.83	0.712	0.296	0.826
2	manhattan	186	114	0.476	0.83	0.706	0.311	0.815
2	minkowski 3	112	188	0.434	0.83	0.717	0.292	0.828
2	minkowski 4	186	114	0.424	0.83	0.728	0.284	0.834

Table F.2: Results of the external validation for k-medoid algorithm. We also used the dissimilarity matrix between elements.

eps	minPts	distance	n_clusters	outliers	purity	entropy	N.M.I	N.V.I
<i>Country Hierarchy</i>								
0.9	10	euclidean	1	12	0.800	0.254	0.130	0.930
0.8		euclidean	1	14	0.800	0.292	0.126	0.932
0.9	5	euclidean	1	12	0.800	0.254	0.130	0.930
0.8	5	euclidean	1	13	0.805	0.263	0.146	0.920
1	10	euclidean	1	11	0.805	0.220	0.157	0.914
1	5	euclidean	1	9	0.795	0.199	0.123	0.934
<i>State Hierarchy</i>								
1.1	10	euclidean	1	13	0.860	0.183	0.160	0.912
1.1	5	euclidean	1	8	0.850	0.129	0.114	0.939
0.9	10	euclidean	1	23	0.873	0.277	0.216	0.878
0.9	5	euclidean	1	13	0.886	0.165	0.201	0.887
1	10	euclidean	1	15	0.860	0.211	0.157	0.914
1	5	euclidean	1	8	0.850	0.129	0.114	0.939

Table F.3: Results of the external validation for DBScan algorithm.