UNIVERSITY OF AGDER

Crisis Analysis Based on Tweets

By:

Enok Karlsen Eskeland

Supervisor:

Ole-Christoffer Granmo

This project is carried out as a part of the education at the University of Agder and is therefore approved as a part of this education.

University of Agder, 2013

Faculty of Engineering and Science

Department of Information- and Communication Technology

Abstract

Information from the public during a crisis is often limited to people calling emergency services. Social media provides new opportunities to get input from the public during times of crisis. To avoid reading through massive amounts of social media data a system automatically detecting events is advantageous. Twitter's simple format makes it easy to create tweets, although analyzing them is more of a challenge. The tweet stream's high noise ratio combined with the shear amount of tweets makes detecting events a formidable challenge. To be able to detect any crisis events, the solution of this thesis is constructed as a general event detector, but with emphasis on spatial detection. This makes it possible not only to detect events, but in many cases also to estimate the location of these events. This approach combines features from crisis centric event detectors with general event detectors. The solution is constructed as a three part pipeline. The first part retrieves tweets. The second part detects events and is called the detection pipeline. The last part is a website called Grapher. It visualizes the detected events. The detection pipeline is the core of the solution. It consists of a temporal, word density and two spatial detection methods. In addition the detection pipeline clusters the suggested words from the methods. The detection methods are based on comparing two statistical models based on historic data and new data. The two spatial methods and the temporal method detects words and locations by comparing kernel density estimates with a state-of-the-art method. The solution pipeline has been extensively tested on real data. It is able to detect both crisis events and events of a more general character. For general events it has an event noise ratio of 65%. For crisis events it has an event noise ratio of 94%. The results show the proposed detection methods are viable and thus impacting the field of social media event detection. The solution could be applied by crisis handling teams and organizations monitoring social media in a specific area.

Contents

1 Introduction **10**

1.1 Background 10

1.2 Problem Statement 13

1.3 Literature Review 16

1.3.1 Related Work - Overview 16

1.3.2 Emerging Topic Detection on Twitter based on Temporal and Social Terms Evaluation [7] 19

1.3.3 TEDAS: a Twitter-based Event Detection and Analysis System [8] 21

1.4 Solution Approach 22

1.5 Contributions 24

1.6 Report Outline 25

2 Theoretical Background **26**

2.1 Kernel Density Estimation 26

2.2 Comparing Two Kernel Density Estimates 29

2.3 Odds Ratio 30

3 Solution **31**

3.1 Tweet Retriever 31

3.2 Event Detector 32

3.2.1 Tweet Parser 33

3.2.2 The Three Detection Methods 35

3.2.3 Clustering Words with Odds Ratio 43

3.2.4 The Bivariate Spatial Method - Bivariate Spatial Comparison of Kernel Density Estimates for Each Graph 45

3.3 Grapher 47

3.3.1 Overview Page 47

3.3.2 Graph Page 48

3.3.3 Map Page 49

4 Discussion 51
4.1 Geotagged Tweets Compared to Non Geotagged Tweets 51
4.2 Temporal Analysis of Geotagged Tweets 53
4.3 Privacy Issues 55
4.4 Results 56
4.4.1 General Event Detection 57
4.4.2 Crisis Detection 84
4.5 General Aspects of Solution 95
4.5.1 Detection Pipeline 95
4.5.2 Grapher 97
4.6 Future Improvements 97
4.6.1 Event Detector Improvements 98
4.6.2 Execution Improvements 99
4.6.3 Visualization Improvements 100
4.7 Solution Applications 100

5 Conclusion 102

Acknowledgments 105

References 106

List of Figures

1.1 Both circles illustrate events originating in the center. The left circle illustrates an event which affects its surrounding area as a bivariate Gaussian distribution. The right circle illustrates an event affecting its surrounding area as a bivariate Uniform distribution. . 15

1.2 Solution pipeline. 22

2.1 Kernel density estimation and the corresponding histogram with optimized bandwidth. Y is the density function. 27

2.2 Kernel density estimates with different bandwidths. The black graph has an optimal bandwidth. The red graph has a too high bandwidth. The blue graph has a too low bandwidth. 28

3.1 Solution pipeline. 31

3.2 Detection pipeline. 33

3.3 $threshold(p_{new}(w))$ 38

3.4 The first is the historic density ordered in descending order. Next is new ordered in the same order as the previous. First graph on the second row starts with the significantly different words. The last graph puts it all together. 39

3.5 The graph on the left side is new (blue) KDE and historic (black) KDE. On the right side is the comparison of the two graphs. f_1 is historic, while f_2 is new. 40

3.6 The leftmost graph contains two plots. Blue is new KDE and black is historic KDE. On the right side is a comparison of the two KDEs. f_1 is historic and f_2 is new. It can be observed when new is has higher value on the left graph it has a negative value in the compared graph. 42

3.7 Bivariate KDE of multiple words in the state of Colorado. 43

3.8 The upper left graph f_1 is a kde based on historical data. The upper right graph f_2 is a kde based on new data. The lower graph is the difference between the two graphs. In the green area f_2 is significantly larger. 45
3.9 Illustrating the three different pages in Grapher. The leftmost screenshot is the Overview page. The middle screenshot illustrates the events in graphs. The right screenshot shows the map page. 47
3.10 Graph illustrating varying node sizes and edge widths. 48
3.11 Map of dots representing tweet positions located in western USA. 50

4.1 The left graph is the word density of the non geotagged tweets. The right graph is the word density of geotagged tweets. They are sorted in the same order. It is observed there are significant differences between the two data sets. 52
4.2 Increasing number of tweets in March. 54
4.3 Privacy issue? 56
4.4 Ratio between the three detection methods, spatial method, temporal method, and WDC. 59
4.5 Part of the Oscars graph describing the movie Life of Pi and its winnings. See Table 4.4 for details about this event. 62
4.6 Part of the Oscars graph describing the James Bond 50th anniversary and Adele who wrote the song Skyfall. 65
4.7 Ratio between the three detection methods, spatial method, temporal method, and WDC. 66
4.8 The Bivariate spatial method show there is an increase in tweets containing the words ”tornado” and ”phone” in Wisconsin (the red area). The map in the center contains the positions of all tweets using the word ”tornado”. The map on the right contains all tweets having the words ”tornado” or ”phone”. 69
4.9 The left map displays all tweets containing the word ”snow” or the word ”storm”. The right map displays all tweets containing the word ”delay”. 72
4.10 The black graphs are based on historic data. The blue graphs are based on new data. It is possible to guess where the arenas are based on the two graphs. 72
4.11 Venn-diagram of the ratio between the three detection methods. . . 74
4.12 The left graph illustrates how people tweet less and less with the word ”tweet” when the time goes toward night. The right graph illustrates how people are using the word ”30” every half hour. Time 60 equals 120 minutes. 74

4.13 From morning towards solar noon people are tweeting more and more using the word ”work”. Each time step is equal to two minutes. 75
4.14 Facebook performed a major release at 18:00 GMT. The black graph is based on historic data. The blue graph is based on new data. Each time step is 2 minutes. 77
4.15 Person seen with gun on campus in Indianapolis. The first kde is based on historic data. The next kde is based on new data. The difference can be seen on the map to the right. 78
4.16 Result of WDC method from the Boston bombings. The red bars show the words which have a significant higher share than they did before. 86
4.17 Venn diagram showing the ratio between the detection methods from the Boston bombings. 87
4.18 The word ”boston” is used in a wider geographical area compared to historic tweets. 87
4.19 Overview of graphs from the Boston terrorist attack. 88
4.20 The first leftmost KDE is based on historical data. The middle KDE is based on new data. The rightmost map is the difference between them. 92
4.21 Results of the bivariate spatial method. 93
4.22 Left KDE is based on 500 positions. Right KDE is based on 9000 positions. 96

A1 Connection settings. 109
A2 Connection settings. 111
A3 (1) Click on the symbol where the arrow is pointing. (2) Connected.112

List of Tables

4.1 Average Number of tweets per weekday and their standard deviations. 55
4.2 Real events detected January 21 17:00 - 21:00 GMT 58
4.3 Natural fluctuations January 21 17:00 - 21:00 GMT 60
4.4 Oscar events part 1 / 3. February 25 01:00 - 03:30 GMT - Eastern USA. 61
4.5 Oscar events part 2 / 3. February 25 01:00 - 03:30 GMT - Eastern USA. 63
4.6 Oscar events part 3 / 3. February 25 01:00 - 03:30 GMT - Eastern USA. 64
4.7 General events. February 25 01:00 - 03:30 GMT - Eastern USA. . 67
4.8 Natural fluctuations February 25 01:00 - 03:30 GMT - Eastern USA. 68
4.9 Real events detected March 4 15:30 - 17:00 GMT - Eastern USA. 68
4.10 Natural fluctuations March 4 15:30 - 17:00 GMT - Eastern USA. . 69
4.11 Real events detected March 7 2013 00:00 - 02:00 - Eastern USA. . 71
4.12 Natural fluctuations March 7 06:00 - 08:00 GMT - Eastern USA. 73
4.13 Real events detected March 7 2013 06:00 - 08:00 - Eastern USA. . 74
4.14 Natural fluctuations March 7 2013 06:00 - 08:00 GMT - Eastern USA. 75
4.15 Real events detected March 7 2013 12:00 - 14:00 - Eastern USA. . 75
4.16 Natural fluctuations March 7 2013 12:00 - 14:00 - Eastern USA. . 76
4.17 Real events detected March 7 2013 18:00 - 20:00 - Eastern USA. . 76
4.18 Natural fluctuations March 7 2013 18:00 - 20:00 - Eastern USA. . 77
4.19 Real events detected March 19 18:00 - 20:00 - Eastern USA. . . . 78
4.20 Natural fluctuations March 19 18:00 - 20:00 - Eastern USA. . . . 79
4.21 Natural fluctuations March 21 06:00 - 08:00 GMT - Eastern USA. 79
4.22 Real events detected March 26 12:00 - 14:00 GMT - Eastern USA. 80
4.23 Natural fluctuations March 26 12:00 - 14:00 GMT - Eastern USA. 80
4.24 Real events detected March 29 18:00 - 20:00 GMT - Eastern USA. 81
4.25 Natural fluctuations March 29 18:00 - 20:00 GMT - Eastern USA. 81

4.26 Natural fluctuations March 30 06:00 - 08:00 GMT eastern USA. . 82
4.27 Useful information related to the Boston bombings 19:24 - 20:24 GMT - Eastern USA. 89
4.28 Emotions related to the Boston bombings 19:24 - 20:24 GMT - Eastern USA. 90
4.29 Noise 19:24 - 20:24 GMT - Eastern USA. 91

5.1 Summary of the methods in the detection pipeline. See Section 4.4.1 and Section 4.4.2 for more details. 103
5.2 Summary of the two studies on geotagged tweets. See Section 4.1 and Section 4.2 for more details. 103

List of Algorithms

1 Splitting tweet text into words. 34
2 The three detection methods. 36
3 Calculate odds ratio. 44
4 Calculate difference between two bivariate kdes. 46

Definitions

R	A programming language for statistical computing.
KDE	Kernel Density Estimation.
ks	Kernel density estimation library in R.
OR	Odds Ratio.
tweet	Twitter message.
geotag	Information about geographical location.
geotagged tweet	tweet containing GPS position from where it was published from.
WDC	Detection method comparing word densities.
Grapher	A website, part of the solution of this project to visualize the test results from this project.

Chapter 1

Introduction

1.1 Background

Social media has changed the world. In less than a decade we have gone from being passive media consumers to become active producers of social media content. Everyone in the western world possessing the right technology can use social media to get their voice heard.

On top of the social media mountain is Facebook and Twitter. Facebook is the largest with more than a 1 billion unique users every month [1]. These produce more than a billion posts every day. Twitter comes second and has more than 288 million [2] active users every month while the total number of users is almost twice as much [3]. Twitter users create more than 500 million tweets [3] per day.

As a microblogging service Twitter lets people share 140 character long texts called tweets. According to a survey from early 2012 [4] 15 % of the American population which is online use Twitter and 8 % use it on a daily basis. Twitter is used by a diverse set of the population where income and education plays a minor role. With smart phones dominating the handset market, Twitter users tweet more and more from their smart phones. As of early 2012 9 % of all American smart phone owners used Twitter on their phone and 5 % used Twitter on a daily basis. [4]

Gaining insights from one of the largest social media platforms like Twitter can be valuable to many organizations. A scenario could be a mobile phone manufacturer wanting to know the impact their new phone have on the social media community. This is important because the social media community consists of customers and potential customers. If people talk about their new phone in a positive manner it will probably help sell more phones. Regardless of phone sales can an analyzes of tweets give useful feedback to the manufacturer. Another scenario could be to monitor the reactions from marketing campaigns. Already, specialized companies sell social media monitoring services to mobile phone manufacturers and other consumer-oriented companies.

Scenarios less explored are monitoring social media during a developing crisis. Detecting events and gaining insights could help government appointed crisis handling teams to get a better overview of the situation and thus improve delegation of aid. During the Haiti earthquake social media was actively used to gain an understanding of the extent of the crisis [5]. Regardless of the scenario, detecting events by monitoring social media data is becoming increasingly important as more and more people find their way to one or more of the social media platforms.

Detecting events by monitoring social media have many difficult aspects. First of all the amount of social media content produced is enormous and coming up with an efficient solution is in many cases a non trivial exercise. Another aspect involves the content of social media and especially Twitter. A study from 2009 revealed that 40 % of all tweets are just babble [6] like, ”I am eating a sandwich”. Because Twitter users are limited to only 140 characters they often resort to unconventional abbreviations of words. In many cases these abbreviations can be difficult to understand. This is in contrast to for example news articles which strives for correct grammar and syntax.

A hypothetical approach to this challenge of analyzing vast amounts of tweets could be to not use any specialized software. This basic approach to detect events on Twitter could be for a group of analysts to read through tweets and collectively reach consensus about emerging events. There are many difficulties with such an approach. Because the amount of tweets is so great the number of analysts would also have to be great. Often time is limited on such assignments which might again require more analysts. A larger group of analysts might also have problems reaching consensus than a smaller group. A dedicated workforce to coordinate the effort might also be required. It is therefore likely that scaling the analytical team would not be linear. A number of other issues would probably reveal themselves. The difficulty of the task and increased popularity of social media makes the area of event detection on social media a growing field of study.

The simple nature and widespread popularity of Twitter are often favorable arguments researchers emphasize when selecting a social media platform to do event detection on. Cataldi, Di Caro and Schifanella [7] proposes a general event detector based on temporal and social terms evaluation. It combines the number of extracted words with the social authority of the Twitter user. Retrieved tweets is a sample from the Twitter firehose (all tweets posted in real time). Twitter users with many followers have a high authority and therefore has a bigger impact on the event detector. This detection method therefore emphasizes what Twitter users are reading and less what they are tweeting about. This user segregation means some events will be detected later or maybe not at all. This assumes users with high authority and to a lesser extent tweet babble. This claim is not backed up in the paper.

Li, Lei, Khadiwala and Chang [8] describes a crisis centric event detector able to detect crime and disaster related events. The system analyses tweets in a spatial and temporal setting. Retrieved tweets are based on search expressions. Even though search expressions are updated it is conceivable there are search expressions the system is unable to pick up and therefore unable to detect some disasters. This limits the event domain. The event detector does not actively retrieve tweets with geotags but exploits them when obtained. For tweets not containing the tags, the location is estimated based on the tweet text or Twitter friends. With people doing more traveling and getting friends all over the world this method might give inaccurate results. Another procedure could be to collect all tweets from a geographical limited area. This approach would leave out many tweets because most tweets does not have geotags. Because of the high rate tweets are posted it could in many cases significantly improve the execution performance of the event detector to only retrieve a sample of the tweets. This sample could be all geotagged tweets.

An ongoing Australian government project [9] seeking to detect crisis related events using a burst detection method which examines the stemmed unigram words in the tweets using a parameter free method[10]. To classify crisis events Support Vector Machines is utilized. The classifier is trained with a data set. The training data describes the limitations of the detection domain. The vocabulary from tweets about a forest fire is probably different from the vocabulary from tweets about a tsunami. Because crisis situations are unpredictable there are crisis events where this approach would fall short.

The event detector proposed in this thesis can be regarded as a combination of a crisis centric event detector and a general event detector. It aims at detecting crisis events by not distinguishing between general events and crisis events. To estimate the location of the event tweets utilized are geotagged. To detect events from a specific area it will compare density distributions in a spatial, temporal and pure frequency based context. Detected words are clustered together and then visualized.

1.2 Problem Statement

This thesis focus on detecting and gaining an overview of crisis situations by monitoring tweets from a specific area. The reason for utilizing geotagged tweets is to get a geographical overview of the crisis and be able to detect events from a specific geographical area.

The word ”crisis” fits the description of how Oxford dictionary defines the word ”event” [11]: *a thing that happens or takes place, especially one of importance*. A crisis can therefore be defined as an event.

There are already multiple papers [18][12] reporting good results on detecting general events in a Twitter environment, but none which utilize geotagged tweets. Exploiting geotagged tweets can give a better overview of a crisis situation. This is often done by crisis centric event detectors [21] [9] [13]. They do however limit their event domain by predefining events to earthquakes, typhoons, floods etc. by defining key words or trained classifiers. This can mean they are unable to detect some crisis situations in case they were unable to imagine the keywords during development or training. Supervised learning is probably better than just utilizing keywords, but would be inadequate when the vocabulary used in the training data is significantly different from the vocabulary used about the current crisis. Pure event detectors are better because they do not have such limitations on their event domain. This thesis proposes a combination of a typical crisis centric event detector with a general event detector. Making it possible to detect any event while the utilization of geotags makes it possible to estimate the area of the event by using a spatial detection method. Resulting in a unique combination not previously seen in this field of study.

This project does not seek to detect events as fast as possible. If a bomb goes of in a city it is not a goal to detect the event within for example 3 minutes. Approaches aiming at detecting crisis events as fast as possible are prone to a high degree of false positives. People playing games could for example use crisis related words like bombing, shooting etc. without making an impact on the current vocabulary of Twitter. The detection methods of this thesis are based on comparing density estimates. The number of event related tweets need to be high enough for the vocabulary in the tweets to make an impact on the total vocabulary for the detection duration. To be able to detect an event a couple of event specific words not commonly used need to be mentioned a few hundreds times each. Crisis centric event detectors might be faster, but not necessarily.

The solution is not a typical crisis centric event detector since it aims at detecting general events. It is therefore important to determine if it can detect general events as well as crisis events. If the latter is possible, it is reasonable to assume that also crisis events can be detected, as long as people are tweeting about the event in a similar manner as the kind of events we are investigating. These events might be theoretically difficult or maybe even impossible for typical crisis centric event detectors to detect. Because the main goal of this thesis is to detect crisis events it is important to check if the solution is able to detect crisis events and not only general events. The proposed solution does not distinguish between crisis events and general events. In reality there is no difference between a general event and a crisis event, but because the proposed solution is a mix of a crisis centric event detector and a general event detector it is important to verify both types can be detected.

From the discussions above, the following research questions are derived:

1. Is it possible to detect general events by comparing word densities, comparing kernel density estimates for each word in a spatial and temporal context and clustering the detected words with odds ratio?

2. Is it possible to estimate the affected area of an event occurring in a limited geographical area comparing bivariate kernel density estimates in a spatial context?

3. Is it possible to detect crisis events by comparing word densities and comparing kernel density estimates for each word in a spatial and temporal context and clustering the detected words with odds ratio?

4. Is it possible to estimate the affected area of a crisis event occurring in a limited geographical area comparing bivariate kernel density estimates in a spatial context?

The first research question is a general question and it may be the most important question of the four. Preliminary investigations suggests that this is possible, but it is important to answer it utilizing the proposed solution.

Research question two takes question one a step further and asks if it is possible to approximate the affected area of the event. To answer this question two coarse concepts are introduced, geographically Gaussian distributed events and geographically Uniformly distributed events. These are not Gaussian and Uniform distributions in their strict sense, but used as representations for events affecting those close by (Gaussian) and events affecting everyone to the same extent (Uniform). In reality these events will have other distributions, but they effectively represents the difference between the types of events. It is only possible to approximate the affected area of a Gaussian event.

A Gaussian event affects those close by the origin of the event more than people further away. An example could be a forest fire. Those loosing their house to the flames are more probable to tweet about the fire than people who live on the other side of the country not affected by the forest fire.

A Uniformly distributed event affects people equally regardless of where they are located in respect to the origin of the event. Launch of a new iPhone affects people equally over a large geographical area and is therefore a Uniform event. Figure 1.1 illustrates one Gaussian event and one Uniform event.

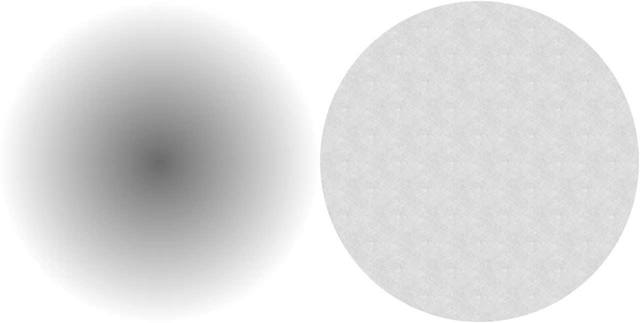


Figure 1.1: Both circles illustrate events originating in the center. The left circle illustrates an event which affects its surrounding area as a bivariate Gaussian distribution. The right circle illustrates an event affecting its surrounding area as a bivariate Uniform distribution.

Question three and four are similar to question one and two, but where the event is a crisis. Although question one and two will indicate if question three and four is possible it is important to verify this with real Twitter data.

Limitations and Assumptions

Two of the core detection methods in this project are based on temporal and spatial analysis. Some temporal and spatial boundaries need therefore to be set. This is to ensure a diverse and realistic data set is used when testing and that an appropriate number of tweets is used, not too many and not too few.

In order to capture and study both small (city) and large events (large part of continent/country) some commodious boundaries are set. The lower limit should be about the size of a large city and its surrounding area. An example is Greater Boston which is Boston and its surrounding area. An upper limit must be devised, due to the limitations of data retrieval. For this thesis the upper limit is about 40 % of the USA. This is only due to Twitter's rate limit. See Section 3.1 for more information about the rate limit.

The duration of each test should be more than or equal to 30 minutes. This to ensure the test data is diverse with respect to the temporal method. There is no upper limit, but it should be adjusted so that the number of tweets in a test is not too low or too high. If the number of tweets is too low the basis for creating reliable statistical models is poor. If the number of tweets is to high the detection methods might be too slow.

The tweets from the surveyed area is assumed to be a representative data set. Meaning the event detector would work in other English speaking and technologically advanced areas. It also assumes the collected tweets are from Twitter users with a wide demographic background [4].

1.3 Literature Review

Event detection using Twitter data has yet to become a large field of study, but some research has been done. The first subsection is a related-work-overview. The two next subsections are in-depth analysis of two papers.

1.3.1 Related Work - Overview

The reviewed papers can be divided into supervised approaches and unsupervised approaches in relation to event detection.

Supervised approaches are methods which need to set some predefined variable and thus limiting the scope of its detection domain. It can also be approaches which requires to learn how a tweet message of the desired event type is constructed by using training data. Supervised approaches are often quick to detect events.

Unsupervised detection methods has advantages when trying to detect a general event or a predefined event type, but might have problems with noise. Their detection domain is unlimited compared to the supervised approaches. They are often based on burst detection which compares word frequencies in a restricted duration. They are in most cases slower to detect events than some of the supervised approaches.

Supervised

Detecting Controversial Events from Twitter [13] first detects an event and then gives the event a controversy score. A controversial event is an event where people opinions are opposed. The event detected is related to a predefined entity (e.g. ”Barack Obama”). This approach is not intended to detect all kinds of event, but limited to one of the predefined entities. Accomplishing this is done by using supervised machine learning techniques and lexicons for sentiment, controversy, bad words etc. Experimental results are reported being promising.

Beyond Trending Topics: Real-World Event Identification on Twitter [14] trains a Naive Bayes classifier using standard machine learning techniques on some defined features. The approach clusters tweets on-the-fly by comparing the message similarity with existing clusters. If no cluster is similar enough, a new cluster is created. TF-IDF [15] weight vector is created for each tweet based on its textual content. The cluster similarity function is a cosine similarity metric [17]. Several feature categories are explored. Temporal features can detect if a word frequency is increasing. Social feature captures the interaction between users within a cluster. This can emphasize that the cluster is an event. A classifier was trained to distinguish between event and non-event clusters. The experiments were carried out by collecting all geotagged tweets from New York area for one month. The experimental results shows some promise, but the paper lacks a good analysis of the performance. There are too few examples.

Emergency Situation Awareness from Twitter for Crisis Management [9] is an ongoing work with the Australian government. The goal is to detect and assess crisis situations and forward tweets with valuable information to help crisis coordination. To detect incidents a burst detection method is used on the stemmed unigram words in the tweets. A burst is a positive change compared to the statistical model of the word. Historical data is used to build a statistical model of word occurrences. A classifier based on Support Vector Machines has been trained to identify certain crisis situations. An incremental clustering algorithm has also been developed. It clusters messages and topics over time. The paper does not give an in depth review of the results, but a summary of the their deployment experience.

Tweet Analysis for Real-Time Event Detection and Earthquake Reporting System Development [21] target some specific events like earthquakes, typhoons and traffic jams. The goal is to detect the mentioned events as fast as possible. A classifier based on a support vector machine is trained with earthquake data. The selected features are keywords, statistical data on the keywords and words describing the context (before or after an earthquake). Probabilistic models are created with a temporal and spatial context. The temporal model is a Poisson distribution. The spatial model estimates where the location of the event is occurring based on the location information from the tweets. This information is again based on where the Twitter user said to be his/her home. The test results produced by the event detector was good. Locating the event was difficult when the event occurred in sparsely populated areas or when the event was moving. An obvious flaw with this approach is the location data used. Because many of the Twitter users do not allow their geographical location when tweeting, it is likely most of the tweets would be appointed the wrong geotag. This could in many cases give the wrong geo location of the event. Another possible problem is the execution speed of the system.

Detection of Unusually Crowded Places through Micro-Blogging Sites [19] describes a procedure to detect areas which are becoming unusually crowded or becoming less crowded. The procedure relies on geotagged tweets. To find the spatial distribution of the data K-means cluster analysis is performed. This is first done on some normal data. Then it is possible to detect unusually crowded places. K-means clustering aims to partition n observations into K clusters. An aggregation model and a dispersion model is created using the clusters. The aggregation algorithm detects if Twitter users converge on one crowded region. The dispersion algorithm is reverse of the aggregation algorithm and detects if users are leaving an area. The paper presents too few experimental results to say if the proposed method in the paper is viable. The K-means clustering algorithm need to know in advance the number of clusters it should divide the observations into. This can be problematic because people are not clustered in a predefined number of ways. Another problem with K-means is that it is a NP hard problem.

Unsupervised

Event Detection in Twitter [12] applies wavelet analysis to detect events. Wavelet analysis can measure when and how a frequency of a signal changes. This is done by first constructing a signal for each word by applying wavelet analysis on the frequency of the word. Trivial words are then filtered away by looking at the corresponding signal auto correlation. The words surviving the filtering are displayed in the graph as events. An event must always contain two or more words. The detected events are also given an event significance value. The test data was from the 1000 most popular Singapore-based Twitter users and their Singapore-based followers. The solution is able to detect events.

Twevent: Segment-based Event Detection from Tweets [18] is an event detection which can be split into three components: tweet segmentation, event segment detection and event segment clustering. Tweet segmentation split the tweet text into segments or words. A segment can be ”steve jobs”. A segment is created by applying a function which measures the stickiness of the words in the segment. If the stickiness value is high the segment will not be more split up. The stickiness function utilizes statistical information derived from Microsoft Web N-Gram service and Wikipedia. The event segment detection component creates a binomial distribution for each segment within a time window. A segment is said to be bursty if the tweet frequency is greater than the mean value of the approximated Gaussian distribution. A bursty segment is potentially related to an event. The set of event segments detected is then clustered into groups in the event segment clustering component. Similarity between two segments is calculated using a similarity function. This function is based on their temporal frequency patterns. The obtained similarity values are then utilized in a k-Nearest Neighbor Graph to create clusters. Before a segment is declared to be event related the newsworthiness is calculated. This is done by checking if the segment is found in Wikipedia. The experimental results are very promising, but does not include information on the performance when no segments of an event is covered in Wikipedia.

1.3.2 Emerging Topic Detection on Twitter based on Temporal and Social Terms Evaluation [7]

The proposed topic detector in this paper is based on five steps. First terms are extracted from the retrieved tweets and formalized. Then a directed graph of the active users based on social relationships is calculated using a page rank algorithm. In the next step each term is modeled as a life cycle according to aging theory. To accomplish this the user authority is utilized together with the number of similar terms. The fourth step is selecting a set of emerging terms based on a calculated energy level. Finally a navigable topic graph is created based on the emerging terms. The paper uses ”term(s)” about word(s) in a tweet while keywords are words in a global context.

The extracted terms are assigned with an associated tweet vector which formalizes the information retrieved from the tweet. No stop words or stemming is utilized prior or during the extraction process which means different languages are no obstacle. Authors of this paper reckon the information flow starts in the geographical origin of the event and then spreads to larger geographical areas if the topic is interesting. The weight vector expresses the information expressed by each collected tweet in the considered time interval.

User authority is calculated from a directed graph. The graph is a model of the relationships between active Twitter users. The authority value is based on how many people are following you or in technical terms how many inbound edges a user has. A Twitter user with high authority has its tweets read by a large group of people and thus having a large influence on the Twitter community. The dangers of such an approach are the authorities setting the agenda. This makes it more difficult for Twitter users with low authority to make an impact, even if there are a number of users tweeting about the same topic. The authors uses an approach where what is read is the emerging topic while what is written is having a smaller impact on the emerging topic. It is not necessarily a correlation between what Twitter users are reading on Twitter and what they are tweeting.

Content aging theory is inspired by aging of living organisms. The organism in this context is a term. For the term to live long it will need nourishment. This is other tweets containing similar information. Naturally when there is no nourishment the term dies. The vitality of the term can be measured by the energy level of a keyword. A high energy level means the term is becoming more popular while low energy means it is becoming less popular. A term having lots of nutrition during a certain time period is considered hot. In other words if a term is used much more than usually it is considered hot. To do this evaluation the authors of this paper uses a temporal evaluation.

The selection of emerging topics can be divided into supervised and unsupervised. The supervised selection is dependent on the user setting a threshold parameter for the energy level. A term should have been selected before it is selected as an emerging event. The unsupervised selection is based on the fact that setting a numeric threshold can be difficult for any user. To do this automatically the keywords are ordered in descending order based on their energy level. Then an average energy level is found and a critical energy value calculated. Every keyword having an energy level above the critical level is selected as an emerging term.

An emerging topic is defined as a minimal set of terms which are semantically related to an emerging keyword. To detect an emerging topic all tweets within the specified time frame are evaluated. Finally a topic graph is created using correlation vectors. These vectors relates the emerging keywords with a topic.

The presented results are promising, but reproducing the results could be quite demanding. The described procedure is comprehensive with many details which are dependent on each other.

1.3.3 TEDAS: a Twitter-based Event Detection and Analysis System [8]

Tedas is an experimental Twitter event detection system which seeks to detect new Crime and Disaster related Events (CDE). This is done by analysing tweets from a temporal and spatial point of view. The system also identifies and ranks the importance of CDE.

A retrieved tweet is fed to a classifier which determines if the tweet is a CDE. The CDE is sent through a meta information extractor which retrieves temporal and spatial information. The meta information together with the tweet is indexed by a text search engine and stored in a database. The tweets are ranked by a model according to how important they are. A clustering model groups similar tweets based on spatial and temporal tweets into geographical regions or temporal ranges. These results can later be visualized.

The system is based on Java, PHP, MySQL, the Twitter api, Lucene and Google Maps. It enables the user to search for a keyword (e.g. tornado) over a specified time period and a geographical area.

Since monitoring all CDE tweets is impossible the authors had to limit their search according to the Twitter api. The options are to get a sample of all tweets, get all tweets containing certain words or tweets from certain users. The authors used a tracking rule containing a set of keywords related to CDE tweets. To maintain and update the tracking rule, the system analyses the CDE tweets and uses the existing keywords as seeds to detect new keywords. The new keywords are then evaluated by using them and analysing the retrieved tweets to verify they contain CDE information. If they do not contain the keyword they will be dismissed. The authors estimate they are able to crawl 85 % of all CDE tweets.

To classify a tweet as CDE the text is analyzed. The authors discovered certain properties in CDE tweets like a time, a number (e.g. 7 people are injured). They also utilized Twitter specific language like @David means replying to David and is often related to personal communication.

To locate where a tweet originated from the authors uses the GPS tag in the tweet. The only problem is that most tweets does not have a GPS tag or any location specific meta data. To compensate for this lack of location specific data they uses an algorithm to locate the origin of the tweet based on the tweet text. The algorithm analyses the text for a geographical location. If it is not present the algorithm will look at where the friends are located. The idea is that you are not too far away from your friends most of the time. Twitter is global media. When a flooding or an earthquake occurs it is not only the people in the immediate area who tweets about it, but people all over the world. A major part of the tweets will contain inaccurate location information since most people are far away from the events. According to the authors the algorithm has a 63 % accuracy.

The ranking is done by content features, user features and usage features. Content features are words related to CDE in the tweet text or in a link provided in the tweet. User features is used to detect the authority and credibility of the user. A verified account (e.g. news agent or police department) is more likely to contain "correct" information than the average Twitter user. An active Twitter user which has many followers has a higher credibility than users with few or none followers. Usage feature is about how far a tweet spreads. The wider it spreads the more significant it is. To accomplish this the authors have looked at the number of re-tweets.

The paper has has a lack of detail and particularly in the areas of related work and the algorithms used. The approach seems simple and straightforward with a few exceptions, such as search term updating procedure. Reproducing the results is hard both due to the lack of details and that it would require some programming effort. Verifying the results are difficult since the algorithm details are sparse and a particular data set not provided.

1.4 Solution Approach

The strategy to detect general events is illustrated by the solution pipeline in Figure 1.2.

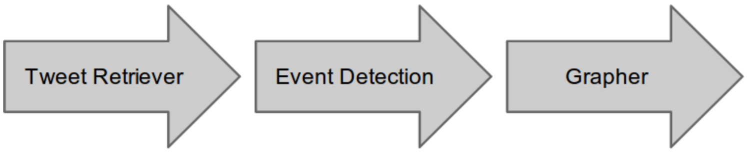


Figure 1.2: Solution pipeline.

The first part of the pipeline is the tweet retriever. This is essential, because it collects tweets from the Twitter API and persists them to a database. The event detection consists of multiple methods executed in succession. This is later called the detection pipeline. Grapher presents the results on a website.

The detection pipeline consists of a tweet parser, three different core detection methods, a clustering method and a bivariate spatial detection method utilizing the clusters.

The tweet parser removes the most frequently used words and counts the occurrences of each word and keeps track of which tweet they were found in. In short the parser converts the tweet text to formats which can easily be handled by the following methods.

The three core detection methods are refined variants from a preliminary approach[1]. Two of them use kernel density estimation in a spatial and temporal context to build a historical statistical model and a new statistical model for each word. These are then compared to each other using a test [20] to find significant differences. The last method compares word density distributions. These methods work independent and discover words which have a positive value compared to the historical data.

Kernel density estimation is a method of creating probability density functions which are not depending on predefined shapes like the Gaussian distribution. A kernel density estimate can look like any function. A bandwidth parameter determines whether the function is oversampled or undersampled. Calculating these parameters have become a field of study [16].

The third part of the event detector reduces the number of discovered words by applying odds ratio calculations. This can be seen as both a noise filter and a refinement of the results from the three detection methods. The clustered words represents events where one cluster can be related to one or more events. An event which is only described by one word is therefore not possible with this system. For a word to survive this process it has to be strongly related to another word detected by by the three detection methods. An example of two words which are strongly related are; ”happy” and ”birthday”.

The last part of the event detector is comparing two bivariate kernel density estimates. This approach is quite similar to the spatial method and the temporal method described above. All of them compares kernel density estimates. Where the spatial and temporal method utilize univariate data this method use bivariate data. The statistical models are constructed from all the words in the graph. The goal is to see if the density distribution of the words in a graph is different compared to historical data. If Twitter users are tweeting about a forest fire one week and the week after there is a forest fire in the same region and the solution approach is only using the week before to build the historical data, the solution would probably be unable to detect the forest fire. Twitter users close to the forest fire will tweet more about the fire than those who are further away. This would result in two different density distributions. Comparing them would help approx-

[1] Eskeland, E (2012). General Event Detection Using Twitter Based on Comparing Kernel Density Estimates and Word Density Distributions

imate the area affected by the fire.

The last part of the detection pipeline is visualization. It graphs the selected words as nodes and the calculated odds ratio as edges between the nodes. The size of the node tells the frequency of it compared to the rest words in the graph. The width of the edge between two words is calculated from the odds ratio compared to the other odds ratios in the graph. A wider connection means stronger relation between the two words.

Compared kernel density estimations and odds ratio is new in this field of study. The proposed event detector is therefore significantly different from other event detectors.

1.5 Contributions

The main research contribution of this thesis is the event detector and particularly the spatial detection methods and the clustering approach with odds ratio.

The use of kernel density estimation is a new approach in this field of study. Together with the statistical method [20] developed by Duong to compare two kernel density estimates the proposed methods are a convergence of technologies which has not previously been explored.

The spatial detection methods provide a new approach to detect words and also locate where people are using them. Estimating the affected area of the event is a new contribution which has not been seen before. These methods are based on the word's geographical densities. For example could the word ”arena” have a different density function compared to the historic density function. As sporting events occur on different arenas around the country, Twitter users use the word ”arena” from different locations.

The temporal method is a new approach to detect if the usage of a word have elevations compared to the historical statistical model in the temporal duration of the test. This method only provides a new approach to a previously solved problem [7].

A typical approach to detect events [9][18] is to use word frequency analysis within a specified time duration. This is not a novel approach in this field of research.

Applying odds ratio to cluster words together is also a new approach in this field. This clustering process has the benefit of removing a great deal of noise and helps improve readability for the analyst when the words are later presented as graphs.

Retrieving tweets only containing geotags is not a common approach. The only known approach to retrieve geotagged tweets for an event detector is [14]. It does not utilize geotags for detection purposes. The goal for using this retrieval approach was to limit the number of tweets. Because of the vast amount of tweets produced other approaches have to use a sample from the Twitter Firehose or restrict their tweet retrieval with for example keywords. Retrieving only geotagged tweets might be considered as a sample from a specific area.

1.6 Report Outline

To just get an overview this thesis it is enough to read the abstract, the first part of the introduction and the conclusion. To gain a deeper understanding of the thesis it should be read in a chronological order. Chapter one gives an introduction to the background for this project and previous work in this field. Chapter two can be dropped if the reader already have an understanding of the presented theory. If not, it is important to read this chapter making sure the concepts in the next chapter is understood. This brings us to chapter three which describes the solution. It describes all parts of the solution with focus on the detection pipeline and less on more trivial implementations like the tweet retriever and Grapher. Chapter four discusses the test results explaining how well it is able to detect crisis events and general events. The research questions raised in the introduction chapter are answered. It also discusses different aspects of geotagged Twitter data with more. Chapter 5 sums up the project in a conclusion.

Chapter 2

Theoretical Background

2.1 Kernel Density Estimation

Kernel Density Estimation (KDE) is a non parametric method to estimate a probability density function (PDF). Parametric versions like the Gaussian distribution is widely used in various areas. They work well when the data fits into the predefined distribution. It is easy to calculate the parameters for such distributions e.g. $N(\mu, \sigma^2)$. But the lack of flexibility is one of its weaknesses.

A well known non parametric distribution is histograms. To create a histogram the first order of business is to select the left most point x_0 which is lowest value to be added to the histogram. The estimator for a histogram is given by

$$\hat{f}(x) = \frac{1}{n} \frac{\text{Number of observations within the bandwidth of x}}{\text{Width of bin for x}} \tag{2.1}$$

where n is the sample size. The calculations are simple. The most difficult part is estimating the bin width also known as the bandwidth. If the bandwidth is wide the degree of smoothing will be high and opposite if the bandwith is low. Figure 2.1 illustrates a kernel density estimate and the corresponding histogram with optimized bandwidth and kernel.

The bandwidth problem also arise when calculating the kernel density estimate. The formula to calculate a kernel density estimate is

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{2.2}$$

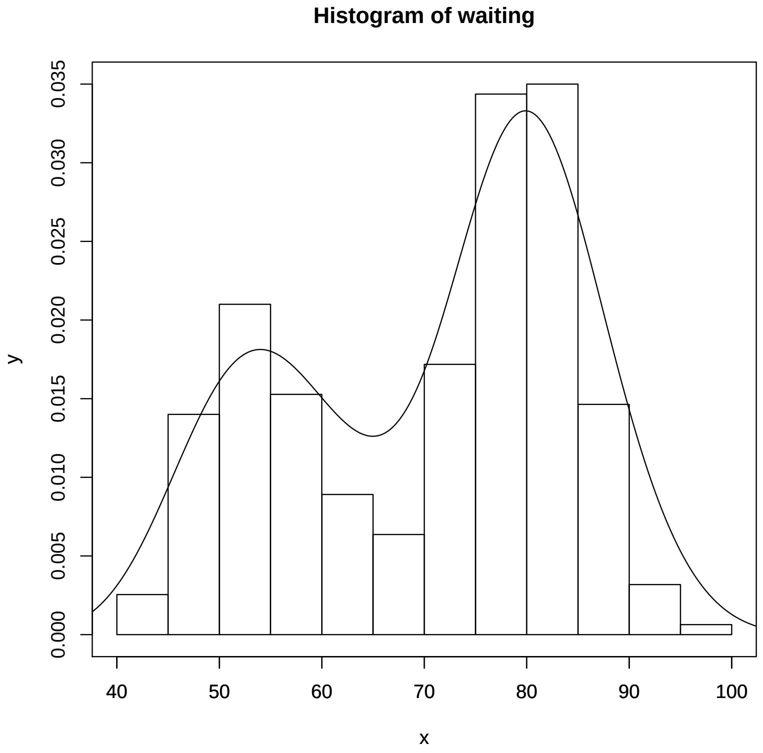


Figure 2.1: Kernel density estimation and the corresponding histogram with optimized bandwidth. Y is the density function.

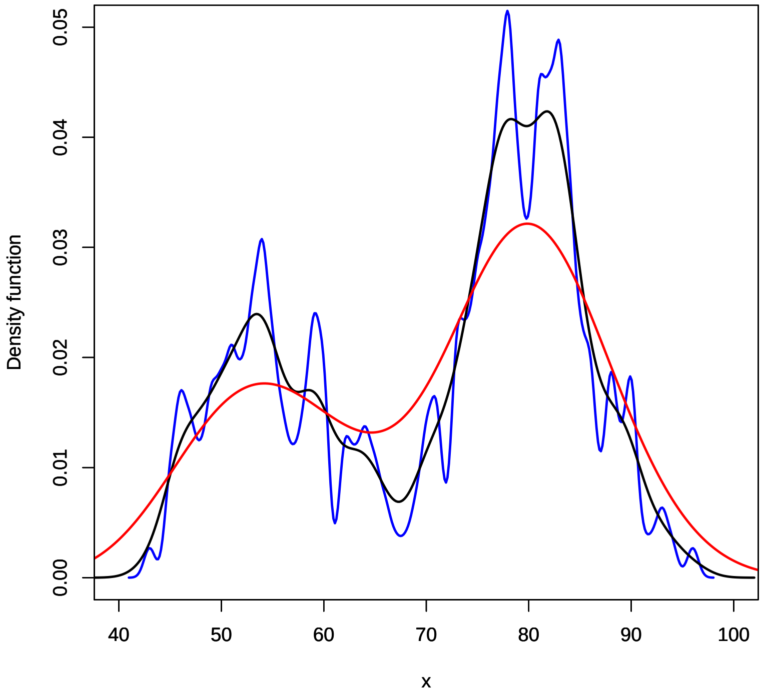


Figure 2.2: Kernel density estimates with different bandwidths. The black graph has an optimal bandwidth. The red graph has a too high bandwidth. The blue graph has a too low bandwidth.

where h is the bandwidth. $x_1, x_2, .., x_n$ are variables from some distribution. K is a kernel which has to be a symmetric function which integrates to one. Examples of kernel functions are Gaussian, triangular, rectangular, biweight etc.

While the smoothness is determined by the bandwidth, the shape is determined by the kernel. A common kernel is the Gaussian density distribution.

Figure 2.2 illustrates different bandwidths. The red kernel density estimate has a bandwidth of 30, the blue has a bandwidth of 0.3 while black has the optimal bandwidth of 2.64. The optimal bandwidth is calculated from the sample set.

It has been put a lot of research in finding the optimal bandwidth parameter [27]. Common for the different solutions is they are based on an error criteria. When the error value is minimized the bandwidth is found. The error function calculate the distance between the estimate $\hat{f}$ and the target density f. One commonly used error function is the Mean Integrated Squared Error (MISE)

$$MISE(\hat{f}) = E\int_{-\infty}^{\infty}(\hat{f}(x) - f(x))^2dx \tag{2.3}$$

From the minimum MISE it is possible to find the optimal bandwidth with

$$h_{opt} = \left(\frac{1}{n}\frac{\gamma(K)}{\beta(f)}\right) \tag{2.4}$$

where $\beta(f) = \int f''(x)^2dx$ and $\gamma(K) = j_2k_2^{-2}$ where $j_2 = \int K(z)^2dz$ and $k_2 = \int z^2K(z)dz$. The kernel K and the number of samples n is important to find the optimal bandwidth. $f''(x)$ describes the curvature in x.

One of the better bandwidth selectors for most data sets is the ”plug-in” selector. It is the same as h_{opt}, except $\beta(K)$ is replaced by an estimator.

It should also be noted that the $MISE(\hat{f})$ can be used calculate the best kernel.

2.2 Comparing Two Kernel Density Estimates

Much research have been done in comparing two sample univariate data. The most known methods are the Kolmogorov-Smirnov, Wald- Wolfowitz and Mann-Whitney tests. The method described here is based on [20] by Duong and is found as the method ks.local.test in the R library ”ks”. The method seeks to find if two data sets are different. If they are different it also figures out where they are different.

The test is based on the null hypothesis test $H_0 = f_1 \equiv f_2$, where f_1 and f_2 are the two density functions.

The test can further be divided into a global significance test and a local significance test. The global significance test conceals the details of the data sets and computes if they are different with

$$T = \int_{\mathbb{R}^d}(f_1(x) - f_2(x))^2dx \tag{2.5}$$

The local significance test further investigates which regions are different in the two data sets with the local hypothesis

$$H_0(x) : f_1(x) - f_2(x), x \in \mathbb{R}^d \tag{2.6}$$

It should be noted that it is possible to use this method for both univariate data and multivariate data.

2.3 Odds Ratio

Odds are the probability of an event occurring divided by the probability of the same event not occurring. Converting probability to odds is done by $probability/(1-probability) = odds$. Odds ratio (OR) is a measure of effect size or the strength between two binary variables. OR is often used in case control studies by medical researchers. An example could be to see if there is a correlation between smoking and cancer. In such a study there would be two binary variables $smoker = \{1, 0\}$ and $cancer = \{1, 0\}$.

	$cancer = 1$	$cancer = 0$
$smoker = 1$	$n11$	$n10$
$smoker = 0$	$n01$	$n00$

where $n11$ is the number of smokers having cancer. $n10$ is the number of smokers not having cancer. $n01$ is the number of non smokers having cancer and $n00$ is the number of non smokers not having cancer. OR can then be calculated using

$$OR = \frac{n11/(n11+n10)}{n10/(n11+n10)} \Big/ \frac{n01/(n01+n00)}{n00/(n01+n00)} = \frac{n11 * n00}{n10 * n01} \tag{2.7}$$

If $n11 = 90$, $n10 = 10$, $n01 = 20$ and $n00 = 80$ OR is

$$OR = \frac{n11 * n00}{n10 * n01} = \frac{90 * 80}{10 * 20} = 36. \tag{2.8}$$

Based on the provided data a smoker is $90/20 = 4.5$ times more likely to get cancer and an OR of 36. It is also possible to find the OR for a non smoker compared to smoker. A non smoker is $1/36$ more likely to get cancer than a smoker.

Chapter 3

Solution

The solution consists of three Python modules. These can be regarded as a solution pipeline as seen in figure 3.1.

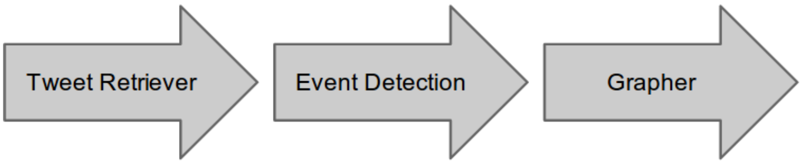


Figure 3.1: Solution pipeline.

For the solution to work tweets need to be fetched. This is done by the Tweet Retriever. When enough tweets have been collected, the Event Detection can start working. This is by far the most complex part and can be considered as the core section of the solution. When events have been identified they can be visualized by Grapher.

3.1 Tweet Retriever

Tweet Retriever receives tweets from the Twitter API and stores them in a MySQL database. A preliminary version of this section was developed by the author of this thesis in an earlier project [1].

[1] Eskeland, E (2012). General Event Detection Using Twitter Based on Comparing Kernel Density Estimates and Word Density Distributions

The Twitter API has two methods of retrieving tweets. A REST[2] service and a streaming service. Both services provide a whole range of search parameters. The REST service has limits on the search parameters, number of queries per 15 minutes and therefore has a limited number of results. One of the limited search parameters is the size of the bounding box which does not cover more than a small city. On the upside it is possible to retrieve tweets up to 2 - 3 weeks back in time. The streaming service delivers tweets real time. Where the upper limit is 1 % of the Twitter firehose. This is to reduce the stress on Twitter's server. Twitter processes 500 million tweets per day [3]. Serving such amounts to everyone who requests it would be close to impossible. The Twitter firehose is *all* tweets posted in real time. Because this project demanded a great deal of tweets the streaming service was used.

The streaming service's boundary box does not have a limit on the geographical area it can retrieve tweets from. It is specified by coordinates of the south-west corner of the box and the coordinates of the north-east corner. No other search parameters need to be specified.

The requirements of the application dictated it should be simple, efficient and stable with respect to up-time. To accomplish this the application has been developed in Python for simplicity. Connectivity to the Twitter API is accomplished through a Python library called Tweepy. This has made it possible for the application to work as stable as possible. The starting point for the application was a Tweepy example where the application overrides a method receiving the tweets. In this method the tweet is converted to the correct table format and stored in the database.

When Tweepy is unable to correct or handle the errors, the application crashes. These unforeseen events are in most cases due to temporary loss of internet connectivity and are outside the application and server domain. The solution is a script which starts the Twitter retriever when it has exited. It also logs the date and time of the fault.

3.2 Event Detector

The event detector can be described as a detection pipeline, visualized in Figure 3.2.

[2] REST stands for **Re**presentational **S**tate **T**ransfer. It is based on a stateless client - server architecture.

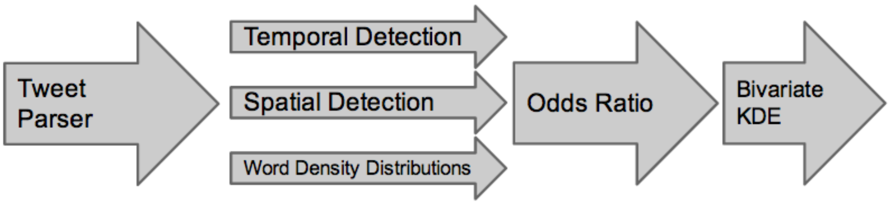


Figure 3.2: Detection pipeline.

The first part of the pipeline is the Tweet Parser. It splits up the tweet text into words and applies a stop word filter. Each word is accumulated into a feature. A feature got the number of occurrences of a word including all the geographical positions and timestamps where the word was used. The third step is the three detection methods. These methods provide the functionality which locates words possibly related to an event. The two first methods are based on spatial and temporal use of the words. The last method is based on comparing the relative frequency of a word. Each of the pipeline sections will be thoroughly explained in the following subsections.

3.2.1 Tweet Parser

Tweet Parser produces two different outputs. The first is a list of parsed tweets. Each parsed tweet contain the tweet id and a list of all processed words. The other output is a dictionary of all the words including their properties. A word in the dictionary contains the following properties

- string representation of the word,
- tweet ids,
- longitudes and latitudes,
- timestamps.

The properties are in plural form because one word can be in multiple tweets. For example could the word *basketball* be in many tweets.

Parsing each tweet text into words contains many steps. First all punctuation is removed from the text. Then the text is converted to lower case.

The process of splitting the tweet text into words is demonstrated in the following algorithm.

```
for $tweet$ in $tweets$ do
    $txt \leftarrow tweet.txt$;
    $remove\_punctuation(txt)$;
    $to\_lower\_case(txt)$;
    $normalize\_space(txt)$;
    $words \leftarrow txt.split(" ")$;
    $lemmatize(words)$;
    $remove\_stop\_words(words)$;
    $add\_words\_to\_parsed\_tweets(words, tweet.id)$;
    $add\_words\_to\_dictionary(words, tweet)$;
end
for $word\_feature$ in $dictionary$ do
    if $word\_feature.mentions < 40$ then
        $word\_feature.remove$;
    end
end
```

Algorithm 1: Splitting tweet text into words.

Removal of punctuation and conversion to lower case are self-explanatory. Normalize space will replace multiple subsequent spaces with a single space. This makes it easy for split to work as intended. The next steps in Algorithm 1 is to remove potential noise and highlight words which are potentially connected to an event.

Lemmatize the words is one of the most important steps. A lemmatizer from Python NLTK (Natural Language Toolkit) is used to lemmatize [29] each word. Lemmatization in linguistic is the process of grouping together inflected versions to a lemma [28]. This is the base of the word or dictionary form of a word. For example would $lemmatize("cars") = "car"$. An alternative which also reduce the number of inflected versions is a stemmer [30]. This finds the root of the word by using an algorithm. This approach was found to intrusive in this context. For example would it $stemmer("president") = "presid"$. If this was used in an index of a search engine it would be fine, but since the word is visualized in its processed form it is important that the word has a meaning.

Before the stop word list is applied common contractions for English are replaced with their proper form. An example of "I would" when contracted is "I'd". The stop word list can then be used remove commonly used words in English and in Spanish. Examples of these are "the", "is" etc for English and "es", "el" etc for Spanish. The list also contains letters and the numbers 0 - 9. The benefit of removing commonly used words is to avoid noise and keep focus on the special words describing an event.

Words in the dictionary with less than 40 mentions are dropped. This is also to avoid noise and help highlight the words which are related to events.

3.2.2 The Three Detection Methods

The three detection methods utilize different properties of the tweets. The first method analyze the spatial property. The second method is based on temporal analysis. The third method compare word densities. A preliminary version of these detection methods was described in a former project [3].

All the three independent detection methods are based on comparing new data with a historical data model. In most performed tests the historical data model comprises 3 different time frames Δt_h. These are the day before, a week before and two weeks before. If a test for $\Delta t =$ 2013.02.17 14:00 - 16:00 was performed the historic data model would consist of

- $\Delta t_{h1} =$ 2013.02.16 14:00 - 16:00,
- $\Delta t_{h2} =$ 2013.02.10 14:00 - 16:00,
- $\Delta t_{h3} =$ 2013.02.03 14:00 - 16:00.

The raw tweets from the time frame Δt_h is parsed in the same manner as Δt by Tweet Parser. After the historical data has been processed by Tweet Parser it is merged together to form one historical data model. This is done by transferring the data to a single data structure and discretize the time. All tests are performed from the same time of day. This enables us to convert the time from *HH:mm - HH:mm* to 1 - T making it possible for the temporal detection method to work. *HH* is hours and *mm* is minutes. The spatial method and the Word Density Comparison (WDC) method does not need similar data conversions to function.

The event detector is a mix of Python and R with the Python library rpy2 as the mediator. The R library ks is used for the statistical calculations while Python does everything else.

The input to the event detector is Δt and a boundary box of what area the tweets should originate from. The use of the three methods are visualized in the following

[3] Eskeland, E (2012). General Event Detection Using Twitter Based on Comparing Kernel Density Estimates and Word Density Distributions.

algorithm.

```
$dict_{new}, dict_{historic} \leftarrow$ **get_words**$(new\_date\_time)$
**synchronize**$(dict_{new}, dict_{historic})$
**compare_word_ensity_distributions**$(dict_{new}, dict_{historic})$
**for** $word_{new}, word_{historic}$ in $dict_{new}.values(), dict_{historic}.values()$ **do**
    **if** $word_{new}.mentions > min\_mentions$ and $word_{old}.mentions > min\_mentions$ **then**
        **compare_temporal**$(word_{new}, word_{old})$
        **compare_spatial**$(word_{new}, word_{old})$
    **end**
**end**
**store_to_db**$(dict_{new}, dict_{historic})$
```

Algorithm 2: The three detection methods.

get_words use Tweet Parser to create $dict_{historic}$ and $dict_{new}$. The dictionary keys is a string representation of the word it represents. The values are objects of type Feature. The Feature object contains the following properties string representation of the word, a count of all the mentions of the word, a list of all tweet ids where the word was found (multiple ids are possible), list of date time objects, list of longitudes and list of latitudes.

$synchronize$ adds the words which are found in $dict_{new}$, but not in $dict_{historic}$ to $dict_{historic}$. This is also done the other way around. This because consistency is important when comparing word for word in the different detection methods.

For the spatial and temporal method to be executed there need to be a certain amount of data. This is determined by $min\_mentions$. An appropriate value for this parameter has been empirically found to be 100. The results with this parameter is good. See the result section in chapter four. The parameter could probably be adjusted if there were very few or very many tweets used by the test.

The result from the detection methods are added to the corresponding Feature object in $dict_{new}$. If the temporal method detects significant difference, the date time objects which was the cause is added to the correct Feature object. The same goes for spatial detection which returns longitude and latitude. The word density comparison (WDC) adds the positive difference between new and historic data.

When the detection process has finished the results they are persisted to disk. There are many different formats of the produced results. Most important is $dict_{new}$ which have been updated with results. Together with $dict_{historic}$ and object representations of their parsed tweets they are stored in a result database. This database is later used by Grapher to present the results. The rest of the results are persisted as four different files. The first file contains multiple graphs for each word for temporal and spatial event detection in addition to a single graph representing the result of WDC for new data and historic data. A graph showing the difference for the two graphs is also provided. When the new data has a higher density the comparison graph is negative and opposite when the historic density is higher. This behavior is illustrated in Figure 3.5. These comparison graphs sometimes has red markers to illustrate significant differences. According to the R library used to create the graph the markers are not always working as intended and should be disregarded. The rest of the files are text files. The first non graphical file is a plain list of words which was triggered by any of the event detection methods. The second non graphical file lists the same words as the previous, but also adds an explanation why the word was selected. The third non graphical file lists tweets containing the selected words ordered by most selected words first. Overall the results gives a good foundation to evaluate the test results.

WDC - Word Density Comparison

This method creates word densities from $dict_{new}$ and $dict_{historic}$. From the dictionaries it is easy to extract $mentions_{new}(w)$ and $mentions_{historic}(w)$ for each word w in $dict$. Both dictionaries must be synchronized so they contain the same set of words, thus adding Feature objects for words with zero mentions. The word densities both sum to one. The percentage for w mentions is calculated with

$$p(w) = \frac{mentions(w)}{\sum_{\in words} mentions(j)} \tag{3.1}$$

where $mentions$ is the number of times word w occurs in the corresponding data set. $mentions$ is replaced with $mentions_{new}$ and $mentions_{historic}$ to calculate $p_{new}(w)$ and $p_{historic}(w)$. The denominator is the sum of all mentions in $words$.

The difference $difference(w)$ of historic and new is calculated with

$$difference(w) = \frac{p_{new}(w)}{p_{historic}(w)} - 1 \tag{3.2}$$

With the difference calculated it is necessary to determine a threshold for when

$difference(w)$ is large enough to suspect w is connected to an event. The threshold is calculated with

$$threshold(p_{new}(w)) = max(p_{new}) * \left(\left(\frac{minDifference}{maxDifference} \right)^{\frac{1}{max(p_{new})}} \right)^{p_{new}(w)} \tag{3.3}$$

where $max(p_{new})$ is the value of the largest share in the new word density. $minDifference$ and $maxDifference$ are empirically found constants with values 0.135 and 0.6. Figure 3.3 illustrates the generic $threshold$ function. If $p_{new}(w)$ is small the difference must be larger than if $p_{new}(w)$ is large. This provides some dynamics to detecting significant changes. A significant change is detected when $difference(w) > threshold(p_{new}(w))$.

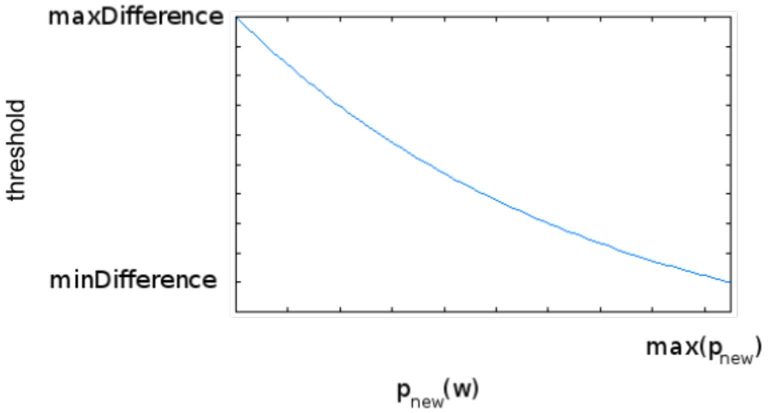


Figure 3.3: $threshold(p_{new}(w))$

Figure 3.4 shows the historic word density (black), the new word density (blue), the detected changes (red) and all put into the same graph. Each bar represents the mentions of a word. For graphing purposes the word densities are ordered in descending order based on $mentions_{historic}$. It should be noted that the bars are a little wider than they are in reality i.e. they overlap each other. This is due to the large number of word mentions being illustrated. The blue graph should have some holes, but these are difficult / impossible to spot.

For the cases where $mentions_{new}(w)$ has more than 100 mentions and $mentions_{historic}(w)$ has zero mentions $mentions_{new}(w)$ is considered significant. It could have been a scaling variable according to the number of tweets analyzed, but 100 mentions have been found adequate.

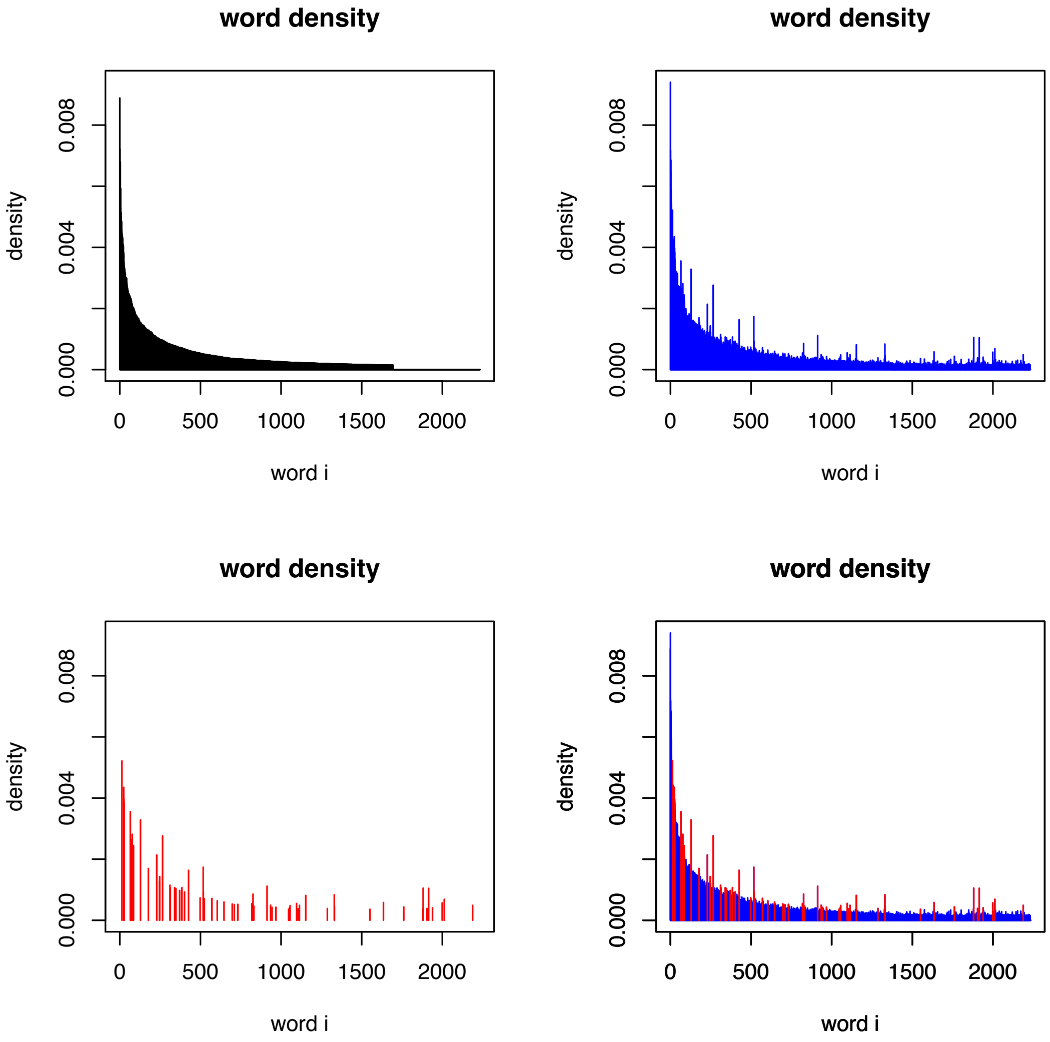


Figure 3.4: The first is the historic density ordered in descending order. Next is new ordered in the same order as the previous. First graph on the second row starts with the significantly different words. The last graph puts it all together.

It should be noted that many other solutions have been considered. Mean and variance were considered, but the data is too sparse. For a single word there are only three different measurements i.e. it would have used the same approach as the solution where each historic word is compared to each new word. Another approach would be to view all the data as one and not compare word to word. This could have been done by either using a kernel density estimate or zeta distribution to describe the word mentions. Problematic with such an approach is the over-smoothing which could occur. If $mentions_{new}(w)$ were to be significantly larger than $mentions_{historic}(w)$ it is a good chance it would be neglected.

The Temporal Method - Temporal Comparison of Kernel Density Estimates for Each Word

Detects if words are being used unusually much within a specified duration compared to statistical model. A probability distribution for word w is calculated using the R library ks. The distribution is a Univariate Kernel Density Estimate (KDE) for new data and historic data. Then the two KDEs are compared using kde.local.test from ks. The result is a list of binary numbers where one mean there is a significant change and zero mean there is no significant change. The positions of the ones in the binary list is used to find the timestamp where the significant change was found. Figure 3.5 illustrates on the left one kernel density estimate for the word "story" and on the right the result of the comparison. The black graph is the based on the historic data while the blue graph is based on the new data. When the comparison graph is positive the historic KDE is larger than the new KDE and opposite when the new KDE is larger.

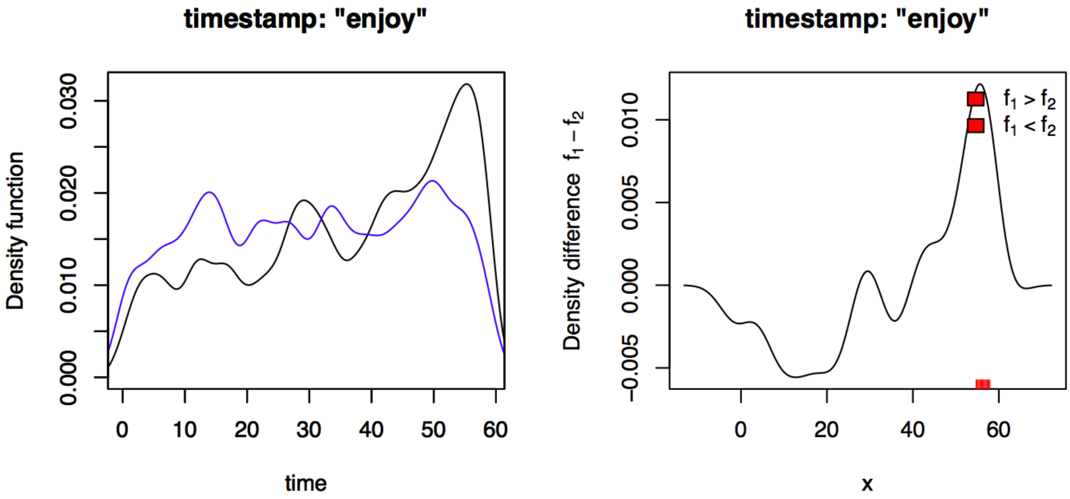


Figure 3.5: The graph on the left side is new (blue) KDE and historic (black) KDE. On the right side is the comparison of the two graphs. f_1 is historic, while f_2 is new.

The input date time can be Δt = 2013-03-17 14:00 - 18:00. Date time associated with a word is translated into a simplified timestamp. In general this is done by simplifying [*HH:mm* - *HH:mm*] to $0 - T$ where *HH* is hour and *mm* is minute.

For the mentioned example which has a date range from 14:00 to 18:00 the corresponding converted timestamp could be from 0 to 120. This is essential to be able to calculate the KDE (comparison). One time step is equal to two minutes. If the time step was larger it would result in more smoothing of the KDE. The same effect could be accomplished by using a larger bandwidth. Since the bandwidth is automatically calculated it would be inadvisable to set the bandwidth manually to something larger or smaller than the optimal bandwidth.

For the distribution to work properly a lower boundary has been set on the amount of data necessary to create the models and perform the comparison. Both new and historic must have more than 100 mentions for KDE and comparison to be calculated. This boundary is the same as $min\_mentions$ in Algorithm 2.

The Spatial Method - Spatial Comparison of Kernel Density Estimates for Each Word

This method compares the historic probability distributions with the equivalent new probability distributions in a spatial context. To maintain a certain efficiency of calculating KDEs, longitude and latitude has been split up into two separate detection methods. Figure 3.6 illustrates the probability distribution for longitude and latitude together with a comparison. The two methods independently detect words having an untypical spatial origin.

When a significant change has been detected a list of binary values with ones is returned. Using the positions of the ones the longitude or latitude of the significant change is found.

The R library ks is used to calculate the KDE and perform the comparison of the historic density distribution with the new density distribution. Ks supports comparison of univariate KDEs and multivariate KDEs. The possibility of multivariate comparison is one of the reasons why ks is chosen instead of a straight forward Kolmogorov-Smirnov approach [31]. In addition it is neatly wrapped into ks. The problem using bivariate or higher when the number of mentions rises is that the calculations take too much time. Figure 3.7 illustrates how such a KDE could look like. By not using this method some details are lost, but a lot is gained in performance. The functionality to create these KDEs and corresponding comparisons has been developed, but are not used as one of the three detection methods due to their time consuming nature. One solution to overcome this problem could be to only use a sample from the mentions of a word. This idea has not been realized and could be possible to loose something if a large portion of the data is removed. A possible outcome could be over-smoothing and therefore loss impor-

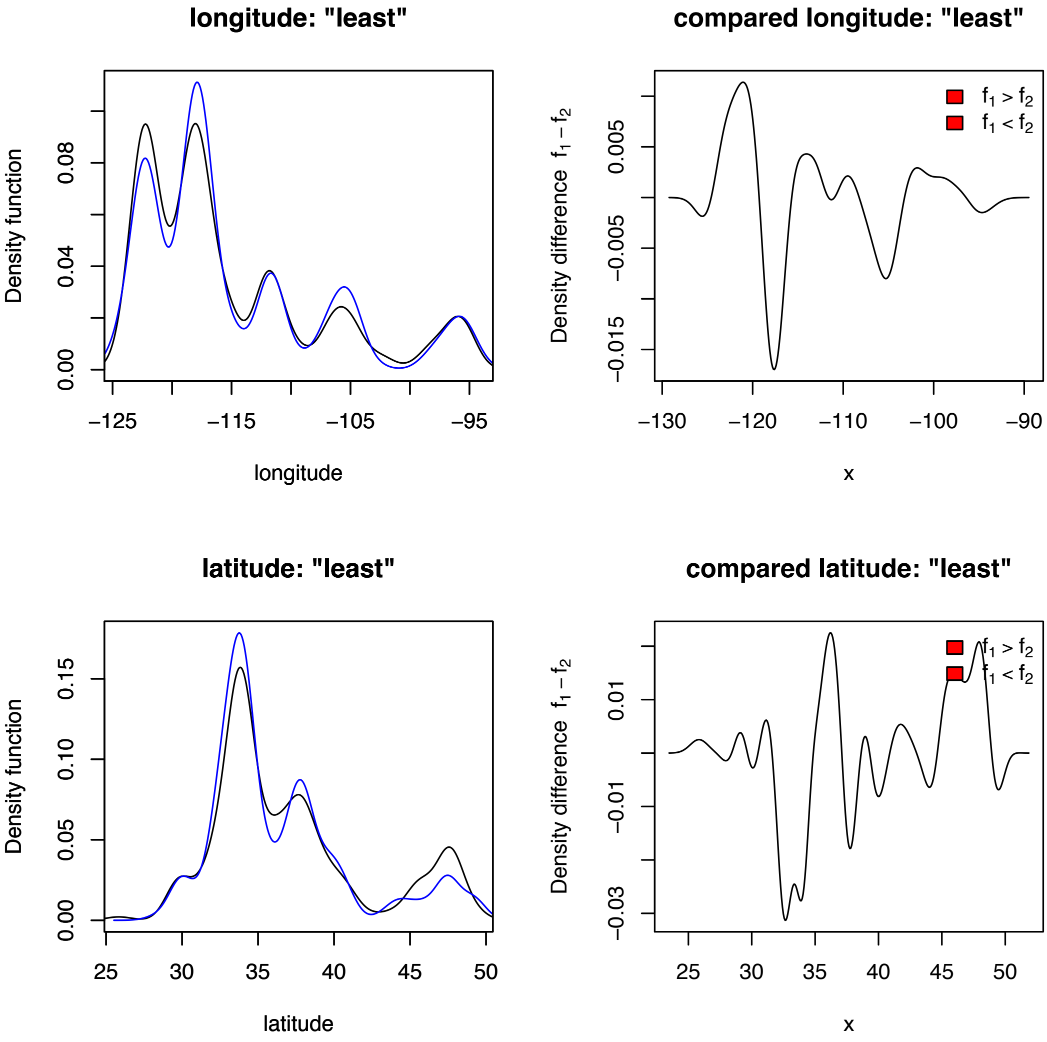


Figure 3.6: The leftmost graph contains two plots. Blue is new KDE and black is historic KDE. On the right side is a comparison of the two KDEs. f_1 is historic and f_2 is new. It can be observed when new is has higher value on the left graph it has a negative value in the compared graph.

tant details.

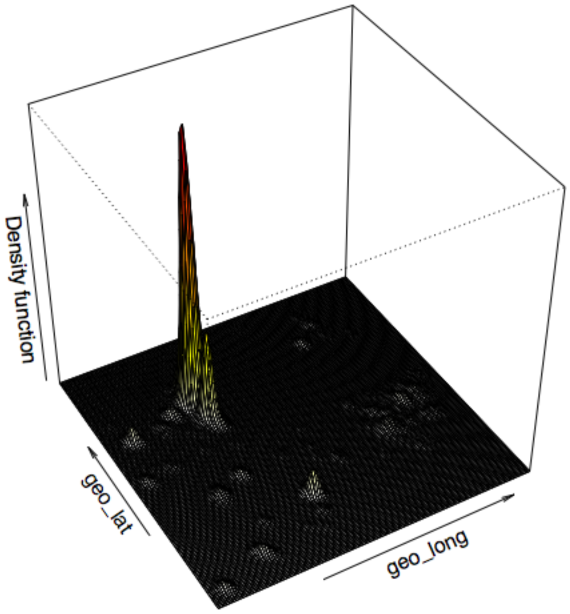


Figure 3.7: Bivariate KDE of multiple words in the state of Colorado.

As with the temporal method a lower boundary for the number of mentions has been set. Both new and historic must have more than 100 mentions for KDE and comparison to be calculated. It is set to make sure there is enough data to build a statistical model. This boundary is the same as $min\_mentions$ in the previous algorithm.

3.2.3 Clustering Words with Odds Ratio

The three detection methods produce an excessive amount of words which potentially are connected to events. The goal of applying odds ratio (OR) is to reduce the number of words and at the same time cluster the words together. Read Section 2.3 if you are unfamiliar with OR.

The developed algorithm calculate OR between $word_x \in all\_detected\_words$ and $word_y \in all\_detected\_words$.

First the two words $word_i$ and $word_j$ are added to a list and sorted alphabetically. Because the calculated OR is accessed later it is important that it can be accessed in a consistent way. $word1$ and $word2$ are encapsulation objects containing a string representation of the word and all the detected features. These can be positions, timestamps, compared percentage etc. The counting is then performed.

```
for $word_i$ in $all\_detected\_words\_dict$ do
    for $word_j$ in $all\_detected\_words\_dict$ do
        $sorted\_words \leftarrow sort(\ [word_i, word_j]\ )$
        $word1\ \leftarrow all\_detected\_words\_dict[sorted\_words[0]]$
        $word2\ \leftarrow all\_detected\_words\_dict[sorted\_words[1]]$
        $both\_count\ \leftarrow count\_both\_occurences(word1, word2, tweets)$
        $word1\_count\ \leftarrow count\_word\_occurences(word1, tweets)$
        $word2\_count\ \leftarrow count\_word\_occurences(word2, tweets)$
        $no\_word\_count\ \leftarrow count\_no\_words(word1, word2, tweets)$
        $OR\ \leftarrow$
        $calculate\_OR(both\_count, word1\_count, word2\_count, no\_word\_count)$
        if $OR > minimum\_OR$ then
            $add\_OR\_to\_dict(OR, word1, word2)$
        end
    end
end
```

Algorithm 3: Calculate odds ratio.

For better visualization the four counts are performed in succession, but in reality they are performed at the same time. Counting is done by looping through all the parsed tweets to see if $word1$ and / or $word2$ are present. OR is then calculated and if it is larger than $minimum\_OR$ it is added to a dictionary where the key is $sorted\_words$. $minimum\_OR = 10$ has been empirically found.

The following table visualizes how the counting when calculating the OR between $word1$ and $word2$ is performed.

	$word1 = 1$	$word1 = 0$
$word2 = 1$	$p11$	$p10$
$word2 = 0$	$p01$	$p00$

$p11$ is all tweets containing both $word1$ and $word2$. $p10$ is all tweet containing $word2$, but not $word1$. $p01$ is all tweets containing $word1$, but not $word2$. $p00$ is all tweets where neither $word1$ or $word2$ is present. These four variables are then used in Equation 2.7 to calculate OR.

The dictionary of calculated odds ratios is used by Grapher to create one or multiple undirected graphs.

3.2.4 The Bivariate Spatial Method - Bivariate Spatial Comparison of Kernel Density Estimates for Each Graph

The goal of this approach is to find geographical regions with a higher level of activity in a selected graph and then visualize the findings on a map. An example of this method is illustrated in Figure 3.8. The geographical area is western USA.

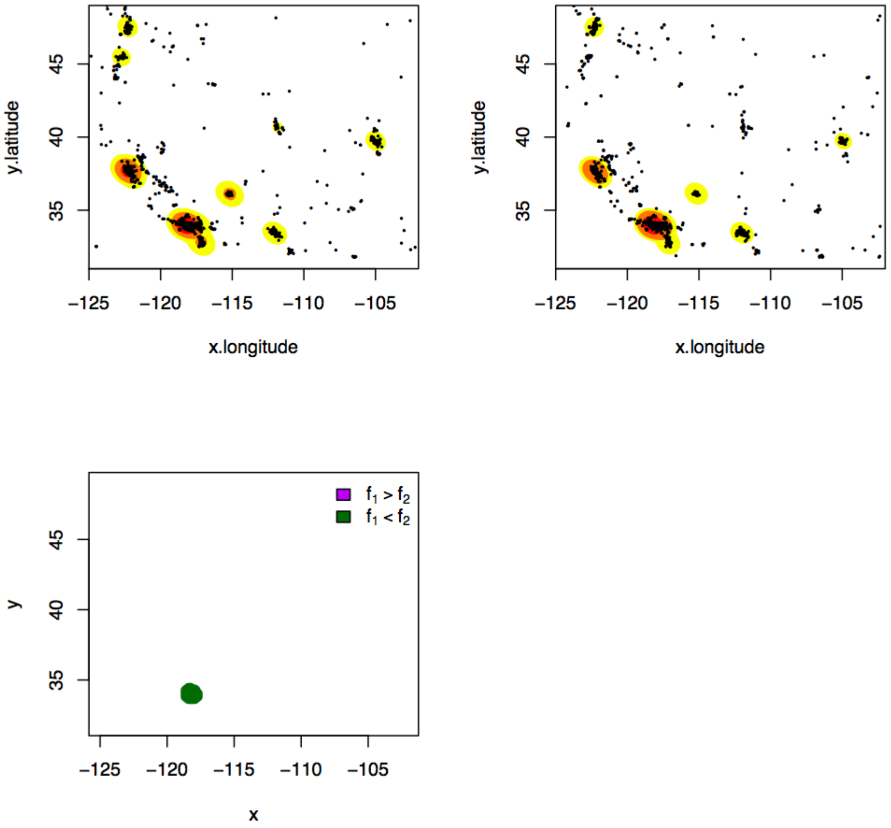


Figure 3.8: The upper left graph f_1 is a kde based on historical data. The upper right graph f_2 is a kde based on new data. The lower graph is the difference between the two graphs. In the green area f_2 is significantly larger.

This approach has many similarities to *Spatial Comparison of Kernel Density Estimates for Each Word* described in Section 3.2.2. Both does a comparison of historic data and new data in a spatial context. But where the spatial method 3.2.2 uses univariate data this approach applies bivariate comparison. The type of data is also different.

The historic data model and the new data model is constructed from graph data produced by the clustering algorithm described in 3.2.3. The clustering data consists of one or in most cases multiple graphs. A graph consists of a list of two words sorted alphabetically, the calculated OR and other properties. An algorithm is applied to retrieve all the words in each graph.

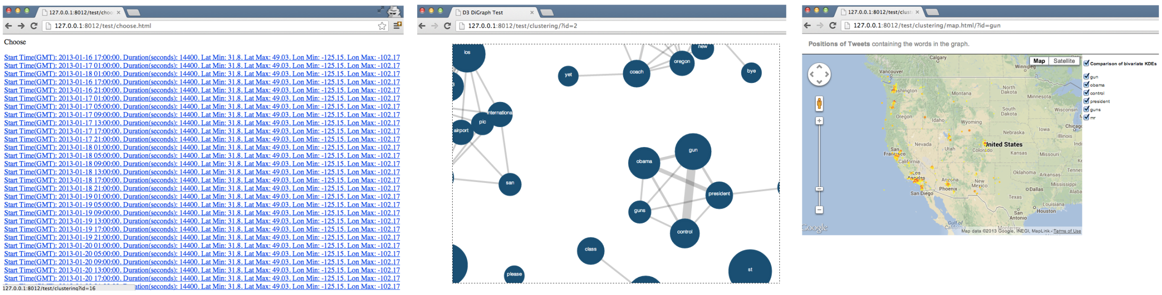
```
$graphs \leftarrow$ get_graphs($id$)
for $graph$ in $graphs$ do
    $words\_sets \leftarrow$ get_words_from_graph($graph$)
    for $words$ in $words\_sets$ do
        $positions_{historic} \leftarrow$ get_coordinates($words, dict_{historic}$)
        $positions_{new} \leftarrow$ get_coordinates($words, dict_{new}$)
        sample_if_too_many($positions_{historic}, positions_{new}$)
        $diff\_positions \leftarrow$
        bivariate_kde_diff($positions_{historic}, positions_{new}$)
        store_diff_positions($diff\_positions, words, id$)
    end
end
```

Algorithm 4: Calculate difference between two bivariate kdes.

The algorithm first retrieves all the graphs with a db id. Then it gets sets of words contained within the graph. $dict_{historic}$ and $dict_{new}$ contains Feature objects of all the words. A Feature object is for one word and contains all the geographical positions where this word originated from including other properties. The Feature object for the word "gun" could for example have positions from Boston, Springfield, Miami etc. All the positions from one set is combined to a list. This results in the instantiation of $positions_{historic}$ and $positions_{new}$. Because calculating the difference between bivariate kdes is computational expensive the number of positions have to be reduced. Accomplishing this is done by creating a sample of a couple of thousand positions. The sample is created by shuffling the list of positions and then slice it down to a couple of thousand positions. With two appropriately sized lists the R library ks can compute the difference between them. Finally the areas which have an increase compared to the historic data is stored in the db.

Because of the cumbersome name of the method it is just abbreviated to the bivariate spatial method.

3.3 Grapher



Choose

Start Time(GMT): 2013-01-16 17:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-17 01:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-18 01:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-16 17:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-16 21:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-17 01:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-17 05:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-17 09:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-17 13:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-17 17:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-17 21:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-18 01:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-18 05:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-18 09:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-18 13:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-18 17:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-18 21:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-19 01:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-19 05:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-19 09:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-19 13:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-19 17:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-19 21:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-20 01:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-20 05:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-20 09:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-20 13:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
Start Time(GMT): 2013-01-20 17:00:00. Duration(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
ation(seconds): 14400. Lat Min: 31.8. Lat Max: 49.03. Lon Min: -125.15. Lon Max: -102.17
127.0.0.1:8012/test/clustering?id=16

Positions of Tweets containing the words in the graph.

☑ Comparison of bivariate KDEs
☑ gun
☑ obama
☑ control
☑ president
☑ guns
☑ mr

Figure 3.9: Illustrating the three different pages in Grapher. The leftmost screenshot is the Overview page. The middle screenshot illustrates the events in graphs. The right screenshot shows the map page.

Grapher is a web site used for visualizing the results produced by the detection pipeline. It consist of three different sites *overview*, *graph* and *map* illustrated in Figure 3.9. The *overview* site lists up all the performed tests. Clicking on any of the listed tests takes you to the *graph* page. Here the calculated OR is visualized as one or more undirected graphs. Clicking on any of the nodes takes you to the *map* page. Here the positions of all the words in the clicked graph are visualized. There are many more features in the map page, but those will be explained in one of the following sub sections. The overview page and the graph page will also be explained in their own sections.

Visualizing the results from the event detector is an important task. It is possible to detect events using the text files and graph pdfs produced by the event detector, but this is a cumbersome and time consuming process. Grapher makes it easy to highlight important events and do a deeper study of them by using the functionality of the *map* page.

To create the sites a Python web framework called Django has been used.

To use Grapher follow the instructions in the appendix.

3.3.1 Overview Page

The *overview* page retrieves all the results produced by the detection pipeline and visualizes them in a simple manner. The description of the test uses the spatial and temporal properties of the performed test. More precisely Δt and the geographical bounding box where the tweets used in the test originated from is displayed. Clicking on any of the tests brings you to the *graph* page.

3.3.2 Graph Page

This site visualizes the results from the OR calculations as undirected graphs. A JavaScript library named d3 is used as the component to visualize the graph. The results from OR is parsed to JSON as nodes and edges.

In theory the detection pipeline could discover a huge number of events based on thousands of words. This could be overwhelming for the *graph* page. To avoid an overload on the page only those edges having the highest OR value would be included. This limit is set to 500 for the moment, but could be deemed to high. 500 nodes on a screen including edges would be bewildering for the analyst trying comprehend what kind of events the graphs contain.

An example of a graph is visualized in Figure 3.10. It demonstrates how node size and edge width can vary. The graph need to have more than two nodes for this to happen. These different graphical properties conveys how the different words are related to each other. A thicker edge between two nodes in graph states that the OR is higher than those which have thinner edge widths.

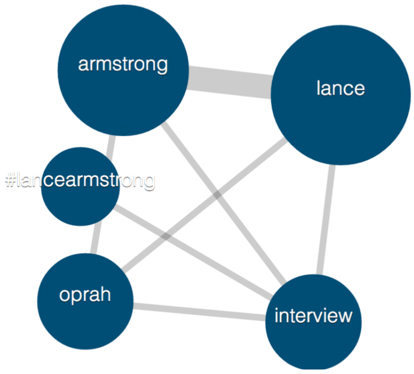


Figure 3.10: Graph illustrating varying node sizes and edge widths.

The size of the nodes is based on how many times the word is mentioned in tweets which the graph is based on. In this example ”armstrong” and ”lance” are mentioned approximately the same number of times. While the other words are mentioned less times. Node size is calculated with the following two equations for word w.

$$m = \frac{max\_size - min\_size}{max\_word\_count - min\_word\_count} \tag{3.4}$$

$$node\_size(w) = (m * word\_count(w)) + min\_size - (m * min\_count) \tag{3.5}$$

max_size is the maximum size a node can have and opposite is min_size the minimum size a node can have. max_word_count is based on a dictionary where the words in the graph are paired up with a word count. The word count is how many times a word is mentioned in the tweets which the graph is based upon. These tweets must have one or more of the words in the graph. max_word_count is therefore whatever words have most mentions. min_word_count is the same only opposite. $word\_count(w)$ is the number of mentions for word w. This results in $node\_size(w)$ for word w.

The edge weight is calculated in the same manner as node size, but uses OR instead of $word\_count$.

Clicking on any of the nodes takes you to the *map* page.

3.3.3 Map Page

The Google map illustrates all the positions of the tweets having one or more of the words in the graph. These positions are color graded dots together with an according size. The smallest dot is yellow. This represents a tweet containing only one of the words in the graph. The largest dot is red. This dot represents a tweet containing five of the words in the graph. Between these two types are dots representing two words, three words and four words. More words means larger dot size and more reddish coloring up to five words. See Figure 3.11 for an example.

Another feature provides the ability to deselect and select dots corresponding only to the word selected. If a word is deselected all the tweets which only has the deselected word and none of the other words in the graph will be removed from the map. If there are words which are not interesting, the positions of these can be removed. The words and their corresponding positions can be thought of as a series expressions combined with the logical operator *or*. The words are sorted in descending order by $word\_count$. See the previous subsection for more details.

Results from comparison of bivariate KDEs are visualized on the geographical map as a heat map. It is only visible when the new bivariate KDE is significantly

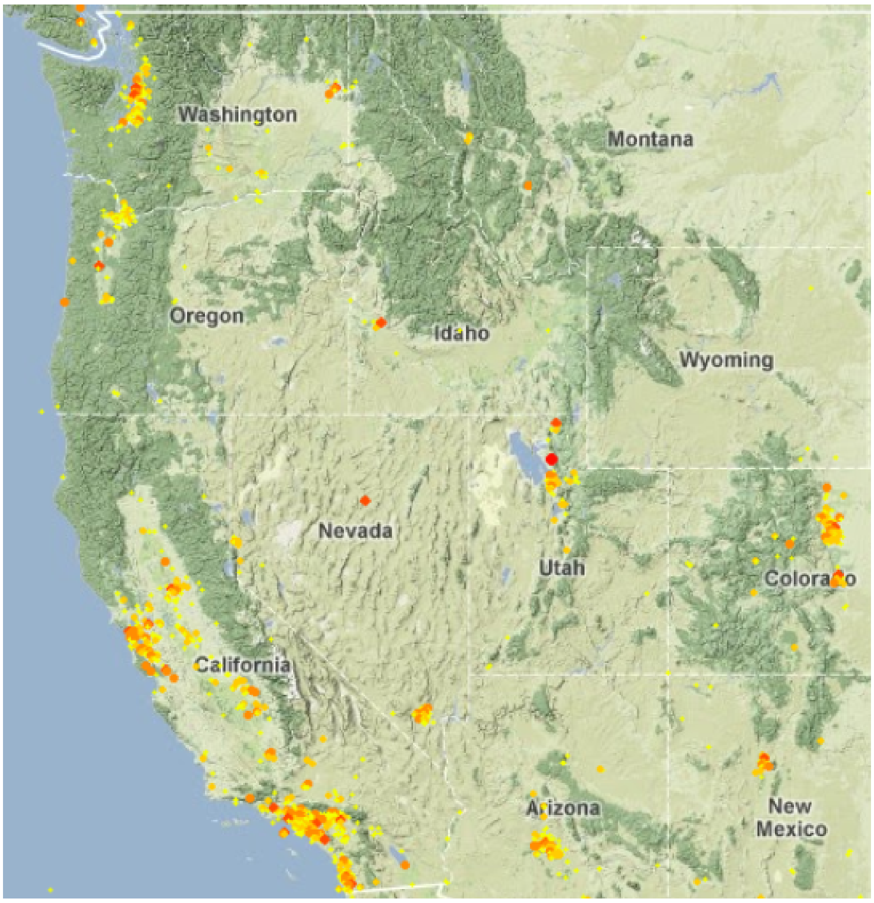


Figure 3.11: Map of dots representing tweet positions located in western USA.

larger in some area than the historic bivariate KDE. The possibility of deselecting and selecting this is present.

The last feature enables the user to click on any of the dots and display the entire tweet together with the user name.

Chapter 4

Discussion

4.1 Geotagged Tweets Compared to Non Geotagged Tweets

All tweets in this project are geotagged. This section compares geotagged tweets with non geotagged tweets to highlight and discuss some of the differences. To accomplish this tweets from Japan have been retrieved.

One of the criteria for choosing Japanese is that it is possible to retrieve tweets from Japan without specifying the tweet should contain a geotag from within Japan. The Japanese language is uncommon outside Japan. English on the other hand is the primary language in many countries and is commonly used in other countries as well. The same goes for Spanish, German, French and many others. It is therefore a reasonable assumption that most tweets written in Japanese originate from within Japan. Another advantage with Japan is that it is an archipelago (island group). This makes it is easy to cover with boundary boxes when retrieving geotagged tweets.

The goal is to compare geotagged tweets from Japan with tweets from Japan which are not geotagged. The time frame is the same for both data sets. Retrieving tweets written in Japanese is necessary. Because the Twitter streaming API does not allow to query only by language an additional parameter have been specified. The 30 most commonly used Japanese words are therefore specified as the extra parameter. These words are also queried for when getting the geotagged tweets to ensure consistency. The geotagged tweets are all in Japanese because they are retrieved from the same time frame.

The duration of the retrieved tweets is approximately 24 hours. Both data sets were retrieved simultaneously to make sure the vocabulary is as consistent as possible. 105 000 tweets without geotag was retrieved. 125 000 tweets with geotag was retrieved. It is interesting that more tweets are geotagged than those without. This can be due to a failure in the retrieval process. This has been investigated and no faults were found. Twitter uses an algorithm to assigns language by analyzing the 140 character long text. As long as Twitter's language classification is working this can indicate Japanese Twitter users for the most part geotag their tweets.

To highlight the differences the WDC method have been used. The spatial method is impossible to use because one data set is missing geotags. The temporal method could have been used, but to keep this comparison simple and straightforward it has been dropped. To further simplify only the 220 most commonly used Japanese words are compared. This should give an indication of how different the non geotagged tweets are compared to geotagged tweets. The tweets without geotags have taken the role as historic data while the tweets with geotags have taken the role as the new data.

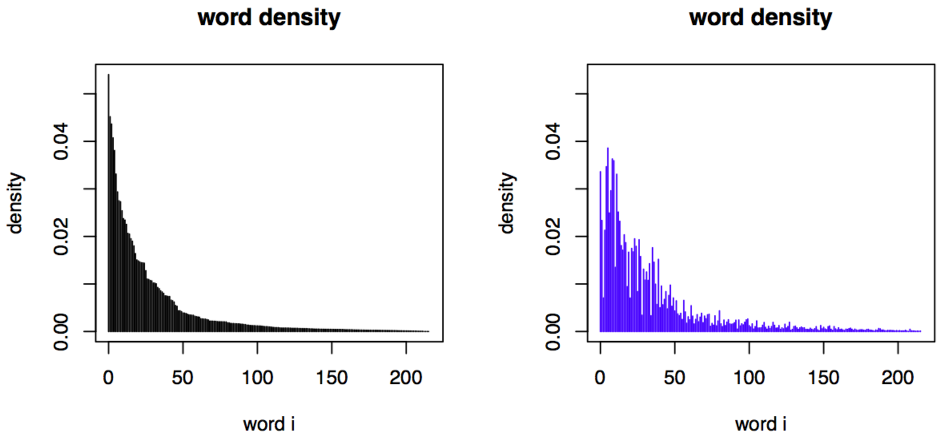


Figure 4.1: The left graph is the word density of the non geotagged tweets. The right graph is the word density of geotagged tweets. They are sorted in the same order. It is observed there are significant differences between the two data sets.

Figure 4.1 illustrates the result from WDC. Although they have approximately the same two word densities there are some noticeable differences. Ideally the density on the right should have had exactly the same shape as the one on the left. This would have strongly indicated that geotagged tweets and tweets lacking geotags are using the same vocabulary.

The differences between the two densities are greater than most tests performed on the solution. There could be many reasons for this. One reason could simply be that the data consists of too few tweets and is therefore not representative. But if the data is representative this indicates that geotagged tweets contains different content than non geotagged tweets. At least it strongly indicates that the 220 most commonly used Japanese words are used differently in geotagged tweets and non geotagged tweets. A possibility is that those who geotag belong to a different demographic group than those who do not geotag. If only a specific demographic group is tweeting with geotags then it might also be possible that some events and crisis situations are unable to be detected by the system. Before drawing any conclusions a more comprehensive study should be performed.

4.2 Temporal Analysis of Geotagged Tweets

During this project more than 90 million geotagged tweets have been retrieved. This subsection presents an analysis of the number of geotagged tweets posted between 2013-02-22 and 2013-04-02 on the eastern seaboard of the US. From Florida in the south to Vermont in the north. The western boundary is located by the borders of Missouri. The reason for choosing USA is because it is large both in population and geographically. According to a survey from 2012 15 % of the American which is online has a Twitter account and 8 % use it on a daily basis [4].

The total number of tweets exceeds 50 million for this duration and geographical area. On average well over a million tweets are posted each day. Figure 4.2 shows the number of geotagged tweets is increasing.

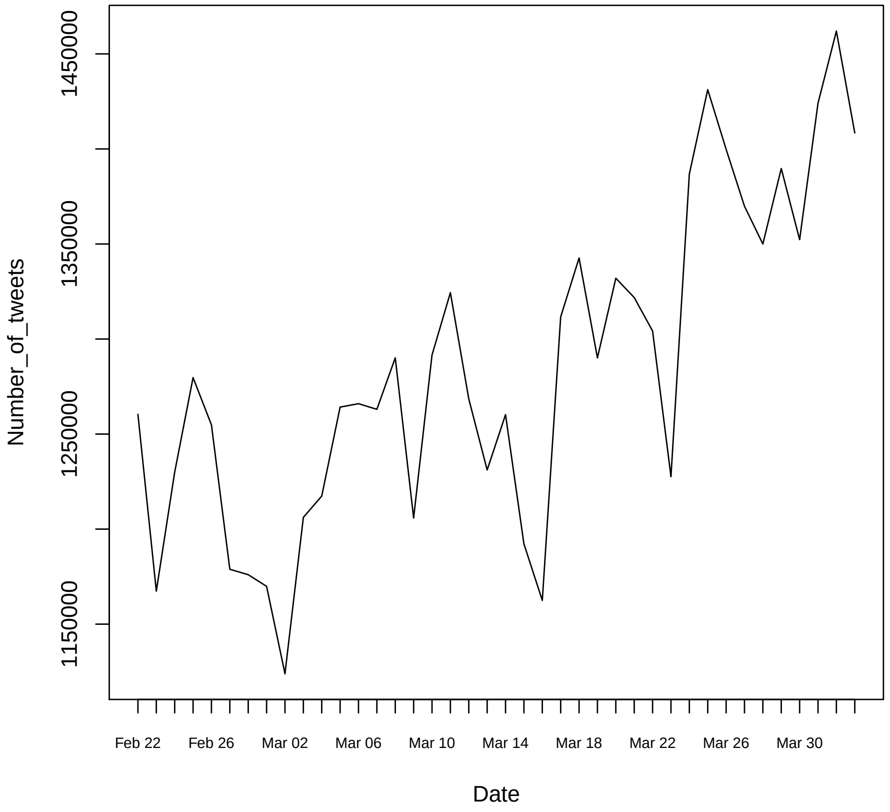


Figure 4.2: Increasing number of tweets in March.

The increase in tweets can be due to a variety of reasons. For example are more people becoming Twitter users [23]. As a result the number of tweets could increase. Another explanation could be that existing Twitter users might become more comfortable with the medium and as a result post more tweets. A third theory could be that the number of tweets is pretty stable but more users tweet from their GPS capable cell phones using geotagging.

In recent years smart phone sales have skyrocketed. A smart phone provides everything a Twitter user need to tweet a geotagged text. The number of capable users increase and as a possible result, more geotagged tweets are produced.

	Mon	Tues	Wed	Thu	Fri	Sat	Sun
avg # of tweets	1342889	1314312	1275564	1274218	1267764	1206589	1308378
Std Dev	91737.7	70572.8	76645.28	67032.4	80058.1	79991.2	85359.9

Table 4.1: Average Number of tweets per weekday and their standard deviations.

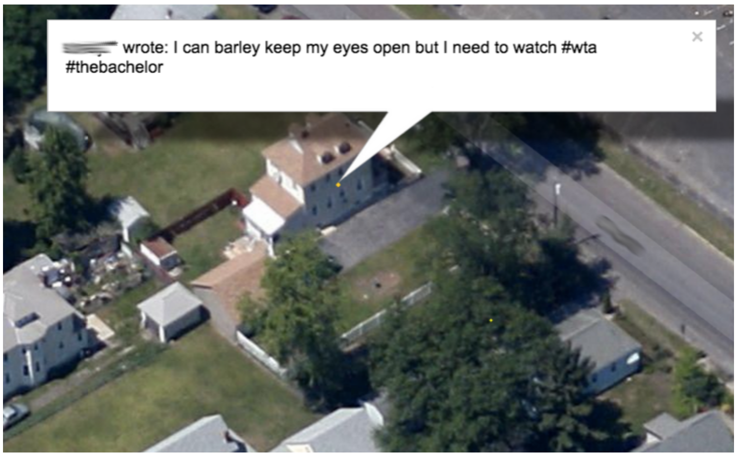
Table 4.25 can help explain why graph 4.2 fluctuates on a weekly basis. The biggest dips in the graph are mostly Saturdays where people seem to tweet less. Sunday, Monday and Tuesday are days where Twitter activity increases. From Thursday to Saturday the number of tweets declines. Tweet Retriever is sometimes unable to retrieve tweets either because of faulty Internet connection or something else. But it is quite persistent and it should have minimal influence on these numbers. A study [26] done by Sysomos, a company providing social media data and analysis reached the same conclusion about how we tweet during the weekdays.

It is likely people will geotag their tweets to a greater extent in the future. At least as long as they are tweeting. People get more gadgets and seem to use it without much concern to the amount of information they make public. This is not a new trend. It has just become much easier for the average Joe to do it. In the late 1990 advanced Internet users created homepages about themselves containing loads of private information. This might be considered the predecessor of Facebook, Twitter and the other social media platforms. This indicates that more people learning to use advanced technology results in more private information made public.

4.3 Privacy Issues

Collecting more than 90 million statements from people and knowledge of their location is a potential minefield when it comes to privacy issues. The proposed detection methods presented in this thesis do not utilize any user information. It can therefore be considered as anonymous information. The detection methods look at the tweets as strings containing temporal and spatial properties. The functionality in Grapher showing the tweet text, who wrote it and the location is the only part of the solution which is problematic in relation to privacy.

Figure 4.3 illustrates how Grapher can show a tweet text, the Twitter user (blanked out) and the location. Google Maps providing a clear $45\,^{\circ}$ angle picture of the



[redacted] wrote: I can barley keep my eyes open but I need to watch #wta #thebachelor

Figure 4.3: Privacy issue?

house in combination with the tweet text is powerful and might even be considered frightening.

The text states that the person likes to watch The Bachelor. It is also likely the person is living in the house since he / she is about to go to sleep. Combining the Twitter user name with the names registered on the address of the house could map the Twitter user to an actual person.

Although this is not entirely in accordance with the Twitter's Rules of the Road there is nothing stopping people to use this for internal use. Utilizing this kind of data in targeted advertisement could be possible. Map the daily routine of Twitter users together with what they are tweeting about. It might even be possible to differentiate areas on what TV shows or cell phone brand the inhabitants like. The possibilities for abuse are virtually endless. Fortunately utilizing user information like user name is not necessary to do crisis analysis on tweets.

4.4 Results

This section discusses the results produced by the solution pipeline. More than 300 tests have been performed on the solution pipeline. A test in this context is performed with respect to spatial and temporal restrictions. All the performed tests are found on Grapher. How to connect to Grapher is found in the appendix. There have also been produced hundreds of thousands of graphs describing the comparisons of the kernel density estimates for the spatial detection methods and the temporal detection method. Because of the vast amount of results it is impossible to present all of them in this thesis. A small, but representative fraction of the results is discussed in the following two subsections.

Deciding if some words are related to an event is a difficult task. An event is therefore defined as something being typical for a limited time and something which does not occur often. It should have some news value. A definition for event provided by Webster dictionary: *the fundamental entity of observed physical reality represented by a point designated by three coordinates of place and one of time in the space-time continuum postulated by the theory of relativity*.

There are multiple approaches to measure the performance of the solution. One approach is to use precision and recall. It requires to know the number of event the solution is supposed to find for each time frame. This task is close to impossible. It would require to know all major events affecting a certain geographical area and during a certain time span. It is difficult to determine if something is an event. A poor approach would be to define events as something newspapers post. Although this is true about many major events. Many events do not even appear in main stream newspapers.

To quantify the performance of the solution an event related words ratio is calculated with

$$\frac{n_{event}}{(n_{event} + n_{noise})} \tag{4.1}$$

where n_{event} is the number of words related to an event. n_{noise} is the number of words related to noise. The performance is quantified at the end of each test.

Noise are words which could not be classified as an event.

The events presented in the following subsections do not necessarily correspond to a single graph, although they do in most cases. Sometimes an event can also be described by just a few nodes in a graph.

Grapher has been used for visualization of the test results complemented by some graphs created by the three detection methods. The events have been verified using historic (news) search.

4.4.1 General Event Detection

The following subsections discuss 10 randomly selected test results. These are representative for the 322 performed tests. The 322 tests are divided into the following two geographical areas and temporal frames:

1. **Western USA** - geographical area stretches from the coast of western USA to Colorado in the East, from New Mexico in the south to the borders of Canada in the north. The time frame for these 67 tests are 2013-01-16 to 2013-01-27. At least one test have been performed per day.
2. **Eastern USA** - geographical area stretches from the coast of eastern USA to Missouri in the West, from Florida in the south to Toronto in the north. The time frame for these 244 tests are 2013-02-23 to 2013-04-01. There is at least one test for almost every day.

The time frames from the tests are from all hours of the day. The discussions will not explain why all the words have been detected, but focus on a few. The words to be explained are chosen to represent different aspects of the detection methods.

January 21 17:00 - 21:00 GMT - Western USA

Words	Event
martin, luther, king, jr, day, jr, dr, mlk, #mlk, holiday, working, cannot	Dr. Martin Luther Kings day. Official holiday.
president, obama, obama's, inauguration, america, american, #inaug2013, #inauguration, inaugural, our, music, listen, video, kelly, clarkson, amazing, national, country, beyonce, singing, speech, gay, watching, until, complete, brother, thank, god	Inauguration of president Obama. Gave a speech about gay rights. Kelly Clarkson sang ”My Country, 'Tis of Thee”. Beyonce sang ”Star-Spangled Banner”.
white, black, house.	Loosely connected to the inauguration. Black man in the white house.

Table 4.2: Real events detected January 21 17:00 - 21:00 GMT

WDC, the spatial detection method and the temporal method is used to detect ”president”, ”obama” and ”king”. These are commonly used words. Meaning they are widely used throughout the last year. Other words are less common both in a spatial and temporal context.

The words ”luther”, ”mlk”, ”martin”, ”inauguration”, ”#inauguration” and ”#inaug2013” are unusual words. They are rarely used and makes the construction of the historical model utilized by the spatial method and the temporal method difficult. People do not usually tweet about inauguration or Martin Luther. These are therefore only detected by WDC which necessarily do not need historical data. The spatial method and the temporal method constructs historical models and when there is too little data these methods can not be used. Comparing a new model with a vague historical model can give bad results. The ratio between the detection methods are illustrated in Figure 4.4.

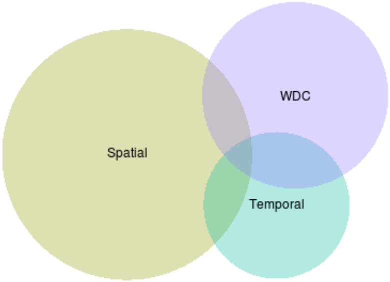


Figure 4.4: Ratio between the three detection methods, spatial method, temporal method, and WDC.

Only a small portion of the detected words are detected by all three methods. Words detected by all three methods are more probable to be real events than those detected by fewer methods. The three detection methods each cover different feature domains. Because of this if a word is detected by only one method it does not necessarily mean it is not connected to an event. Uniformly distributed events are less likely to be discovered by the spatial detection method. But it does not mean the event is off less importance. The methods complement each other. If one method does not detect an event it is a good chance one of the other two does. The three methods suggest many words, some of these can be considered as noise.

Noise, in this thesis also called natural fluctuations occur in most of the performed tests. Word usage fluctuates and determining the cause can be a tricky task. Words considered to be noise are in this project defined to be all words which are not related to any specific event.

Noise detected in this test are described in Table 4.3.

Words	Explanation
please, follow	Strongly connected words. Natural fluctuations.
city, park, studio, resort	Strongly connected words. Natural fluctuations.
airport, international, san	Strongly connected words. Natural fluctuations.
sweet, home	Strongly connected words. Natural fluctuations.
california, 21	The day of month is 21.

Table 4.3: Natural fluctuations January 21 17:00 - 21:00 GMT

Event related words ratio = **0.772**.

January 23 09:00 - 13:00 GMT - Western USA

No events or noise were detected. This is probably due to the time frame. Local time frame would be 02:00 - 06:00. This is usually when most people are asleep and thus little Twitter activity.

Event related words ratio = **0.0**.

February 25 01:00 - 03:30 GMT - Eastern USA

This time frame is during the Oscars. The results are extensive. The three first tables are Oscar related events. While the next two are general events and noise. In Grapher the Oscar related nodes are for the most part represented as one huge graph. At first sight it can be intimidating. Looking more closely at the nodes and how they are connected provides a great overview of the event.

Words	Event
#oscars", #oscar2013, #academyawards, show, academy, award	Academy Movie Awards.
seth, mcfarlane, mcfarland,	Seth Macfarlane hosted the Oscars.
renee	Renee Zellweger had taken too much botox.
best, documentary, sugar	The movie "Searching for Sugar Man" won best documentary.
paperman, won, animated, short, film, yay, ralph, brave, deserved, nominee, nominated	"Paperman" won best animated short film.
django, unchained, movie, racist, ever, seen, christoph, role, christopher, waltz, well, deserved, congrats, winning, supporting	The movice "Django Unchained" was nominated in multiple categories. It is considered by many to be racist. Christoph Waltz won best supporting actor and best writing.
amour	Amour won best foreign film.
#lesmiserables, #lesmis, miserables, cast, performance, category, winner, makeup, ann, anne, hathaway, hathaway's, actress, nipple, dress, speech, cut, hair, princess, #annehathaway, deserved, eddie, aaron	Les Miserables won 3 Oscars. Best Makeup and Hairstyling. Best Achievement in Sound Mixing. Best Performance by an Actress in a Supporting Role was Anne Hathaway.
pie, pi, life, tiger, cinematography, effect	Life of Pi won best cinematography and best visual effects.
melissa, mccarthy, paul, rudd, beard, rihanna, chris, brown, joke	Seth Macfarlane made jokes about Melissa Mccarthy, Paul Rudd, Rihanna and Chris Brown. Paul Rudd had a beard.
robert, downey, jr	Not going to star in another Iron Man movie.
jamie, fox, foxx, daughter, kelly, daughter, gorgeous, via	Jamie Foxx Hits On Kelly Rowland In Front Of His Daughter.
silver, lincoln, argo	Nate Silver predicted Argo and Lincoln would win.

Table 4.4: Oscar events part 1 / 3. February 25 01:00 - 03:30 GMT - Eastern USA.

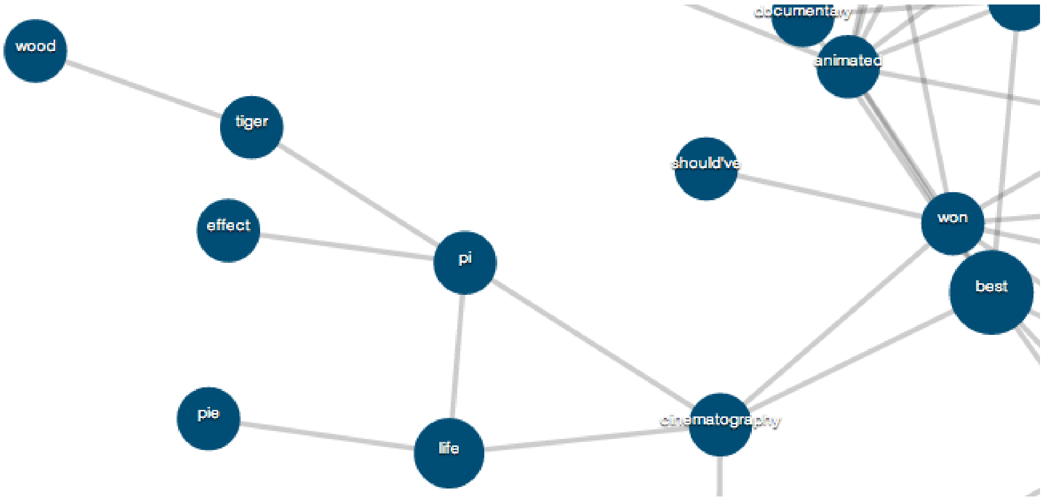


Figure 4.5: Part of the Oscars graph describing the movie Life of Pi and its winnings. See Table 4.4 for details about this event.

Figure 4.5 illustrates how an event can be found by utilizing Grapher. The name of the movie is Life of Pie. The nodes ”pi” and ”pie” (misspelled) are both connected to life. The word ”of” is removed in the beginning of the detection pipeline by the text parser. A tiger has a big part in the movie and thus connected to ”pi”. The odds ratio have not been high enough for ”tiger” to connect to ”life”. ”life” is a commonly used word and the two words have probably not been mentioned together enough times. ”wood” is lemmatized from ”woods” and since it is connected to ”tiger” it is probably Tiger Woods. The node ”wood” therefore has no relation to the movie in this context. ”effect” is connected to ”pi” and not to ”life” because ”effect” and ”life” are very common words. ”cinematography” is an unusual word and is therefore connected to both ”life” and ”pi”. Life of Pi won best cinematography.

Words	Event
daniel, day, lewis, lincoln, actor, tommy, lee, *jones*, supporting	Daniel Day-Lewis won best actor for his role in Lincoln. Tommy Lee Jones nominated for best supporting actor.
catherina, zeta, catherine, jones, lip, jazz, chicago	Catherina Zeta Jones was lip synching while singing ”All That Jazz” from her movie Chicago.
daniel, radcliffe, harry, potter, dancing, star, joseph, gordon, levitt.	Daniel Radcliffe and Joseph Gordon-Levitt was singing.
resse, whiterspoon, blonde, hair,	Reese Witherspoon’s Oscar Dress.
#redcarpet, #bestdressed, dress, sandra, bullock, halle, berry, pussy, shoulder, arm, jennifer, jen, anniston, aniston, garner, hudson, charlize, theron, stunning, reese, witherspoon, jessica, channing, tatum, kerry, washington, dance, amy, adam	Sandra Bullock, Halle Berry, Jennifer Anniston, Charlize Theron, Kerry Washington, Amy Adams, Reese Witherspoon, Amy Adams, Channing Tatum had nice dresses.
ben, affleck, argo, beard	Ben Affleck had a beard and stared in Argo.
george, clooney, beard, first	George Clooney had a beard.
russel, russell, crowe, singing, sing	Russel Crowe was singing.
hugh, jackman, sing	Hugh Jackman was singing.
james, bond, 50, year, skyfall, adele	50 years of James Bond. Adele’s song Skyfall for the movie with the same name won best music. Won best sound editing as well.
captain, james, kirk	William Shatner played captain James T. Kirk in a sketch.
seth, macfarlane, mcfarlane, macfarland, hosting, host, peter, hilarious, puppet, sock, flight	Seth Macfarlane hosted the Oscars and made and had a sketch sock puppets on a flight.
kristin, kristen, stewart, queen, interview, awkward, red, carpet, chenoweth	Kristen Stewart and Kristin Chenoweth co-hosted the Oscars.
samuel, samual, jackson, red	Samual Jackson had a red outfit.
mark, ted, bear	Mark Whalberg presented an award with the bear Ted.
beasts, southern, wild, beauty	Beasts of the Southern Wild nominated for four Oscars.

Table 4.5: Oscar events part 2 / 3. February 25 01:00 - 03:30 GMT - Eastern USA.

Words	Event
sally, field, nun	Sally Field was a flying nun in a sketch.
john, travolta	John Travolta's hair had grown.
liam, neeson	Liam Neeson was at the Oscars.
kilt, brave, wearing, pant	Seth MacFarlane's dad wore a kilt.
sound, music, reference	Sound of Music reference to introduce some actor.
theater, theatre, musical, tribute	
Oscars present tribute to musical theater featuring Chicago, Dreamgirls and Les Misérables. anna, design, costume	Anna Karenina wins Oscar for costume design.
possible, tie	Tie for Best Sound Editing.
avenger, cast	Cast of the Avenger at the Oscars.
boob, theme, song, saw, jaw, rude, play	Seth Macfarlane's boob song.
zero, dark, thirty	The move "Zero Dark Thirty" was believed to win an Oscar.
bradley, cooper, zoe	Bradley Cooper and Zoe Saldana was at an after party.
shirley, bassey	Shirley Bassey led James Bond anniversary.

Table 4.6: Oscar events part 3 / 3. February 25 01:00 - 03:30 GMT - Eastern USA.

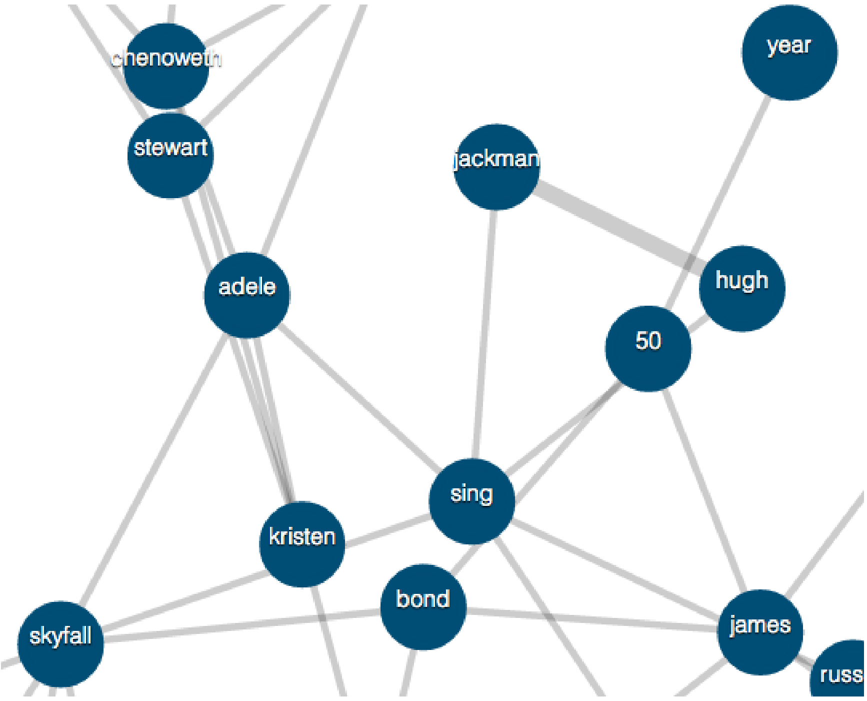


Figure 4.6: Part of the Oscars graph describing the James Bond 50th anniversary and Adele who wrote the song Skyfall.

From the graph in Figure 4.6 it can be read that ”year” is connected to ”50”. ”james” and ”bond” are both connected to ”50”. ”year” is not connected to ”james” or ”bond”. Probably because the two words are too common. ”50” is an unusual string in this context. ”sing” is used for multiple events. It is connected to ”adele”, ”skyfall”, ”kristen”, ”hugh”, ”jackman”. It easy to comprehend Hugh Jackman sang, but Kristen Stewart did not sing. Seth Macfarlane sang about her. It it also obvious James Bond had its 50th anniversary. Adele won best music for the song Skyfall.

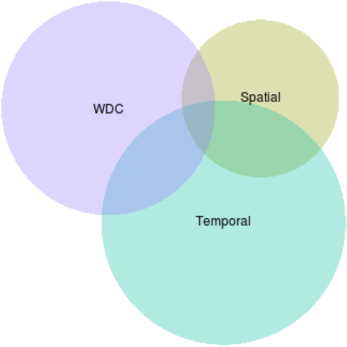


Figure 4.7: Ratio between the three detection methods, spatial method, temporal method, and WDC.

Figure 4.7 is a venn-diagram showing the usage composition of the detection methods. The spatial method is by far the smallest and thus underlining that the Oscars is an uniformly distributed event. Most of the events related to the Oscars have been detected by the by WDC and the temporal method.

Words	Event
#thewalkingdead, #walkingdead, walking, dead, andrea, governor, rick, #iwantyou, daryl, eye, amc, carol, carl	In the tv series ”The Walking Dead” the governor reveals what is under his eye-patch.
lion, king, abc, #lionking, disney	Lion King was aired on ABC
rain, thunder	Rain and thunder in southern Louisiana and Mississippi.
#blackhawks, hawk, center, win	Blackhawks wins over Edmonton Oilers at United Center. (NHL)
knicks	Knicks won over Philadelphia 76ers. (NBA)
james, lebron, wade, miami, heat, cavs, game	Miami Heat vs Cavs. Baskeball. James LeBron scored many points and Wade played well. (NBA)
pen, #pens, paul, martin, goal, net, center,	The Penguins Paul Martin played great vs Tampa Bay Lightning.They played on Consol Energy Center. (NHL)
snow, white	Snow White and the Huntsman.

Table 4.7: General events. February 25 01:00 - 03:30 GMT - Eastern USA.

The natural fluctuations from this time frame is found in Table 4.8. No event related to the actor Kevin Hart could be found, although he is probably related to the Oscars.

Table 4.8 lists up the noise. Many of the words are probably related to the Oscars, but cannot be associated with a real event.

Words	Explanation
beach, fl, lake	Natural fluctuations. Beach is in relation to Miami in the above event.
#relationshipwontworkif, taylor	Natural fluctuations.
gt, ”	Natural fluctuations.
math, test, tomorrow, school, homework, hw	Natural fluctuations.
black, twerk	Could snow somewhere, but mostly natural fluctuations.
kevin, hart	Could not find an event.
during, commercial	Might be related the Oscars, but difficult to find the connection.
happy, birthday	Natural fluctuations.
bull	Natural fluctuations.
wednesday, due, essay	Natural fluctuations.
should’ve	Natural fluctuations.
wood	Is connected to ”tiger”. But ”tiger” has to do with the movie Life of Pi.
east west	Natural fluctuations.
official, here, video, country	Could not relate to event.
prom	Could be related to Oscar dresses.
war	Natural fluctuations.

Table 4.8: Natural fluctuations February 25 01:00 - 03:30 GMT - Eastern USA.

Event related words ratio = **0.873**.

March 4 15:30 - 17:00 GMT - Eastern USA

Words	Event
tornado, phone	People in Milwaukee, Wisconsin got false tornado warnings to their phones.
wednesday, snow	A coming snowstorm.
spring, break	Spring break.

Table 4.9: Real events detected March 4 15:30 - 17:00 GMT - Eastern USA.

The first event presented in Table 4.9 was people in Milwaukee, Wisconsin who got false tornado warnings on their phones. The event originates in Milwaukee and

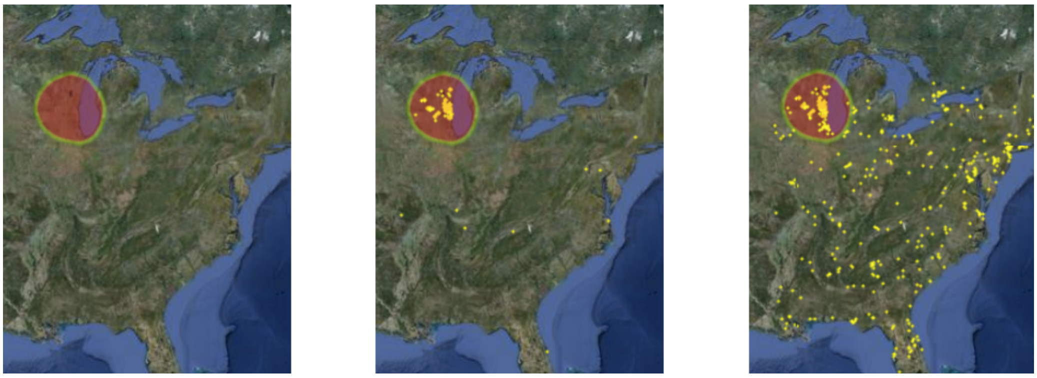


Figure 4.8: The Bivariate spatial method show there is an increase in tweets containing the words ”tornado” and ”phone” in Wisconsin (the red area). The map in the center contains the positions of all tweets using the word ”tornado”. The map on the right contains all tweets having the words ”tornado” or ”phone”.

affects the Twitter users in this area much more than Twitter users in for example Florida and can therefore be considered a Gaussian event. Figure 4.8 illustrates this in all three maps with a red area using the bivariate spatial method. Where all the words in the graph are related to the event the bivariate spatial method seem to work.

The first three rows in Table 4.10 are typical noise. ”math”, ”test”, ”teach” are also typical noise. Students are having math tests regularly. Math is for many a difficult subject and it is conceivable many students are not looking forward to these kind of tests and therefore wants to share their frustration. ”new” and ”york” are unusual words to find in this table. These could be related to some kind of event. But this event has yet to reveal itself. It is therefore more likely it is a natural fluctuation.

Words	Explanation
follow, back	Strongly connected words. Natural fluctuations.
last, night	Strongly connected words. Natural fluctuations.
2013, 11	Natural fluctuations.
math, test, teacher	Someone is having a math test, but not considered a real event.
new, york	Strongly connected words. Natural fluctuations.

Table 4.10: Natural fluctuations March 4 15:30 - 17:00 GMT - Eastern USA.

Event related words ratio = **0.35**.

March 7 2013

This test consists of four tests from the same day equally distributed. Each test has a duration of two hours. The goal is to show how the events of a day develops.

00:00 - 02:00 GMT

Words	Event
paul, rand, drone, #standwithrand, obama, american, power	Drone attacks in the US.
blackhawks, hawk, game, michigan	Blackhawks vs Colorado Avalanche.
miami, fl, st, heat, magic, game, fl, arena	Orlando Magic vs Miami Heat.
movie, twilight, mile	Twilight Movie and 8 mile was shown on tv.
#americanidol, charlie, nicki, nick	Charlie was judged harshly especially by Nicki in American Idol.
mac, miller	Donald Trump disses Mac Miller and his platinum record with the Donald Trump song. Mac Miller is also part of the cast in a tv show.
#duckdynasty, dynasty	new Episode of a reality show called Duck Dynasty.
carolina, maryland	North Carolina and Maryland played mid-week basketball games.
taco, bell, ranch	Taco Bell fans are angry after a delay in the launch of new tacos made with Cool Ranch Doritos shells.
#blackpeopleactivities, #whitepeople-activities	Talking about stereotypical activities for different ethnicities.
leaf, #tmltalk, goal	The artist Tom Connors died. He used to sing at every home game for Maple Leafs.
snow, storm, wind, weather, rain, cold, weather	Snowy storm in the northeast. Cold everywhere.
school, tomorrow, delay, hour	Many schools started the teaching two hours later.

Table 4.11: Real events detected March 7 2013 00:00 - 02:00 - Eastern USA.

The last event in Table 4.11 is in relation to the snow storm. Students in north-eastern schools had to wait two hours before going to school. This is illustrated in Figure 4.9.

”snow” or ”storm” was rarely mentioned in the tweets about the delayed school run. With the snow piling up outside and wind blowing it is probably obvious why there was a delay and no need to mention ”snow” or ”storm” in a limiting 140 character text. It should also be clear these are Gaussian events.

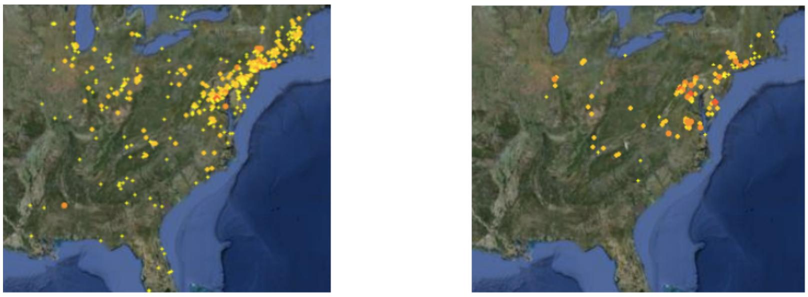


Figure 4.9: The left map displays all tweets containing the word "snow" or the word "storm". The right map displays all tweets containing the word "delay".

The bivariate spatial method is unable to detect the stormy area based on words from the school delay. This is because the graph where these words are located consists of more than 20 words extra. The extra words represents multiple other major events and therefore smooths the bivariate density function. The problem in this case is not the bivariate spatial method, but the graph it is using.

"hit", "team" and "goal" are words related to one or more sporting events. But since "leaf" and "tom" is mentioned it is probably because a singer called Tom Connors died. He used to perform at every home game for Maple Leafs.

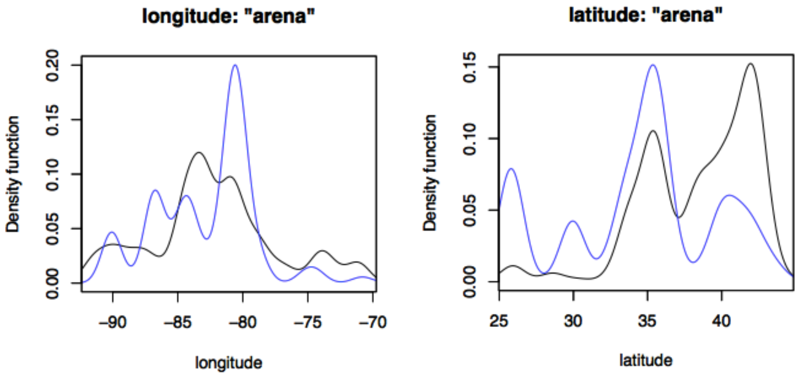


Figure 4.10: The black graphs are based on historic data. The blue graphs are based on new data. It is possible to guess where the arenas are based on the two graphs.

Figure 4.10 illustrates how "arena" is discovered as an event. From the two graphs it is possible to extrapolate that there are two geographical locations where "arena" is used more compared to the historical model. The first one is the popular basketball team, Miami Heat which played on their home-field, AmericanAirlines Arena. This field has the coordinates (25.78, 80.1). The second is a bit more difficult. It is tempting to believe it originates where ever Blackhawks played, but this is wrong. Blackhawks played at Pepsi Center in Denver. This is outside the geographical limitations of this test. The second location is in Bridgestone Arena, Nashville where Bon Jovi had a concert. The location of the arena is (36.17, 86.7).

The noise from this test is listed in Table 4.12. It consists mostly of typical noise. But also some which should have been removed by Text Parser. ”los” is the Spanish plural form and should have been removed.

Words	Explanation
ty, fb	Natural fluctuations.
best, friend	Natural fluctuations.
oh, god	Natural fluctuations.
los, ma	Should have been removed by Text Parser.
cannot, wait	Natural fluctuations.
twerk, hit, fy, new, check, track, follower, follow, back, instagram, #mentionyourfavoritefollower	Natural fluctuations.
day, need, over, 21, march, 27, 23, wife, gift	Natural fluctuations.

Table 4.12: Natural fluctuations March 7 06:00 - 08:00 GMT - Eastern USA.

The tweets about Instagram are mostly about following and having followers. ”march” surfaces because it is the start of the month and the historical model consists of tweets from February.

The venn-diagram in Figure 4.11 illustrates which of the three detection methods are used to discover the event related words. This demonstrates that all the detection methods are being used. The spatial method is the method discovering the most words. This is related to the snowstorm, ”arena” and other Gaussian distributed events.

Event related words ratio = **0.625**.

06:00 - 08:00 GMT

The density function in the left graph in Figure 4.12 illustrates how people are tweeting less and less during the evening and night with the word ”tweet”. The same decrease is observed for the rest of words which have been executed by the temporal method for this test.

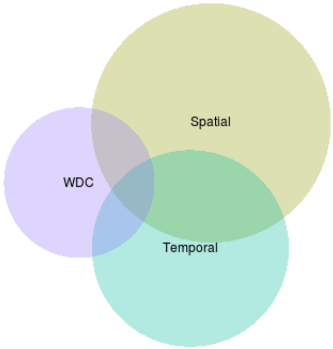


Figure 4.11: Venn-diagram of the ratio between the three detection methods.

Words	Event
5am, toronto, drake, new	A singer called Drake launched a single called ”5AM in Toronto”.

Table 4.13: Real events detected March 7 2013 06:00 - 08:00 - Eastern USA.

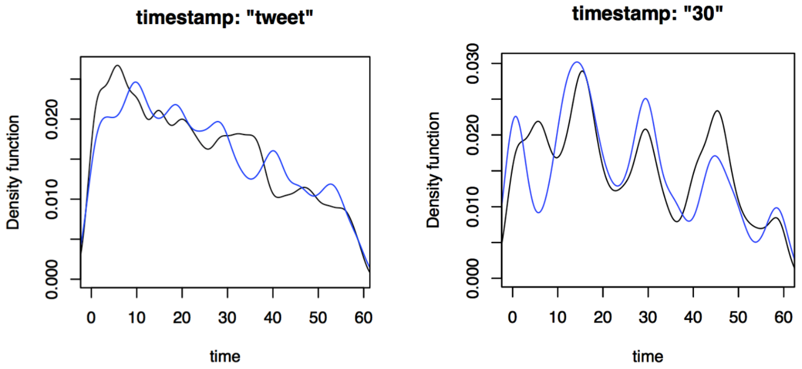


Figure 4.12: The left graph illustrates how people tweet less and less with the word ”tweet” when the time goes toward night. The right graph illustrates how people are using the word ”30” every half hour. Time 60 equals 120 minutes.

The right graph also illustrates how time of day affects the density function. In addition this graph illustrates time of hour. Every half hour a peak can be seen. Since each time step is two minutes the peaks are at 0, 15, 30, 45 and 60. These two examples illustrates how word densities fluctuates through the day and through the hours.

Words	Explanation
avi, cute	Natural fluctuations.
follow, thanks	Natural fluctuations.
night, party, house, saturday, drink, free, lady	Natural fluctuations.

Table 4.14: Natural fluctuations March 7 2013 06:00 - 08:00 GMT - Eastern USA.

Event related words ratio = **0.267**.

12:00 - 14:00 GMT

Events found during this time frame contains many of the same elements as a few hours before, stated in Tables 4.11 and 4.13.

Words	Event
cold, outside, weather, snow, snowing, today	Snowing in the northeast. Cold everywhere, but especially in Florida.
delay, hour	
new, toronto, drake, music, video, must	A singer called Drake launched a single called ”5AM in Toronto”

Table 4.15: Real events detected March 7 2013 12:00 - 14:00 - Eastern USA.

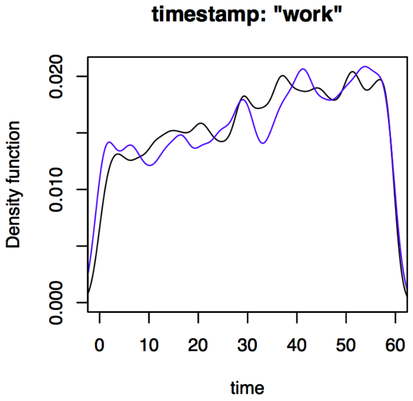


Figure 4.13: From morning towards solar noon people are tweeting more and more using the word ”work”. Each time step is equal to two minutes.

The density function in Figure 4.13 illustrates how people are tweeting more and more during the morning using the word ”work”. Twitter users are tweeting more and more about work before lunch time. Together with Figure 4.12 illustrating how people are tweeting less during evening / night. This helps to illustrate how Twitter usage fluctuates throughout the day.

Words	Explanation
good, morning	Natural fluctuations.
7th, mar	Natural fluctuations.
follow, cry, bos, mar, 7th, ny	Natural fluctuations.
beach, florida, st	Natural fluctuations.

Table 4.16: Natural fluctuations March 7 2013 12:00 - 14:00 - Eastern USA.

Event related words ratio = **0.518**.

18:00 - 20:00 GMT

Words	Event
north, korea	North Korea threatening to attack the US, Japan and South Korea.
toronto, drake	A singer called Drake launched a single called ”5AM in Toronto”
spring, break	Spring break starting.
new, facebook, feed	Facebook announce new look for News Feed.

Table 4.17: Real events detected March 7 2013 18:00 - 20:00 - Eastern USA.

Fewer words are used to describe the event of the artist Drake launching the single 5am in Toronto. It could mean the event has already peaked and is already declining. The historical data have no more tweets about the new single than the the historical data used to find the events in Table 4.13 and have probably not anything to do with the decrease.

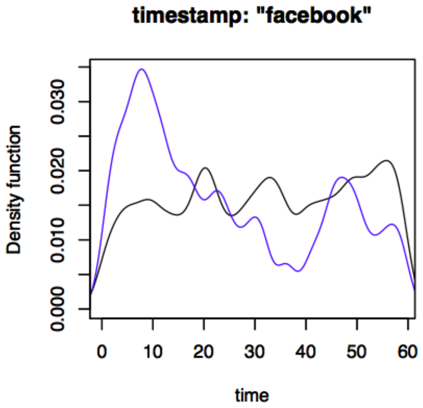


Figure 4.14: Facebook performed a major release at 18:00 GMT. The black graph is based on historic data. The blue graph is based on new data. Each time step is 2 minutes.

Facebook made a major release. This is illustrated in a temporal context in Figure 4.14. The number of mentions for ”facebook” peaks at 18:20. At 19:20 it reaches the low point before the number of mentions increases again. This shows how the temporal method can work.

Words	Explanation
march, bday	Natural fluctuations. ”bday” is an abbreviation for birthday.
da, ”, gt, listen	Natural fluctuations.
wind, today	Could not relate to a specific event.
miami, beach, fl	Natural fluctuations. ”fl” is Florida .
shake, harlem, night, video, saturday, free, party, drink, lady, house, shoot, ave, #coupon, gas	Natural fluctuations.

Table 4.18: Natural fluctuations March 7 2013 18:00 - 20:00 - Eastern USA.

Event related words ratio = **0.36**.

March 19 18:00 - 20:00 GMT - Eastern USA

Words	Event
terry, jason	LeBron's dunk was the cause of Jason Terry's death on Wikipedia.
campus, gun, student	Man with gun seen on campus in Indianapolis.
rain, today, wind, snow	Snow and rain north east of New York. Windy in Ohio and Pennsylvania.
weather, nice, cold, special	Cold in the north.
beautiful, day	Nice day in the south.

Table 4.19: Real events detected March 19 18:00 - 20:00 - Eastern USA.

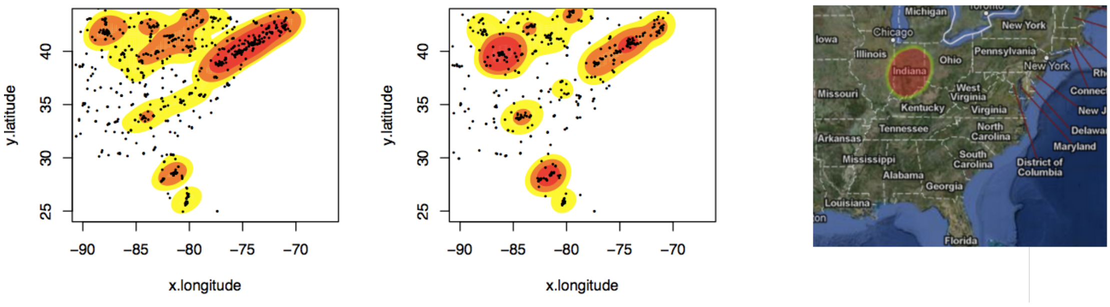


Figure 4.15: Person seen with gun on campus in Indianapolis. The first kde is based on historic data. The next kde is based on new data. The difference can be seen on the map to the right.

Figure 4.15 illustrates historic and new kde for the gun incident on a campusu in Indianapolis. The map on the right is the result. It approximates the affected area of this Gaussian event.

Table 4.20 contains the noise from this test. It is worth noting there is some spam in the last row. Ideally this kind of spam should be removed either by Twitter or by this solution. Although it is annoying it is simple to identify using Grapher.

Words	Explanation
park, check	Difficult to find specific park, but probably have something to do with it being nice weather in the south.
favorite, show	Strongly connected words. Natural fluctuations.
best, friend	Strongly connected words. Natural fluctuations.
thanks, follow, back	Strongly connected words. Natural fluctuations.
watching, movie	Strongly connected words. Natural fluctuations.
miami, beach	Strongly connected words. Natural fluctuations.
album, new	Could not find a specific event.
free, 13, service, 21, ppl, 100, 1st, bottle, ticket, shine, #coupon, 19, st, ma, ave, rd, tu, los, lane, je	Some kind of spam.

Table 4.20: Natural fluctuations March 19 18:00 - 20:00 - Eastern USA.

Event related words ratio = **0.3**.

March 21 06:00 - 08:00 GMT - Eastern USA

The event detector could not find any events, but some noise was intercepted.

Words	Explanation
thing, coming, ready	Natural fluctuations.

Table 4.21: Natural fluctuations March 21 06:00 - 08:00 GMT - Eastern USA.

Event related words ratio = **0.0**.

March 26 12:00 - 14:00 GMT - Eastern Seaboard of the US

Words	Event
spring, break, snowing, snow, cold, degree, weather, march, easter	People are planning spring break, but it is cold and snow on the ground.
supreme, court, #scotus, gay, marriage, support, state, red	Supreme court weighted whether gay couples have a constitutional right to marry. Many marriage equality activists used a red square with a pink equal sign to show support.

Table 4.22: Real events detected March 26 12:00 - 14:00 GMT - Eastern USA.

Words	Explanation
please, follow, dm, #catsaresluts	Natural fluctuations.
today, 26th, fl, 00, park	Natural fluctuations.
good, morning	Natural fluctuations.
dia, kiss	Natural fluctuations.
last, night	Natural fluctuations.

Table 4.23: Natural fluctuations March 26 12:00 - 14:00 GMT - Eastern USA.

Event related words ratio = **0.531**.

March 29 18:00 - 20:00 GMT - Eastern USA

Words	Event
movie, temptation	The movie Temptation premiered.
joe	The movie ”G.I. Joe: Retaliation” premiered the day before.
good, 29, 30, friday, st, church, jesus, #goodfriday, cross	Good Friday and Easter. People are going to church.
easter, egg	With Easter comes Easter eggs.
sunny, weather	Mostly nice weather.
rain	Raining in Tennessee.
spring, break	Spring break.
zoo, garden	People are going to the zoo and attending their garden.
central, park, #nyc	Many people in Central Park.

Table 4.24: Real events detected March 29 18:00 - 20:00 GMT - Eastern USA.

The bivariate spatial method discovers an area where the words ”central”, ”park” and ”#nyc” are used significantly more than usual. This area have no correlation with Central Park in New York. But some Twitter users are using these words in the vicinity of Baltimore. It therefore marks this area. In the previous and coming discussion of tests this is the only time the bivariate spatial method has failed to approximate the area of an event.

Words	Explanation
state, university	Natural fluctuations.
fl, miami, beach, hot, id, hotel, station	Natural fluctuations.
tonight, lady, party, money, open, plus, free, twerk 12, till, black, saturday	Natural fluctuations.

Table 4.25: Natural fluctuations March 29 18:00 - 20:00 GMT - Eastern USA.

Event related words ratio = **0.533**.

March 30 06:00 - 08:00 GMT - Eastern USA

No events were found, but some noise emerged. These are listed in Table 4.26.

Words	Explanation
used, #elementaryschoolconfessions	Twitter typical.
need, new, good, follower, help, check, hit, lil, sum, yu, whats	Natural fluctuations.

Table 4.26: Natural fluctuations March 30 06:00 - 08:00 GMT eastern USA.

Event related words ratio = **0.0**.

Summary of Performed Tests

The solution in this thesis is able to detect a variety of events. In some cases the solution is able to show an incredible level of detail (the Oscars 4.4.1). On the negative side some noise is produced. The solution could therefore be deemed sensitive. It is easy to detect events with great detail and noise is easily produced. Grapher makes it easy to avoid noise. Noise consists often of very few nodes and in most cases only two nodes.

The summed Event related words ratio is calculated with

$$\frac{\sum n_{event}}{\sum n_{event} + \sum n_{noise}} \tag{4.2}$$

and when substituting and calculating the event related words ratio it gives

$$\frac{423}{227 + 423} = \mathbf{0.65}. \tag{4.3}$$

This approach of quantifying the performance does not fully explain the performance of the solution, but provides an overview of how well it does in relation to noise. Translating the test results from graphs to tables looses how the words are connected.

In the problem statement four research questions were asked. Two of those will now be answered and the two last will be answered in Subsection 4.4.2.

1. Is it possible to detect general events by comparing word densities, comparing kernel density estimates for each word in a spatial and temporal context and clustering the detected words with odds ratio?

As demonstrated in the previous sections the solution is able to detect a wide variety of events. A bi-product of detecting events is noise. These words which have no relation to specific events are fortunately in many cases easy to spot in Grapher. They are typically located in smaller graphs. Example of noise are:

- ”please”, ”follow”, ”thanks”, ”back”.
- ”good”, ”morning”.
- ”last”, ”night”.
- ”miami”, ”beach”, ”fl”.
- ”free”, ”lady”.
- ”cannot”, ”wait”.
- ”math”, ”test”, ”teacher”, ”homework”.
- ”happy”, ”birthday”.

Major events often consists of graphs with many nodes. If the event is really large it commonly hold the biggest share of nodes in the test result. Among the highlights from the discussed tests are:

- Martin Luther King Day (Table 4.2).
- President Obama’s inauguration (Table 4.2).
- The Oscars in great detail. Everything from who won a best supporting actress to who had a beard (Tables 4.4, 4.5, 4.6).
- People watching a tv series called The Walking Dead (Table 4.7).
- Sporting events like Miami Heat vs Cavs with NBA superstar James LeBron (Table 4.11).
- Special weather conditions like for example snow storms in the northeast (Tables 4.7, 4.9, 4.11, 4.13, 4.19, 4.22, 4.24).
- Drone attacks in the US. Paul Rand and Obama as central figures (Table 4.11).
- Taco Bell fans angry after a delay in the launch of new tacos (Table 4.11).
- An artist called Drake launched a new album called 5am in Toronto (Tables 4.13, 4.15, 4.17).
- North Korea threatening to attack the US (Table 4.17).
- Supreme court weighted whether gay couples have constitutional rights to marry (Table 4.22).

- People going to church on Good Friday during easter (Table 4.24).

2. Is it possible to estimate the affected area of an event occurring in a limited geographical area comparing bivariate kernel density estimates in a spatial context?

The bivariate spatial method makes it possible to approximate the area of a geographically distributed Gaussian event. In the discussed tests it is able to do a good job estimating the area affected by an event.

Of the discussed tests there is only one instance where the approximation have been wrong. It is described in Section 4.4.1.

The bivariate spatial method is depending on a graph which often contains multiple words. The words in a graph can be connected to multiple events. This makes it more difficult for the spatial method to perform. Despite this it produces good results where these are:

- False tornado warnings in Wisconsin (Figure 4.8).
- Man seen with gun on campus in Indianapolis (Figures 4.15 and table 4.19).
- Thunder and rain between Pensacola and New Orleans (Table 4.7).
- Snow in the northeast (Table 4.15).
- Nice and cold day in North Carolina (Table 4.19).
- Wind, rain and snow in the northeast (Table 4.19).
- Snowing in Atlanta (Table 4.22).

To further examine these results utilize Grapher. A guide on how to connect to Grapher is found in the appendix.

The bivariate spatial method is not the only method able to detect an event in a spatial context. The univariate spatial method is able to detect where words are used more than normal. Figure 4.10 shows how the word ”arena” is detected. Where the blue graph has a higher value than the black graph is where the affected areas are.

4.4.2 Crisis Detection

The crisis data is from the terrorist attack in Boston, April 2013. The first bomb went off 18:49 (GMT). The collected data is from 19:24 (GMT). The first 35 minutes of the crisis is therefore not part of the data set. Ideally there should have been a couple of more tests and they should have started a couple of hours before the bomb went off, but due to data shortage it has not been done. The bombings happened after the testing phase of this project was supposed to end and therefore the tweet retrieval process was not monitored sufficiently. Despite this the results are quite good.

Two different types of tests have been performed. The first type consists of four tests and covers Eastern USA (defined in Section 4.4.1) and has time frames of one hour. The second type consists of one test and covers the Greater Boston Area, but has a longer time frame.

The discussed results will differentiate between peoples emotions and information of a more useful character. Normal fluctuations will also be accounted for.

Terrorist Attack on Boston Marathon - Eastern USA

This part discusses four successive tests, each with a time frame of one hour. The first test starts 35 minutes after the explosions.

The results are very similar. Only the first test will be properly examined and the following tests will focus on the differences compared to the previous tests.

19:24 - 20:24 GMT

Figure 4.16 illustrates the tweet vocabulary from the Boston bombings compared to the vocabulary used in historic tweets. From this single graph it is possible to understand something extraordinary is happening.

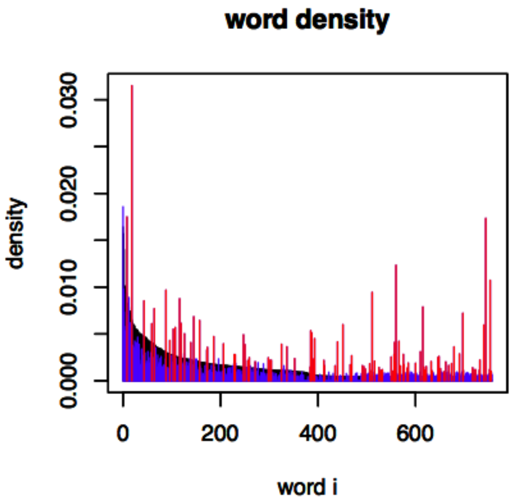


Figure 4.16: Result of WDC method from the Boston bombings. The red bars show the words which have a significant higher share than they did before.

WDC suggests many words. Figure 4.17 illustrates how it compares to the other two detection methods. WDC and the spatial method both suggest about 140 words each. The temporal method only suggests 9 words. This contributes to confirm this is a Gaussian event. If the time frame of the test was when the bombing started it is conceivable the Temporal method would have had as big impact as the other two. The reason for the WDC to have such a major share is that many of the words are not commonly used. The temporal method and the spatial method need a certain amount of historical and new mentions to work therefore some words can only be discovered by WDC.

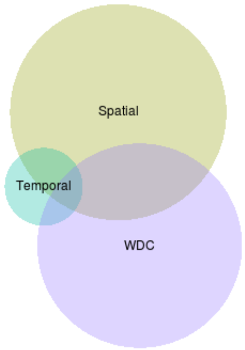


Figure 4.17: Venn diagram showing the ratio between the detection methods from the Boston bombings.

Many words are used more frequently in the Boston area compared to the historical data. These are for example ”phone”, ”car”, ”crazy” etc. This coincides with a Gaussian event. But some exceptions exists. ”boston” is one of those. In Figure 4.18 the new graph has more spread than the historic graph. This word is historically used close to Boston and less the longer away you get. Because the bombings happened in Boston and the news value of this terrorist attack was national and international it is natural Twitter users all over the USA use ”boston” more frequently in their tweets.

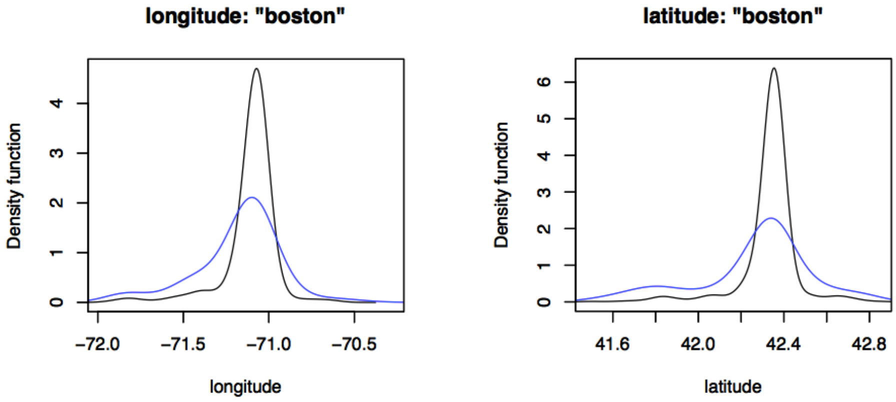


Figure 4.18: The word ”boston” is used in a wider geographical area compared to historic tweets.

Figure 4.19 are all the graphs found in this time interval. The huge graph in the middle is related to the terrorist attack, but so are many of the smaller surrounding graphs. It illustrates how a test with a complex result can look like in Grapher.

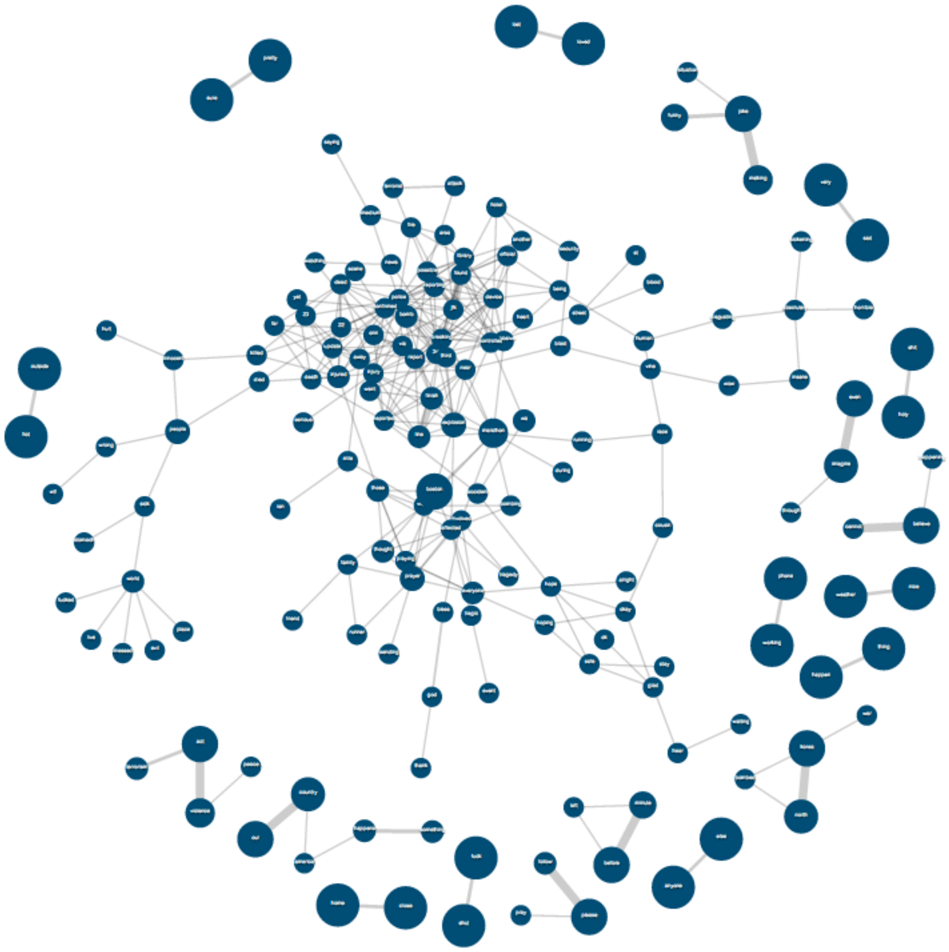


Figure 4.19: Overview of graphs from the Boston terrorist attack.

Table 4.27 lists up all the events considered useful to for example a crisis handling team. From Grapher it is difficult to determine if there are 22 or 23 dead or injured. ”22” and ”23” are both related to ”injured” and ”dead”. This makes it difficult to read the correct information, although going to the map helps. Because 22 is used more often than 23 it is more likely to be the correct number. Reading some random tweets containing 22 and 23 shows there are 2 - 3 dead and 22 - 23 injured.

It should be noted that the two most significant nodes are ”boston” and ”marathon”. The two most significant edges are connected to [”finish”, ”line”] and [”jfk”, ”library”]. These are vital to describe the attack.

Words	Event
boston, marathon, explosion, blast, near, finish, line, area, bombing, report, running	Explosion / bombing near the finish line in Boston Marathon.
jfk, library, fire, medium, another, device, explosive, 3rd, third, bomb, went, reporting, explosion	A fire in JFK library was the cause of an explosion and was incorrectly believed to be in relation to the terrorist attacks.
possible, terrorist, attack	Uncertainty if it actually is a terrorist attack or an accident.
22, 23, died, dead, death, killed, serious, injury, injured, confirmed, yet, far, innocent, people, hurt	22 injured and 2 dead.
hotel, security, possible, another, device, being, found	Unexploded device found outside Boston’s Mandarin Oriental Hotel. Many hotels evacuated.
watching, breaking, news, cnn, via, heart, scene	People watching the news.
vine	People linking to Vine videos of the explosion.
north, korea, bombed, war	Some people believed North Korea was responsible for the bombings.
phone, working	Police took down the phone service in the city center.
act, violence, terrorism, peace	Many Twitter users think it is a terrorist attack.
mile, ran, far, away, saying, victim, during	Many knew or where about a mile close to the finish line.
police, update, reporting, official, reported, watching, controlled	
blood, street, st	Blood on the streets near the finish line.

Table 4.27: Useful information related to the Boston bombings 19:24 - 20:24 GMT - Eastern USA.

Words	Emotion
human, disgusting, absolutely, sickening, horrible, insane, wow,	Shock and anger.
people, wrong, wtf, innocent, sick, stomach, world, live, fucked, evil, place, messed	Shock and anger.
our, country, something, happens, america	Despair.
holy, shit	Holy shit.
loved, lost	Afraid of having lost loved ones.
before, minute, left	People who were close to the bombing in time and place.
very, sad	People are very sad
joke, making, funny, situation	People should not make fun of the situation.
pray, praying, thought, please, follow	Praying.
family, friend, runner, friend, sending, prayer, victim	Many know someone who was running. and hope everything is ok with them.
hoping, safe, everyone, okay	Hope.
cousin, race, okay, hope, ok, hoping, alright, safe, stay, glad, hear, waiting	Hope.
involved, tragedy, affected, those, everyone, marathon	Sadness.
bless, god, thank, tragic, event	People glad their loved ones are ok.
cannot, believe, happening	Difficult to grasp.
weather, nice	Weather is nice.
hot, outside	Warm weather.

Table 4.28: Emotions related to the Boston bombings 19:24 - 20:24 GMT - Eastern USA.

Words	Explanation
pretty, sure	Natural fluctuations.
anyone, else	Natural fluctuations.
imagine, even, through	Natural fluctuations.
shut, fuck	Natural fluctuations.
home, close	Natural fluctuations.
wa	Natural fluctuations.
happen, thing	Natural fluctuations.

Table 4.29: Noise 19:24 - 20:24 GMT - Eastern USA.

Event related words ratio = **0.915**.

20:24 - 21:24 GMT

The results are pretty much the same as the previous hour, but with minor differences. It is more clear the cell phone service was clogged due to heavy traffic. It also became more clear this was no accident, but a terrorist attack. The word ”muslim” is connected to the words, ”blame”, ”terrorist” and ”american”. This is a guess since no confirmation existed at this point.

The word ”newtown” is connected to ”family”, ”apparently”, ”shooting”, ”explosion” etc. This is because families of the Newtown school massacre was together on an event related to the Boston Marathon.

The word ”hospital” is connected to the words ”suspect”, ”runner”, ”ran”, ”finish”, ”line” and ”blood”. Cops guarded a suspect at the hospital. A TV channel reported about runners who crossed the finish line and continued to the hospital to give blood.

Event related words ratio = **0.961**.

21:24 - 22:24 GMT

Figure 4.20 illustrates how the bivariate spatial method works. The first map is a KDE based on historical data. The black dots represent tweets. The map in the middle is a KDE based on new data. It is possible to see there are more tweets in the Boston area compared to the historical data. The rightmost map shows the positive difference between the historical KDE and the new KDE.

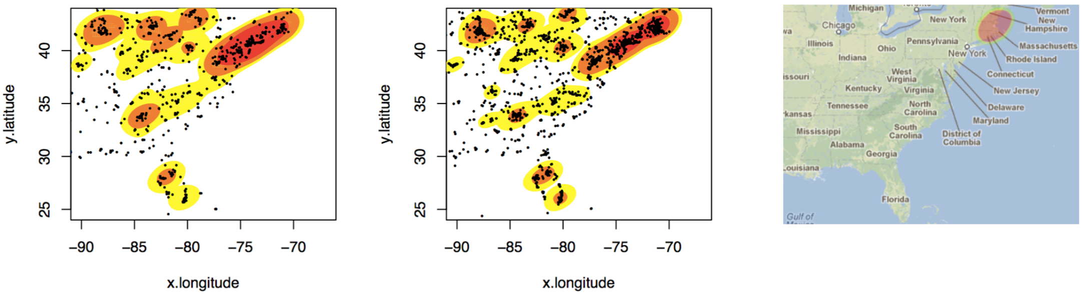


Figure 4.20: The first leftmost KDE is based on historical data. The middle KDE is based on new data. The rightmost map is the difference between them.

President Obama made a speech to the nation. A sentence cited by Twitter users was: ”Boston is a tough, resilient town and so are its people”.

Event related words ratio = **0.982**.

22:24 - 23:24 GMT

No significant change from previous tests.

The emotions related words found in Table 4.28 describes the emotions Twitter users had right after the bombings. These correspond quite well with the words found in an experiment performed by the University of Vermont.

University of Vermont is currently performing an experiment called the Hedonometer [24] where they measure the happiness of people in the USA for each day. When this is written the Boston bombings was the saddest [25] day in the four years the Hedonometer had operated on. The Hedonometer presents the most influential words. For the day of the bombings, these were:
sad, sick, explosion, victims, injured, died, bomb, tragedy, prayers, love, bombs, suspect, dead, haha, bombing, blood, killed, no, terrorist, horrible, hospital, shit, hate, terrible, attack, world, me, hahaha.
Except for the words *hahah, hahahaha, hate, me* and *no* they are found in one or more of the above tests. *no* and *me* are found in the stop word list and therefore have no chance at being present. Some words are in a different form because of lemmatization.

Event related words ratio = **0.964**.

Terrorist Attack on Boston Marathon 19:24 - 01:24 - Boston Area

The geographical area this test focus on is illustrated as the map in Figure 4.21. It also shows areas of unusual high Twitter activity. This area is from the largest graph in this test.

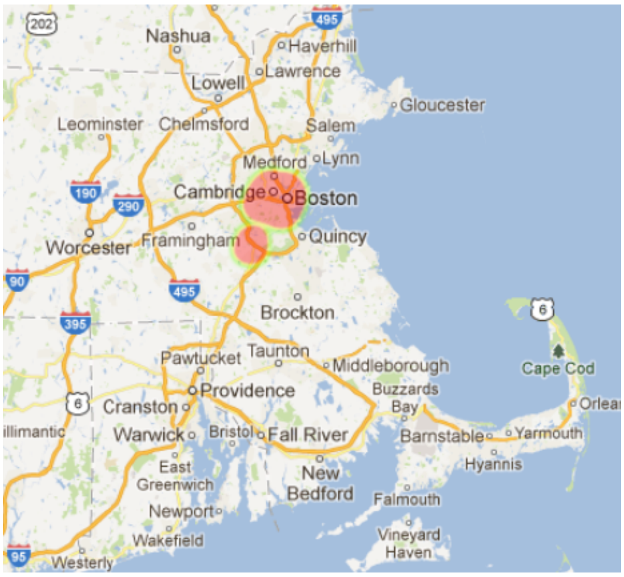


Figure 4.21: Results of the bivariate spatial method.

The results are very similar to the previous crisis results, but with fewer words describing the event. The most important details are still present. The number of tweets analyzed is smaller compared to previous tests and this is the probable cause. The nodes are also more loosely connected. This is because the number of tweets is less than in the previous tests.

Event related words ratio = **0.872**.

Summary of Crisis Centric Tests

To quantify the performance, Equation 4.2 is used. The event related words ratio is

$$\frac{585}{38+585} = \mathbf{0.94}. \tag{4.4}$$

Where 585 is the number of words related to an event. 38 is the number noise related words. The ratio is much higher than for general events. This is probably because this event had a national and international impact. It is probable people are very concerned, especially when a terrorist attack occurs in their own country.

The two last research questions are in the following questions answered.

1. Is it possible to detect crisis events by comparing word densities and comparing kernel density estimates for each word in a spatial and temporal context and clustering the detected words with odds ratio?

As demonstrated in the previous sections the solution is able to detect a crisis event in great detail. Detecting general events gave much more noise than in these tests. Crisis events are often of major importance and are therefore represented in Grapher with many nodes. The event related words ratio is much better than for general events.

Among the highlights from the discussed tests are:

- The bomb went of close to the finish line.
- Explosive fire in JFK library. Falsely believed to be in relation to the bombings.
- Cell phone service went down in the Boston area due to clogging.
- Blood in the street.
- People linking to Vine video from the bombing.
- Unexploded device found outside Hotel.

According to the Hedonometer developed by the University of Vermont [24] the top words describing the emotions of the day of the Boston bombings were almost all found by the solution approach in this thesis. This is a strong indication that the detection methods are working. See the above test for more information about the Hedonometer and its correspondent findings.

2. Is it possible to estimate the affected area of a crisis event occurring in a limited geographical area comparing bivariate kernel density estimates in a spatial context?

The bivariate spatial method makes it possible to approximate the area of a geographically distributed Gaussian event like a crisis situation. In the discussed tests it is able to do a decent approximation of Boston. It does lack some accuracy.

One could argue the best would be to locate the exact position of the explosions. A bomb exploding in the dessert is not affecting anyone and people would not bother to tweet about it. Explosions in a densely populated area is of more concern. With this understanding the bivariate spatial method works quite well. People who feel directly affected are more likely to tweet about the situation. In short, the solution approximates the areas which ”feels” struck by the bomb. This can have some unfortunate effects. If people who live further away also feels struck by the bombings also tweet a lot other areas will also be highlighted. Figure 4.21 illustrates this with the highlighted area south west of Boston.

To further examine these results utilize Grapher. A guide on how to connect to Grapher is found in the appendix.

4.5 General Aspects of Solution

This section discuss various technical aspects of the solution. For example how fast the different parts of the solution pipeline executes. All tests have been performed on 4 year old mid range computers.

It should be noted that the solution pipeline is intended to execute in a server environment with multiple machines. The solution can therefore afford to be somewhat computational expensive.

Tweet Retriever is simple and efficient and will therefore not be discussed in this section. But the detection pipeline and Grapher will be discussed.

4.5.1 Detection Pipeline

The detection pipeline is the slowest performing part of the solution pipeline.

Three Detection Methods

Because the spatial detection method is not bivariate, but univariate it executes as quickly as the temporal method. WDC is quicker than both KDE methods. All the methods use on average about 12 minutes to execute. The methods are performed in series, but execution time could be cut down by doing them in parallel on a multiprocessor system.

The temporal method and spatial method have natural limitations when it comes to what they can detect. They also produce some noise which is natural based on what time frames the historical tweets are based on. Massive terrorist bombings from the time frames the historic data is based on would result in few words related to the terrorist attack. To avoid this it could be possible to mark time frames from terrorist attacks.

A more unavoidable problem is noise. Using the day before and week before the day of week and current month can often surface. More annoying noise is [last night], [good morning] etc. These can occur respectively on Sunday mornings and Monday mornings. People are up late on Saturdays compared to Fridays. People wake up earlier on Mondays than on Sundays. These examples are more of a challenge to remove.

Clustering with Odds Ratio

The three detection methods produce an excessive number of words which are potentially related to events. The execution time increases exponentially with the number of words produced by the three methods. This is therefore by far the slowest part in the solution. For the larger nodes it can take multiple hours to execute.

Bivariate Spatial Method

The number of positions in each statistical model cannot exceed 1800 positions. If the number of positions exceeds 1800 positions the list is shuffled and sliced down to 1800. This is referred to as sampling. This is performed because comparing kernel density estimates is computational complex.

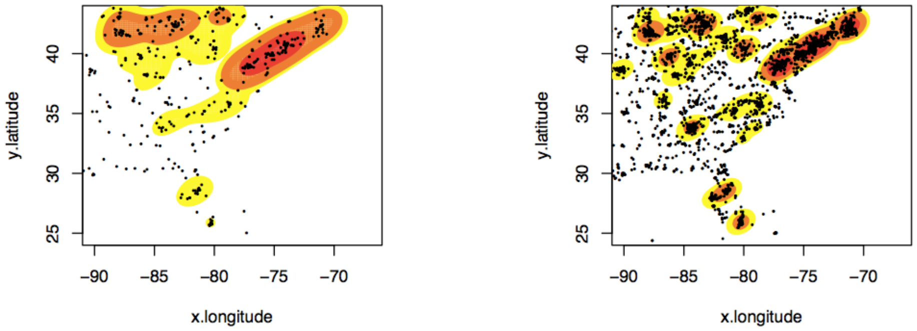


Figure 4.22: Left KDE is based on 500 positions. Right KDE is based on 9000 positions.

Figure 4.22 shows two KDEs with different sample sizes. The KDE consisting of few positions is smooth and has a lack of detail. The KDE consisting of many positions has an abundance of details.

The ideal approach would be to utilize all available positions. Sampling makes the statistical detection method less accurate, but it is still able to find major differences if they exist. See Section 4.4.2 for an example of this method in use.

4.5.2 Grapher

Visualizing the results produced by the detection pipeline is vital to how the analyst perceive the words. The graphs found in the tests makes it easy to read what the event is about. If there were no graphs, but instead a textual representation it would be much more difficult to understand what event the words were describing.

The JavaScript library D3 used to illustrate the graphs in Grapher is based on gravitational forces and spring systems. This ensures the largest graphs are in the middle while the smallest graphs are on the edge of the screen. This focus the attention of the analyst to the important nodes. Graphs with only two nodes are therefore often on the edge of the visible area or completely nudged out of the canvas.

The approach to read the graph is to start with an interesting word (node). Then the connected words are read. Two nodes which are directly connected are found in the same tweet. If there is an mediate node then they the two words have a lower odds ratio and therefore does not appear as often in the same tweets.

The map is not necessary to understand if the words are related to an event. But it is important to see an approximation of the area of which the event affected. It is also helpful to click on the tweet positions to get the tweet text. It can help confirm the existence of the event by reading some tweet texts. A tip to better see the tweet positions is to only check the words you want to visualize. Often it also is better to switch to satellite imagery without labels to get a better contrast to the dots.

4.6 Future Improvements

Even though the solution is a completely working event detector there are many possibilities for improvements. These can be divided into event detector improvements, execution performance improvements and visualization improvements.

4.6.1 Event Detector Improvements

This subsection discuss how to improve the event detection. More specific where in the pipeline it is room for improvement.

Text Parser could have an updated frequently used word list. Specially with Spanish words. It should also be some sort of spam remover. If for example a user is posting many tweets with typical spam content like sale, coupon etc. the tweet should be ignored. One approach could be to train a Naive Bayes classifier [32]. When an analyst perceive a tweet as spam he / she should mark it as spam. Then the classifier could train with this new data and this way update itself.

The lemmatizer in Text Parser could also perform better. It could be beneficial to look into replacements for the Python NLTK Lemmatizer which is utilized. But a better lemmatizer has not yet been found. Implementing one could be done. One approach could be to gather the 10 000 most commonly used English words in their different forms. Implementing a fast way to look up in this table would be the challenge. The goal of the lemmatizer is to get a grammatically correct base of the a word in any form. It is although possible this approach is to computational complex.

The spatial and temporal detection methods are difficult to improve. It is tempting to make the spatial method bivariate. Doing this would require more computational power. It could be necessary to implement the R functions in a distributed system. For example using MapReduce. See Section 4.6.2 for more information on MapReduce.

WDC have some room for improvements. In its current form it is a bit too rigid. The non-linear function determining if a word is unusual could be improved. It could be more flexible when it comes to the total number of mentions in the data set and the mentions for the word it is evaluating.

The bivariate spatial method performs quite well, but could in come cases narrow down the approximated area of the event. This could be done by first detecting an event area. It could then check if any of the words in the graph was suggested by the temporal method. This could help estimate when the event started. Because it takes some time before the temporal method detects the peak it would be advisable to maybe find a timestamp a little before the peak. Maybe somewhere on the incline. This timestamp could be used to weight the word positions based on when the tweet was posted. A linear weight function could be created. No positions would be used prior to the found timestamp. The position weight would decrease as the timestamp of the position increased. This approach is based on the idea of geographically distributed Gaussian event. People close to an event will be quicker to tweet about something close to them. Especially if the event is of a major importance.

The other parts of the event detection pipeline is working very well.

4.6.2 Execution Improvements

This subsection discuss how to improve execution time and scalability of the solution. The time it takes to execute the event detection pipeline is too great. It can take anything from a couple of minutes to hours. This is on a four year old mid range desktop computer. Because this is supposed to run on a server environment computational power can decrease the execution time of a test.

Large amounts of data is persisted to multiple databases. Some of the databases are too big and show a decrease in performance. This is not a problem for this project, but for continuous use it would be quickly be a major problem. To different approaches could be used to solve this. The first is a traditional sharded database. For example a MySQL environment sharded based on tweet dates. With about 20 million tweets in each shard the system would perform well. A similar environment for the result database would also be needed. This is could be considered a traditional solution to the problem. A more unconventional solution to the problem could be to use a NoSQL environment. More specifically a graph database called Neo4j could be utilized. This database would fit very well to the result database with the clustering data. Some research would have to be performed for such a solution to usable. A mix of the two approaches might also work.

There are two methods which are particularly time consuming, odds ratio and the bivariate spatial method. The easiest and most important job is to improve the clustering method. Two approaches are devised in the next paragraphs. Because the execution time is depending on the number of words detected by the three detection methods (Section 3.2.2) it could take 24 hours to finish the execution of one clustering job. To avoid this it could be possible to limit the number of suggested words from the three detection methods. Setting a threshold to the number of words which makes the execution of the clustering method acceptable could be wise. This would make the execution time of the clustering method constant since the number of words would be constant. The challenge is how to slize down the amount of words produced by the three detection methods. Before slizing the words they should be ordered according to ”how detected” they were by combining the findings from each of the detection methods. This is difficult because the three detection methods work independently. If only the temporal method detects a word, but none of the other methods. It is not certain this word is less important than a word discovered by all three detection methods. Ranking and slicing the words suggested by the three methods might therefore be a challenging task.

A second approach which could reduce execution time of the clustering method could be to distribute the computation. Using MapReduce might be viable option. MapReduce is a model for processing large amounts of data in a distributed environment. A master node maps what part of the data should be processed by what node. When the nodes finish the results are combined and reduced to a single data set. A well known MapReduce library is Apache Hadoop.

4.6.3 Visualization Improvements

The current visualization is working well as it is now, but adding some more functionality could make more information more easily accessible.

When clicking on a node it would be practical to get information about what detection method detected the word. If the spatial and / or the temporal method was used also provide the graphs. In general the details explaining why the words was suggested. These details are today persisted to databases, text files and pdfs. Since all the data exists it should be manageable task.

Another visualization improvement is to show the odds ratio when hovering the mouse over an edge. In complex graphs where the edges often are of the same width it could be very useful to for example compare multiple odds ratios.

Grapher could be faster. Specially when test results are complex. The map can be slow when there are many dots are set.

There is also a known bug which should be fixed. This happens when deselecting words in the map page. If the word contains a single quote it is not able to remove the dots from that word.

4.7 Solution Applications

The solution is more than just a crisis detector, it is also an event detector. Possible applications therefore increase dramatically.

As an event detector there is a wide range of applications. A newspaper could use the solution to find what people care about. With this basis they could do in-depth articles on the detected topics. The solution of this thesis is well suited because it can be aimed at a specific area and duration. Newspapers cover specific areas mostly on a daily basis. It might also measure if articles they write makes an impact on Twitter. This could also be used by marketers to see if their product could or campaign made an impact on Twitter. This service could be sold as a service where marketers payed a monthly fee independent of how many report they receive.

As a crisis detector it can give an overview of a crisis situation given it is of a certain magnitude. For example like the Boston bombings. For a crisis handling team it could be useful to get information from other media sources than callers and traditional media. Insights to what people are thinking could be very advantageous. Rumors could for example be invalidated to calm down the population and make sure the official version reached out.

Chapter 5

Conclusion

The goal of this thesis is to detect crisis events. To make sure all kind of crisis situations are detectable, the solution is constructed as a general event detector. A crisis is an event often of major importance to those living close by. A crisis can therefore make a big impact on an event detector monitoring the area where a crisis event occurs. The secondary goal of this thesis is to estimate the area affected by an event.

The solution is implemented as a three part pipeline. The first part retrieves tweets from the Twitter streaming API. The second part is the core and consists of its own pipeline called the detection pipeline. The last part is a web site called Grapher. It visualizes the test results.

The detection pipeline is the core of the solution. It consists of a temporal, word density and two spatial detection methods. In addition the detection pipeline clusters the suggested words from the methods by calculating odds ratios. The detection methods are comparing two statistical models based on historic data and new data. The two spatial methods and the temporal method detects words and locations by comparing kernel density estimates with a state-of-the-art method [20]. Kernel density estimation is a mean of constructing a nonparametric density function. A summary of these detection methods and their contributions to this field of study is shown in Table 5.1.

Method Name	New Approach	New Problem	Assessment
Spatial method	Yes	Yes	Works as intended.
Temporal method	Yes	No	Works as intended.
WDC	No	No	Works as intended, but is sensitive to the number of words it is analyzing. It cannot be too many or too few.
Clustering with odds ratio	Yes	No	Works as intended, but is computational complex.
Bivariate spatial method	Yes	Yes	Works as intended, but relies on the graph not containing too much noise.

Table 5.1: Summary of the methods in the detection pipeline. See Section 4.4.1 and Section 4.4.2 for more details.

More than 90 million geotagged tweets have been retrieved as test data for the proposed solution of this thesis. Minor studies have been performed on the geotagged tweets. A summary of these can be seen in Table 5.2.

Study	Findings
Geotagged Tweets Compared to Non Geotagged Tweets	Japanese tweeters geotagging their tweets has a different vocabulary than Japanese tweeters who does not geotag their tweets. It is hypothesized if the two groupings of Twitter users belongs to different demographic groups.
Temporal Analysis of Geotagged Tweets	In a little over a month the daily number of geotagged tweets increased from 1.2 million to 1.4 million.

Table 5.2: Summary of the two studies on geotagged tweets. See Section 4.1 and Section 4.2 for more details.

To thoroughly validity the proposed solution and hence answer the research questions more than 300 tests have been performed where 17 of these are discussed in this thesis. The results are very good although some noise is produced. The proposed solution is able to detect both crisis situations and events of a more general character.

Among the highlights are President Obama's inauguration, the Oscars, lots of sporting events, and drone attacks in the US. Among the highlights for estimating the area affected by an event are special weather conditions, false tornado warnings in Wisconsin and man seen with gun on campus in Indianapolis. The results are a bit noisy. The event related words ratio is **0.65** for general event detection.

The solution have been tested on real Twitter data from the Boston bombings. It performed very good. It is able to detect numerous details. Among the highlights are that the bomb went of close to the finish line, cell phone service was clogged, blood in the streets, unexploded device found outside hotel and the number of people injured and dead. The event related words ratio is **0.94** for crisis events. The solution is also able to detect that the bombing occurred in Boston.

Although the solution is working well there are room for improvements. The most significant improvement would be to distribute the clustering method by applying MapReduce. This would require more computational power, but reduce execution time.

Since the solution is able to detect general events it could be used by organizations monitoring social media. The solution is also capable of estimating the affected area of many events and could therefore be used by crisis handling teams to monitor a crisis and its aftermath.

Acknowledgments

I would like to thank my supervisor Ole-Christoffer Granmo for his enthusiasm, constructive feedback and his help in the selection of methods. His help has been greatly appreciated.

Furthermore I would like to thank my dad, James Karlsen for giving valuable feedback on the thesis report.

Also I would like to thank my co-worker Sondre Glimsdal for his valuable feedback and specially on the technical part of the thesis report.

Finally I want to thank my workplace Integrasco and my boss, Tarjei Romtveit for being understanding and granting leave the two last months of the workings with this thesis report.

University of Agder, 2013

References

[1] Kiss, J. (2012). Facebook hits 1 billion users a month. , . retrieved April 19, 2013, from http://www.guardian.co.uk/technology/2012/oct/04/facebook-hits-billion-users-a-month

[2] Mari, M. (2013). Infographic: Twitter The Fastest Growing Social Platform. , . retrieved April 19, 2013, from http://www.globalwebindex.net/twitter-the-fastest-growing-social-platform-infographic/

[3] Holt, R. (2013). Twitter in numbers. , . retrieved April 19, 2013, from http://www.telegraph.co.uk/technology/twitter/9945505/Twitter-in-numbers.html

[4] Smith, A., & Brenner, J. (2012). Twitter use 2012. Pew Internet & American Life Project.

[5] Merchant, R. M., Elmer, S., & Lurie, N. (2011). Integrating social media into emergency-preparedness efforts. New England Journal of Medicine, 365(4), 289-291.

[6] Analytics, P. (2009). Twitter Study–August 2009. San Antonio, TX: Pear Analytics. Available at: www. pearanalytics. com/blog/wp-content/uploads/2010/05/Twitter-Study-August-2009. pdf.

[7] Cataldi, M., Di Caro, L., & Schifanella, C. (2010, July). Emerging topic detection on Twitter based on temporal and social terms evaluation. In Proceedings of the Tenth International Workshop on Multimedia Data Mining (p. 4). ACM.

[8] Li, R., Lei, K. H., Khadiwala, R., & Chang, K. C. (2012, April). TEDAS: a twitter-based event detection and analysis system. In Data Engineering (ICDE), 2012 IEEE 28th International Conference on (pp. 1273-1276). IEEE.

[9] Cameron, M. A., Power, R., Robinson, B., & Yin, J. (2012, April). Emergency situation awareness from twitter for crisis management. In Proceedings of the

21st international conference companion on World Wide Web (pp. 695-698). ACM.

[10] Fung, G. P. C., Yu, J. X., Yu, P. S., & Lu, H. (2005, August). Parameter free bursty events detection in text streams. In Proceedings of the 31st international conference on Very large data bases (pp. 181-192). VLDB Endowment.

[11] Event [Def. 1]. (n.d.). In Oxford Dictionaries Online, Retrieved June 1, 2013, from http://oxforddictionaries.com/definition/english/event?q=event.

[12] Weng, J., & Lee, B. S. (2011, July). Event detection in Twitter. In Proceedings of the 5th International AAAI Conference on Weblogs and Social Media (Vol. 3, No. 4).

[13] Popescu, A. M., & Pennacchiotti, M. (2010, October). Detecting controversial events from twitter. In Proceedings of the 19th ACM international conference on Information and knowledge management (pp. 1873-1876). ACM.

[14] Becker, H., Naaman, M., & Gravano, L. (2011, July). Beyond trending topics: Real-world event identification on twitter. In Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM'11).

[15] Ramos, J. (2003, December). Using tf-idf to determine word relevance in document queries. In Proceedings of the First Instructional Conference on Machine Learning.

[16] Zucchini, W. (2003). Applied smoothing techniques.

[17] Kumaran, G., & Allan, J. (2004, July). Text classification and named entities for new event detection. In Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 297-304). ACM.

[18] Li, C., Sun, A., & Datta, A. (2012, October). Twevent: segment-based event detection from tweets. In Proceedings of the 21st ACM international conference on Information and knowledge management (pp. 155-164). ACM.

[19] Fujisaka, T., Lee, R., & Sumiya, K. (2010, April). Detection of unusually crowded places through micro-blogging sites. In Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on (pp. 467-472). IEEE.

[20] Duong, T. (2012). Local significant differences from non-parametric two-sample tests.

[21] Sakaki T, Okazaki M, Matsuo Y. Tweet Analysis for Real-Time Event Detection and Earthquake Reporting System Development (2012)

[22] Associated Press. http://finance.yahoo.com/news/number-active-users-facebook-over-years-214600186–finance.html (2012).

[23] Smith, T. (2013). Twitter Now The Fastest Growing Social Platform In The World. , . retrieved May 4, 2013, from http://www.globalwebindex.net/twitter-now-the-fastest-growing-social-platform-in-the-world/

[24] Dodds PS, Harris KD, Kloumann IM, Bliss CA, Danforth CM (2011) Temporal Patterns of Happiness and Information in a Global Social Network: Hedonometrics and Twitter. PLoS ONE 6(12): e26752. doi:10.1371/journal.pone.0026752

[25] University of Vermont (2013, April 30). Happiness: There's an app for that; Boston bombings unhappiest day in five yers, new sensor shows. ScienceDaily. Retrieved May 2, 2013, from http://www.sciencedaily.com/releases/2013/04/130430131108.htm

[26] Cheng, A & Evans, M. (2013, June). An In-Depth Look Inside the Twitter World. retrieved May 4, 2013, from http://www.sysomos.com/insidetwitter/

[27] Turlach, B. A. (1993, January). Bandwidth selection in kernel density estimation: A review. In CORE and Institut de Statistique.

[28] Kettunen, K., Kunttu, T., & Järvelin, K. (2005). To stem or lemmatize a highly inflectional language in a probabilistic IR environment?. Journal of Documentation, 61(4), 476-496.

[29] Perkins, J. (2010). Python text processing with NLTK 2.0 cookbook. Packt Pub Limited (pp. 28-29).

[30] Perkins, J. (2010). Python text processing with NLTK 2.0 cookbook. Packt Pub Limited (pp. 26-28).

[31] William J. Conover (1971), Practical Nonparametric Statistics. New York: John Wiley & Sons. Pages 295–301 (one-sample "Kolmogorov" test), 309–314 (two-sample "Smirnov" test).

[32] Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).

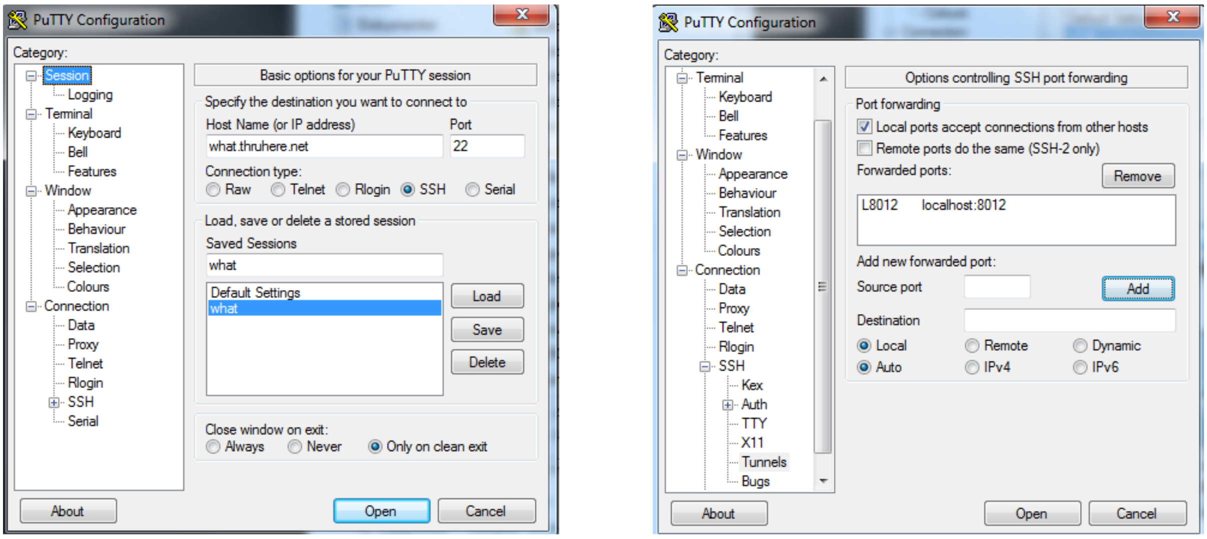
Appendix

How to access Grapher.

To access Grapher a SSH tunnel must be set up. This takes about 5 minutes and is accomplished by following the guide for your operating system.

Windows
Download ”Putty” (putty.exe) from putty.org and start it.

1. Click ”Session”
2. Host Name = ”what.thruhere.net”
3. Port = ”22”
4. Choose ”SSH”
5. Click ”Connection”
6. Click ”SSH”
7. Click ”Tunnels”
8. Select ”Local ports accept connections from other hosts”
9. Sourceport = ”8012”
10. Destination = ”localhost:8012”
11. Choose ”Local”
12. Choose ”Auto”
13. Click ”Add” and you should see the rightmost image of figure A1.



Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address): what.thruhere.net
Port: 22
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial
Load, save or delete a stored session
Saved Sessions: what
Default Settings
what
Close window on exit: ○ Always ○ Never ● Only on clean exit

Options controlling SSH port forwarding
Port forwarding
☑ Local ports accept connections from other hosts
□ Remote ports do the same (SSH-2 only)
Forwarded ports:
L8012 localhost:8012
Add new forwarded port:
Source port
Destination
● Local ○ Remote ○ Dynamic
● Auto ○ IPv4 ○ IPv6

Figure A1: Connection settings.

14. Click ”Open”
15. Click ”Yes” on the pop-up asking if you want to trust the server.
16. A command line pops up. Type in ”grapher-user” as user name.
17. Type in ”guest2013guest” as password. When you are logged in the tunnel is open as long as the command line window stays open.
18. Use Chrome and go to ”http://127.0.0.1:8012/test/choose.html” and you are at the overwiew page. It shows all performed tests. Have fun.

Mac OS

There are two methods of setting up an SSH tunnel, using the terminal or a GUI tool.

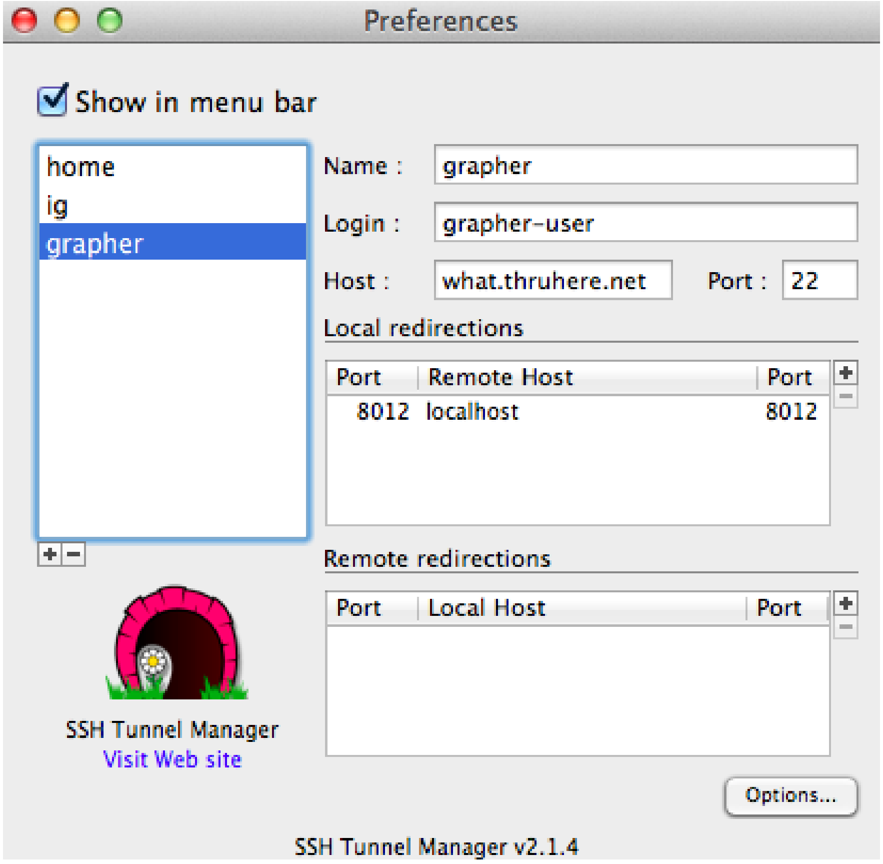
Terminal

1. Type: ”ssh -L 8012:localhost:8012 grapher-user@what.thruhere.net -p 22”
2. password is ”guest2013guest”

GUI Tool

Install ”SSH Tunnel Manager” from App store.

1. Open ”SSH Tunnel Manager” and click on ”Configuration”.
2. Click on ”+” and do the same as in the following screenshot.



☑ Show in menu bar

Name : grapher

Login : grapher-user

Host : what.thruhere.net Port : 22

Local redirections

Port	Remote Host	Port
8012	localhost	8012

Remote redirections

Port	Local Host	Port

Figure A2: Connection settings.

3. Name = ”grapher”
4. Login = ”grapher-user”
5. Host = ”what.thruhere.net”
6. Port = ”22”
7. Click ”+” on Local redirections
8. (left) Port = ”8012”
9. Remote Host = ”localhost”
10. (right) Port = ”8012”

11. To save click on ”grapher” on the left side.
12. Close Configuration.

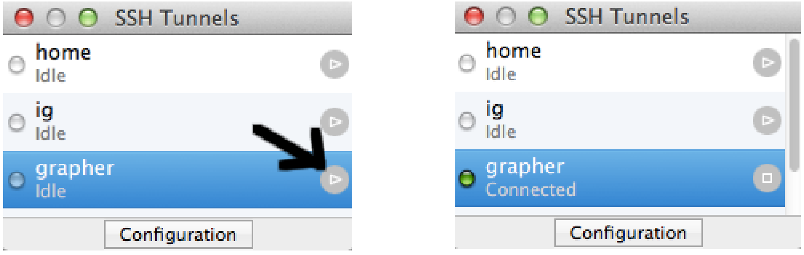


Figure A3: (1) Click on the symbol where the arrow is pointing. (2) Connected.

13. You will maybe be asked if you want to trust the host. Click ”Yes”.
14. You will be prompted for password which is ”guest2013guest” without the quotes.
15. Use preferably Chrome and go to ”http://127.0.0.1:8012/test/choose.html” and you are at the overwiew page. It shows all performed tests. Have fun.

Ubuntu

The instructions for Mac also applies here except that the GUI tool called gSTM and is installed by typing

”sudo apt-get install gstm” into the terminal