Understanding Traffic Events by Enriching Traffic Data with Geosocial Data

Master's Thesis, 28-08-2018

Bas de Böck

Understanding Traffic Events by Enriching Traffic Data with Geosocial Data

THESIS

Submitted in the partial fulfillment of
The requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE
TRACK SOFTWARE TECHNOLOGY

by

Bas de Böck
Born in Vlissingen, the Netherlands

Web Information Systems
Department of Software Technology
Faculty, EEMCS, Delft University of Technology
Delft, the Netherlands
http://wis.ewi.tudelft.nl

Understanding Traffic Events by Enriching Traffic Data with Geosocial Data

Author: Bas de Böck
Student ID: 4366174
Email: b.debock@student.tudelft.nl

Abstract

Non-recurrent traffic events, consisting of events of an unpredictable nature such as incidents and vehicle breakdowns, can either directly or indirectly influence road traffic. A better understanding of these events could prove beneficial towards improving a multitude of facets concerning the management of the Dutch road network. Traditional traffic event detection, based on significant changes in traffic flow/speed characteristics, is often limited by sparse road sensor coverage. More importantly, traditional detection methods are unable to categorize and describe traffic events.

The aim of this study is to explore to which extent geosocial data (e.g., data from Twitter and Waze) could enrich traditional traffic data (e.g., traffic speed/flow data), in order to improve the detection, categorization, and description of traffic events in the Netherlands. In order to achieve this, a pipeline was designed for extracting knowledge on traffic events from geosocial data sources. We collected geosocial data from Twitter, Waze, and TomTom and used traffic data provided by DiTTLab. We specifically focused on reports by real road users, which we define as natural persons that report on their own account, therefore excluding all legal person entity accounts such as public/private organizations, and bots. A machine learning approach was applied to automatically classify tweets as either traffic event related or not. In order to categorize tweets into a traffic event category, a rule-based traffic domain annotator was created. Additionally, a geocoding method to link tweets to a geographic location was developed. As Waze and TomTom event reports are classified and geocoded by default, we could cluster these reports together with the processed tweets based on their categorical, spatial and temporal extent into a combined traffic event. These combined traffic event reports were then linked to traffic data, based on corresponding spatial and temporal aspects. In order to present the collected data, a web-based interactive map application was built.

This methodology was applied to data collected over the period from 05-12-2017 to 17-02-2018. From the set of collected tweets approximately 6.71% proved traffic event related. Based on a linear support vector machine classification model we achieved an average f1-score of 0.95 and an accuracy of 0.954, for detecting traffic event-related tweets. The rule-based traffic domain annotator showed an average f1-score of 0.874, and an accuracy of 0.964. The geocoding method proved able to geocode tweets to a location that covers all place indicators in a tweet in 86% of the evaluated cases. The remaining 14% of the tweets either got geocoded to a part of relevant indicators or to no relevant indicators at all. Our clustering approach is able to cluster 39.61% of the event reports into a traffic event report cluster consisting out of more than one event report, from which 48.66% could be linked to traffic data.

All in all, based on the achieved results, this work shows that geosocial data can be used to enrich traffic data towards the improvement of the detection, categorization, and description of non-recurrent traffic events.

Thesis Committee:

Prof. dr. ir. G.J.P.M. Houben, Faculty EEMCS[1], TU Delft
Dr. ir. A. Bozzon, Faculty EEMCS, TU Delft
Prof. dr. Ir. J.W.C. van Lint, Faculty CEG[2], TU Delft
Dr. A. Psyllidis, Faculty EEMCS, TU Delft

[1] Electrical Engineering, Mathematics and Computer Science

[2] Civil Engineering and Geosciences

Preface

Before you lies the thesis “Understanding Traffic Events by Enriching Traffic Data with Geosocial Data”. It has been written to fulfill the graduation requirements of the MSc programme in Computer Science at the Delft University of Technology. I was engaged in researching and writing this thesis from May 2017 to September 2018. This project was undertaken at the request of the Web Information Systems research group in the Software Technology department.

Throughout the challenging work on my master’s thesis I have learned a great number of new useful technologies, and gained experience in the art of performing scientific research. I am glad I was able to finish this thesis in a satisfying matter. However, this work would not have been possible without the valuable advice and support of the following number of persons.

First, I would like to thank my supervisors Alessandro Bozzon and Achilleas Psyllidis. Throughout the project, I have always been able to ask you for advice and you helped me with supportive criticism and suggestions. Second, I would like to thank Panchamy Krishnakumari who acted as my contact person at the DiTTLab, and Hans van Lint for providing advice on the traffic research part of my thesis.

Third, my thanks go out to Alexander Grooff and Jan Zegers for teaming up through the entirety of the master track. I have always been able to count on your advice and assistance when needed, and discussing my work with you proved invaluable.

Lastly, I would like to thank my father Alex and brother Kees for their support and love throughout the years. Also, I would like to dedicate this thesis in memory of my lovely mother Ciska, who has always supported and inspired me to pursue my study in computer science.

Bas de Böck
Vlissingen, the Netherlands
August, 2018

Contents

List of Figures

List of Tables

1 Introduction

Traffic events that cause road congestions are a daily phenomenon on the road network of the Netherlands. These traffic events can be divided into seven root causes: capacity, work zones, traffic control devices, fluctuations in normal traffic, traffic incidents, weather, and special events (Systematics & others, 2005). Capacity is the maximum amount of traffic that a highway is able to cope with, determined by factors such as the number and the width of lanes, shoulders, and interchanges. Traffic incidents cause the disruption of the normal traffic flow by road blockages which are the result of, e.g., vehicular crashes and breakdowns. Work zones are planned construction activities that influence the normal physical state of the road by, e.g., reducing the number of available or width of travel lanes. Weather could cause the normal driving behavior of drivers to change by, e.g., reducing the travel speed of vehicles due to icy roads or causing impaired vision due to heavy rain. Traffic control devices such as railway and bridge control systems could cause a change towards the typical traffic flow. Special events such as sports matches and festivals could also influence the typical traffic flow. Fluctuations in normal traffic are caused by varying traffic demand volumes while having roads with a fixed capacity. Additionally, these events can influence each other, e.g., bad weather could lead to car crashes.

Traffic events can be divided in those with recurrent predictable causes and non-recurrent unpredictable causes. These non-recurrent events, consisting of traffic incidents, unplanned roadworks, weather, and special events, are a critical but difficult problem to detect, categorize and describe. These traffic events are of interest to a multiple of different stakeholders, which can be divided into three groups. First, stakeholders that are interested in traffic event detection, which takes place in the period moving towards the event. Second, stakeholders that have to apply traffic event management during the event. Third, stakeholders that apply a historical offline analysis on the traffic events and research the causes behind and statistics on the events. These organizations have multiple tools at their disposal to achieve their tasks, such as roadside detection sensors and cameras. In addition, they have the availability of information provided by traffic inspectors and emergency services. However, the problem with the current tooling is that their capabilities to detect, categorize and describe traffic events are limited and flawed. On the one hand, the detection part is not always reliable when it comes to events that are too small to have an impact on the traffic at that moment (Stephanedes & Chassiakos, 1993). However, this event could cause another event, later on, that is measurable and could, therefore, have been used as a predictor. Take for example traffic that has to evade road debris (tree branches, sharp objects, auto parts), which stays undetected at first but could cause future accidents due to flat tires or sudden dangerous evasion maneuvers. The underlying problem behind this is that these events are automatically detected with the use of algorithms, which assume that traffic events immediately cause a change in the traffic flow and speed characteristics. The data on which these algorithms depend is provided by traffic sensors which are limited in amount and cannot cover every point on the road. Besides, these algorithms are road-type dependent, algorithms that can be applied on freeways, are often not suitable for arterial situations which are much more complex. On the other hand, the description and categorization of traffic events are limited and inconsistent as they depend on the observations and deductions made by the instances arriving after the event occurred, as the traffic data itself does not contain the semantics to achieve this.

This thesis explores the extent to which new forms of geosocial data (e.g., data from Twitter and Waze) could enrich traditional traffic data (e.g., traffic speed and flow data from roadside detection sensors), in order to improve the detection, categorization, and description of traffic events. Geosocial data refers to data created by individuals that is voluntarily and knowingly shared on online platforms, and contains some sort of geographic property. Social media platforms as Twitter[3] and Instagram[4] have become a widely used tool to extract geosocial data within the Web Information Systems research field. Research by der Veer, Sival, and van der Meer (2017) shows that Instagram is used by 3.2 million Dutch users of which 1.5 million are daily users and 2.6 million Dutch Twitter users of which are 871,000 daily users. As traffic is a part of almost everyone's daily life, the assumption can be made that this also reflects on people their online social life, resulting in tweets and Instagram posts about traffic events. Additionally, less general but more traffic specialized social platforms such as Waze[5] could be used. Waze is a community-based traffic and navigation app, which enables users to share traffic event reports. These traffic event reports can be seen as categorized traffic based geosocial posts, and therefore could contribute towards the enrichment of traffic data.

In recent years, limited research has been performed on how geosocial data can be used to detect and describe (traffic) events. Most of the research either focus on how geosocial data can be utilized to derive new and improved traffic event detection, categorization, and description approaches (e.g., Schulz, Ristoski, and Paulheim (2013), D'Andrea, Ducange, Lazzerini, and Marcelloni (2015), and Gu, Qian, and Chen (2016)). However, using only one geosocial data source comes with a number of disadvantages:

1. Reliability of the category assigned to a detected traffic event: Did the user use distinctive enough words to derive the correct event category?
2. Reliability of the spatial aspects of the detected traffic event: Was the user really on the location of the event at the time of posting the geosocial post, or did he post about an event he read or heard about? And did the user use accurate enough locational words to be able to derive the correct event location?
3. Reliability of the temporal aspects of the detected traffic event: Was the geosocial post composed directly after the traffic event, or did it refer to a historical or future event?

These disadvantages would mostly be non-existent if the number of tweets that refer to a single event would always be of a high quantity. That way, tweets could be aggregated together to improve the reliability of the detected event. However, research shows that the ratio of the number of traffic event-related tweets per location and time range proves to be very low, and thus additional data sources are needed to compensate for this.

[3] Twitter.com
[4] Instagram.com
[5] Waze.com

In the last few years research towards the combination of traffic and geosocial data has been conducted (e.g., Daly, Lecue, and Bicer (2013) and Giridhar, Amin, Abdelzaher, Wang, Kaplan, George, and Ganti (2017)). The limited amount of research shows that there is still a gap to fill by combining multiple (new) geosocial data and traffic data sources in order to improve upon existing traffic event detection, categorization, and description approaches.

1.1 Research Objectives

The main goal of this thesis is to investigate how geosocial and traffic data relate to each other and how this relationship can be utilized to improve upon the current state of the art traffic event detection, categorization, and description approaches. The main research question is therefore defined as follows:

***RQ**: To what extent can geosocial data enrich traffic data to improve the detection, categorization, and description of non-recurrent traffic events?*

In order to answer this main research question, the following research sub-questions are posed:

***RQ1:** What is the current state of the art regarding non-recurrent traffic event detection, categorization, and description by using traffic data and geosocial data, individually or combined?*

***RQ2:** How can non-recurrent traffic event-related geosocial posts be detected?*

***RQ3:** How can detected non-recurrent traffic event-related geosocial posts be categorized by event type?*

***RQ4:** How can categorized geosocial posts be used to describe non-recurrent traffic events?*

***RQ5:** How to develop a software system that is able to perform the detection, categorization, and description of non-recurrent traffic events?*

1.2 Methods

To address research question 1, we have to understand the decisions, demarcations, conclusions and future work directions that have been made in previous related scientific work. Related work is reviewed based on the data source types used in their research: traffic data sources, geosocial data sources, and a combination of the two. This way choices can be taken towards the selection and extension of certain data collection, pre-processing, feature engineering, classification, categorization, linking, aggregation, and visualization approaches. This literature study provides the foundation for the approach taken in the remaining research questions.

To address research question 2, a data retrieval process is set up for the selected geosocial data sources Twitter and Instagram. In this approach, we only focus on Dutch data that is related to the road network of the Netherlands. The Twitter REST API is used to collect tweets based on an adaptively created traffic event-related keyword set. The main goal to achieve when collecting traffic event-related tweets, is to create a keyword set that maximizes the percentage of traffic event-related tweets over all acquired tweets and maximizes the amount of acquired traffic event-related tweets in the pool (all Dutch tweets in the Netherlands within a specific time range). Furthermore, a filtering method is applied to filter out the majority of non-real road user accounts. We define a real road user as follows: a natural person that tweets on his/her own account, therefore excluding all legal person entity accounts such as public organizations (government agencies, police, and infrastructure agencies), private organizations, and bots. This same keyword set is used to collect Instagram posts by using the Instagram API Platform. However, initial experiments show that Instagram provides extremely low amounts of traffic event-related posts. Therefore, we make a well-substantiated decision to no longer include Instagram in our setup. Besides Twitter data, data from Waze is collected by extracting a GeoRSS web feed from its web-based live map[6]. This way all traffic event data within a bounding box covering the entirety of the Netherlands is collected every 2 minutes. Furthermore, data from TomTom[7] is collected through their Online Traffic Incidents API. Here, also a bounding box covering the entirety of the Netherlands is used to collect TomTom data every 2 minutes.

Next, as preparation for the creation of a traffic event classifier, part of the tweets from the collected Twitter dataset are manually labeled as either traffic event-related (TE) or non-traffic event-related (NTE). Next, pre-processing is applied to the TE tweets by applying tokenization and stop word removal. Subsequently, feature selection is applied to the Twitter data. Features based on the following characteristics are used: term frequency-inverse document frequency (TF-IDF) weighting, bag of words/n-grams, syntactic features (exclamation/question marks, emoticons, and total number of capital characters), and traffic domain categories based on our custom created rule-based traffic domain annotator. In order to automatically determine if a tweet is related to a traffic event, a classifier is applied. A mix of different machine learning classification algorithms (Support vector machine and Naïve Bayes), features and dataset sizes are explored to achieve the best classifier for identifying traffic event-related tweets. In order to estimate the performance of the model, 10-fold cross-validation is applied on the Twitter training dataset. To measure the performance of the

[6] waze.com/livemap

[7] https://developer.tomtom.com/online-traffic

classification approaches the following metrics are reported: accuracy, precision, recall, f1-score, and the area under the curve of receiver operating characteristic (ROC AUC). This provides us with an overview that evaluates all features and their combinations with different classifiers. The classifier with the best performance is used to classify the tweet in the Twitter dataset. Contrary to data collected from Twitter, data from Waze and TomTom have little pre-processing needs. Attributes that do not contain any descriptive value will be omitted and attribute terms are made uniform between the datasets. Besides, Waze and TomTom reports are by definition traffic event-related and therefore do not need to be classified by a classifier.

In order to address research question 3, a rule-based traffic domain annotator is created. This annotator is used for extracting relevant traffic domain information from tweet text data. This allows for the automatic categorization of a tweet into one of 27 distinct traffic domain categories (e.g., categories that describe road users, spatial features and traffic events) of which 13 are related towards traffic events (e.g., traffic jam, accident and roadworks). The traffic domain categories are based upon the event categories from Waze, TomTom, the categories in the police accident reporting dossier (BRON)[8], and acquired knowledge from reviewing literature and annotating tweets. The annotator uses a Backus-Naur form (BNF) grammar, allowing for partial matching of tokens, while using a combination of place names, temporal expressions, traffic domain knowledge, and lexical pattern dictionaries.

To address research question 4, multiple data sources (Waze, TomTom, and a data source provided by Delft integrated Traffic & Travel Laboratory (DiTTLab)[9]) are combined with traffic event-related annotated and categorized Twitter data, to describe traffic events. Our description approach consists of the following three stages:

1. Geocoding: tweets have to be linked to a geographic location, also known as geocoding. Approximately, only 1% of the tweets contains a geotag. Therefore, to identify the location of the other 99%, a location linking method is developed. This method utilizes the rule-based traffic domain annotator, which is able to annotate a multitude of spatial indicators from tweets. Based on the location category the Google Places API[10], Google Directions API[11], or custom created road database (consisting of road numbers and mile markers, with their respective coordinates) is queried to obtain a location. Tweets can contain multiple spatial indicators bringing the following challenges: contradiction/confirmation of each other, relation to different forms and scales, and ambiguity. Therefore, a model is designed to mitigate these challenges, by computing the intersections of spatial indicators in a tweet.
2. Clustering of traffic event reports: in this step geocoded tweets are clustered together with other related tweets, Waze and TomTom event reports, eventually forming a described traffic event. First, a traffic event described by a newly incoming traffic event report (e.g., a tweet, Waze, or TomTom report) is compared to a previously reported traffic event report cluster. Matching is based on a rule-based approach, in which a rule specifies the categorical, spatial and temporal extent, used to assert if the new traffic event report should be part of an existing traffic event cluster. A traffic event report is

[8] https://www.rijkswaterstaat.nl/apps/geoservices/geodata/dmc/bron/

[9] dittlab.tudelft.nl/

[10] https://developers.google.com/places/web-service/

[11] https://developers.google.com/maps/documentation/directions/

added to an existing traffic event cluster when there is a match, otherwise a new traffic event cluster is created based on that traffic event.

3. Linking to traffic data: based on the clustering results, traffic event reports are formed based on geosocial data. However, as it is not our goal to map traffic events based on geosocial data alone, but to enrich traffic data, an additional approach is taken. For this purpose, a traffic data set from DiTTLab is used, containing interpolated speed and flow values per 100m segments for each motorway (A-roads) in the Netherlands. This data could be used as a source for traffic event detection algorithms. However, as traffic event detection algorithms greatly depend on the type and properties of the road, it is not feasible to implement this for every highway. Besides, it would fall out of the scope of this research. As stated before this traffic data on its own does not tell anything about the kind of traffic event that has happened, is happening or will happen. Besides, a traffic event can also happen without influencing the traffic speed and flow, making this data in some cases on its own more or less useless. Therefore, a method is created that links traffic events to traffic speed and flow data, based on temporal and locational similarity.

To address the final research question 5, the parts developed in the answering of research questions 2 to 4 are combined into a pipeline. This pipeline is able to perform the detection, categorization, and description of traffic events, and forms the back-end of the system. To present the collected data to the user a web-based interactive map application is build. This application enables the user to view the traffic events and their descriptions on an interactive map. Besides, a user will be able to filter traffic events based on event category, time range and location.

1.3 Contributions

The main contributions made in this thesis are fivefold:

1. A literature survey on state of the art techniques regarding non-recurrent traffic event detection, categorization, and description by using either traffic data, geosocial data or the combination.
2. A model that combines multiple geosocial data sources to enrich traffic data to improve the detection, categorization, and description of traffic events. We extend upon previous related work by combining detection, categorization and description methods with each other, instead of focusing on one in particular. Additionally, instead of focusing on a single data source, we combine multiple social and traffic based data sources including Twitter, Waze, TomTom, and DiTTLab. We specifically focus only on geosocial posts by real road users, instead of a mix of posts by real road users, news agencies, bots etc. Lastly, we focus on Dutch geosocial data, which has not been researched before besides in the study by Dokter (2015).
3. A dataset containing annotated tweets as well as Waze, TomTom and traffic data. This dataset can be used in future studies regarding this topic. This dataset could prove useful for any future research regarding this topic.
4. Patterns and insights into the properties of the different data sources, and their relation towards getting a better understanding of traffic events.
5. A software system named SocialTerraffic. This system consists of two parts. First, a pipeline that is able to perform the detection, categorization and description of traffic events, and store this data in a database. Second, a web-based interactive map that uses the collected and processed data from the pipeline to present traffic events to a user. This application enables a user to filter traffic events based on event category, date range and location. Additionally, the application is able to generate speed/flow charts based on traffic data related to a traffic event.

1.4 Thesis Outline

The remainder of this thesis is organized as follows. 2 introduces the scientific background of this thesis and discusses related work on how geosocial data can be used to detect, categorize and describe traffic events. The experiment design, in which the approaches and methodologies used in this work are described, takes place in 3. 4 describes the implementation and results of the designed experiments. In 5 we discuss and interpret the outcomes of our experiments. Finally, in 6 a conclusion of this thesis is provided, and opportunities for future research are explored.

2 Background and Related Work

In this chapter, previous work regarding traffic event detection, categorization and description is discussed and compared. This is done in order to show how this thesis builds upon and extends from previous research on similar topics. Therefore, this chapter should provide an answer to the first research sub-question:

- ***RQ1:*** *What is the current state of the art regarding non-recurrent traffic event detection, categorization, and description by using traffic data and geosocial data, individually or combined?*

In order to answer this question, this chapter is divided into three sections based on the data source types used in the research:

1. Traffic data sources, mostly used in Transport & Planning research field.
2. Geosocial data sources, mostly used in Computer Science research field.
3. A combination of traffic and geosocial data sources used in the Transport & Planning and Computer Science research fields.

As we focus specifically on traffic event detection, categorization and description, some research using a combination of traffic and geosocial data would fall out of scope. This research could, however, contain valuable information for our research. Therefore an additional section is devoted to possible relevant topics including traffic prediction, traffic and geosocial data correlation, and traffic congestion monitoring.

2.1 Traffic Data

Research within the Transport & Planning field mainly focusses on the detection of traffic events. These traffic event detection systems can be divided into a data collection and a data processing part. Data collection describes the measurement techniques used to obtain the traffic data. These technologies can be divided into roadway-based and probe-based sensors.

2.1.1 Data Collection

Roadway-based sensors are integrated into the roadway infrastructure system, being embedded in the roads, at the side of the road or over the road. They provide traffic information from the passing vehicles over a fixed point or short segment. Therefore, the advantages of this system are that traffic volumes can be measured directly, while the traffic speed can be inferred from the traffic volume based on an average vehicle length. A disadvantage, however, is that the quality of travel time measurements is dependent on the density of the sensor network. Other disadvantages come with high deployment costs and intensive maintenance costs (Young, 2007). Roadway-based sensors can be divided into magnetic (piezoelectric detectors, active/passive magnetic detectors, inductive loop detectors (ILD)), range detectors (infrared detectors, ultrasonic detectors, microwave/millimeter wave radar, passive acoustic detector arrays, photoelectric detectors, spread-spectrum wideband radar), and image sensing detectors (video image processors (VIP)) (Kon, 1998).

Probe-based sensors are carried by vehicles instead of being part of the underlying road infrastructure. This allows for direct travel time measurements and increases the traffic flow coverage space. However, the quality of these sensors highly depends on the number of vehicles equipped with a sensor. Probe-based sensors can be categorized into: cell phone probes (by using signaling information or GPS), automated vehicle location (AVL) services by in-car systems that monitor the GPS of the car, and automatic vehicle identification (AVI) systems that use an in-vehicle tag or transponder to wirelessly communicate with a roadside unit to identify the vehicle location (Young, 2007).

2.1.2 Data Processing

Data processing uses traffic detection and classification algorithms by analyzing traffic data obtained from the data collection sensors. These algorithms can be classified by the traffic data they rely on resulting in roadway-based and probe-based algorithms.

2.1.2.1 Roadway-based Algorithms

Roadway-based algorithms can be divided into the following five main categories: comparative, statistical, time series, traffic modeling, and image processing algorithms.

1. Comparative algorithms compare the traffic data to a pre-defined threshold value. This category includes algorithms based on decision trees which assume that traffic events cause significant increases in upstream occupancy (the percentage of time the detection zone of a detector is occupied by some vehicle) while reducing downstream occupancy (Tignor & Payne, 1977). And algorithms based on pattern recognition, which compare historically estimated vehicle speeds for particular traffic patterns with pre-established thresholds.
2. Statistical algorithms use statistically determined traffic characteristics to find deviations in the traffic data. Dudek, Messer, and Nuckles (1974) propose a method based on the standard normal deviate to find sudden changes in traffic data that could suggest occurrences of traffic events. Levin and Krause (1978) propose a method based on Bayesian statistical techniques that use the relative distances of occupancies from comparative algorithms to compute if an event signal is caused by a lane-blocking event.
3. Time series algorithms compare the traffic data to time series models that contain historically predictable traffic patterns. The commonly used techniques are the autoregressive integrated moving-average (ARIMA) model and the high occupancy (HIOCC) algorithm (Ahmed and Cook, 1979).
4. Traffic modeling algorithms use traffic flow theory to develop models that describe traffic behavior when a traffic event occurs. One common technique is the dynamic model that uses speed and flow density relationships to apply traffic flow models to capture the dynamic nature of traffic (Willsky, Chow, Gershwin, Greene, Houpt, & Kurkjian, 1980). Another technique is based on the catastrophe theory model, which is based on the assumption that when a state changes from congested to uncongested, the traffic speed changes sharply while flow and occupancy change smoothly (Forbes and Hall, 1990).
5. Image processing algorithms process surveillance video footage and use this processed data to provide traffic measures or to detect traffic events. Li and Porikli (2004) propose a mechanism to detect highway traffic events by extracting features directly from the videos, based on the Gaussian Mixture Hidden Markov Model framework. Additionally, they classify the traffic events into six traffic patterns (heavy congestion, high density

with low speed, high density with high speed, low density with high speed, low density with low speed, and vacancy) by using the Viterbi algorithm to determine the most likely traffic condition. Ikeda, Kaneko, Matsuo, and Tsuji (1999) performed a feasibility study towards detecting abnormal traffic events by using image processing technologies. In this study, they automatically detect traffic events and classify them into the following four categories: stopped vehicle, slow vehicle, and fallen object. Aköz and Karsligil (2014) propose a detection and classification mechanism by using traffic event severities at intersections. By clustering vehicle trajectories the system learns common traffic flow patterns which are used to detect abnormalities. These events are then classified into low and high severity classes.

The major drawback of roadway-based algorithms (with the exception of image processing algorithms) is that the data source they use is easily corrupted by noise, which therefore should be filtered out before use. With a noisy dataset traffic event patterns may not be detected easily, and fluctuations could be misinterpreted as events. As a result, only severe traffic events can be detected with these kinds of algorithms (Stephanedes & Chassiakos, 1993).

2.1.2.2 Probe-based Algorithms

Probe-based algorithms can be divided by their most commonly used probe sensor technology.

- AVL sensor-based algorithms: Sethi, Bhandari, Koppelman, and Schofer (1995) propose a travel time algorithm that uses the event link and adjacent upstream link and average speed measures, based on GPS data. Sermons and Koppelman (1996) use GPS based algorithms that are based on the assumption that vehicles passing traffic events have higher travel times and a higher coefficient of speed variation. Kamran and Haas (2007) combine dynamic road segmentation logic with individual vehicle behavior identification methods based on GPS data to detect traffic events.
- AVI sensor-based algorithms: Parkany and Bernstein (1995) discuss three algorithms (headways, lane switches, lane-monitoring algorithm) to use vehicle-to-roadside communication sensors in the form of electronic toll transponders. Niver, Mouskos, Batz, and Dwyer (2000) use the statistical travel time comparison between the TRANSMIT traffic surveillance and incident detection system (based on E-ZPass electronic toll collection tags) and probe reports.

2.1.2.3 Freeway vs Arterial Algorithms

Most of the described road-way and probe-based algorithms are only applicable on freeways and are not directly applicable towards arterials (high-capacity urban roads). This has a number of reasons. First, the variation of traffic on arterials is more complex and varied than on freeways. Second, arterials are susceptible to certain events that could signal false traffic events when applying freeway based algorithms, e.g., events caused by bus stops, parking maneuvers, traffic leaving and entering from side streets, traffic signal control (Ivan, Schofer, Koppelman, & Massone, 1995). Due to these additional difficulties, research on arterials has only caught the interest of researchers in the last number of years, while research towards freeways has been going on for the last few decades. An example of these arterial algorithms, which do not fall in the previously discussed freeway algorithms are the fuzzy logic-based algorithms. These are based on human-interference-oriented AI techniques and used for models that operate in real-time and deal with uncertainty and need approximate reasoning (Yaguang and Anke, 2006). Hawas (2007) uses such a fuzzy-based system for traffic event detection at intersections in urban street networks. He developed a simulation-based methodology and tested its logic under various real-world scenarios. Additionally, show that a combination of SVM and fuzzy logic-based on volume and occupancy data from fixed detectors can be used to detect traffic events on urban arterial streets.

2.1.2.4 Traffic Event Detection Key Points

The most important theoretical key point on the detection of traffic events is that traffic is a spatiotemporal problem. This means that in order to detect an event by using just traffic data a couple of things are needed:

1. There should be some sort of congestion.
2. The outflow out of the congestion should be (much) lower than the (expected) capacity.
3. The congestion is often homogeneous with very low speed and flow values.
4. The congestion takes place at locations without a known bottleneck (known bottlenecks include ramps, bridges/tunnels, weaving sections etc.).

Such cases of congestion can be translated to heat map charts depicting the speed/flow values on a lane over time. Figure 2-1 and Figure 2-2, depict a heat map of traffic speed/flow over a 500 meters road lane segment, where the time is placed on the x-axis, the distance (in km) on the y-axis, and speed (km/h) /flow (vehicle/hour/lane) on the z-axis. These figures provide an example of a case where the congestion is homogenous with very low speed and flow values represented by the red segment in the traffic speed and blue segment in the traffic flow heat map. Such congestion could thus indicate something is going on, however this traffic data provides no context on the type of event. Traffic data can help to predict the traffic consequences of an event, but only if there can be made a prediction on how long the event itself will last.

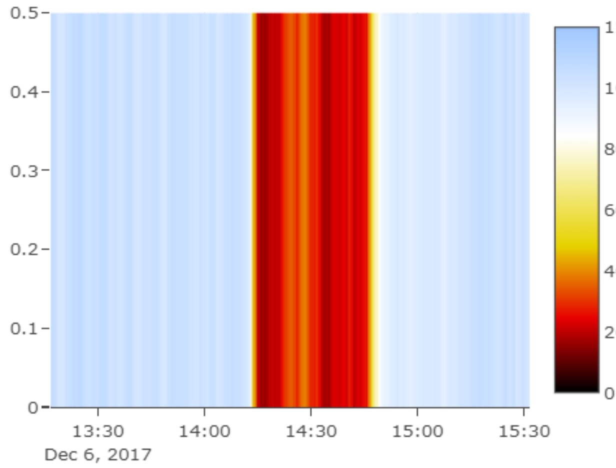


Figure 2-1: Heat map of traffic speed (km/h)

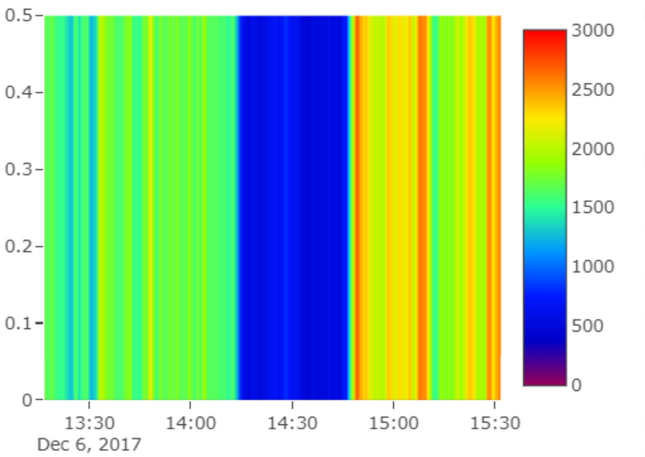


Figure 2-2: Heat map of traffic flow (vehicle/hour/lane)

2.1.3 Traffic Data Evaluation

Research within the Transport & Planning field shows us that there are numerous algorithms to detect traffic events based on roadway-based and probe-based sensors. However, these algorithms do not always provide reliable and constant results due to three factors. First, the quality of measurements from roadway-based sensors depends on the density of the sensor network. The same is true for probe-based sensors as they depend on the number of vehicles equipped with sensors. Second, data sources are easily corrupted by noise which makes detection of less severe traffic events near to impossible. Third, the difference between a freeway and arterial (urban) traffic data. Algorithms that can be applied on freeways are often not suitable for arterial situations which are much more complex. Beside these traffic event detection approaches, no methods for traffic event categorization and description could be found. This makes sense, as traffic data misses the semantics to derive these methods. In conclusion, this works shows that traffic data sources on their own can only be used for traffic event detection on a specific selection of roads. This shows that there lays an opportunity for this thesis to enrich this traffic data by adding a traffic event categorization and description approach. Additionally, by enriching the traffic data source itself we can show that our contribution is suitable for all algorithms that are based on such traffic data sources.

2.2 Geosocial Data

Contrary to the previously discussed Transport and Planning field where the focus laid on traffic event detection, research within the Computer Science is more evenly focused on traffic event detection, categorization, and description. Each evaluated research paper contains at least one of these three categories, which will get discussed in depth. For the traffic event detection part, an overview is given of the used geosocial data and the data collection, preprocessing and feature engineering and machine learning techniques that are used to create a binary traffic event classifier. For the traffic event classifier part, techniques are discussed that are being used to classify a geosocial post into a traffic event categories. For the traffic event description part, we look at how (categorized) geosocial data is used to infer a traffic event, which includes linking, aggregation and visualization strategies.

2.2.1 Geosocial Data based Related Work

Wanichayapong, Pruthipunyaskul, Pattara-Atikom, and Chaovalit (2011) propose an extraction and classification technique for traffic information. They collect Thai tweets by using a query of two traffic-related keywords on the Twitter REST API. The resulted tweet set is tokenized and the tokens get parsed into four dictionary categories: “Place” (names of roads, places, crossroads, and alleys), “Verb” (traffic conditions, e.g., traffic jam), “Ban” (vulgarity, profanity, and question words), and “Preposition” (road directions). A tweet is considered traffic-related if it contains at least a word in the “Place” and “Verb” categories and does not contain a “Ban” category word. This dictionary and rule-based detection method are able to detect traffic event-related tweets with an accuracy of 91.75%, precision of 91.39%, and recall of 87.53%. Based on a dataset of 1249 tweets, consisting for 21% of traffic information center based tweets and for 79% of individual users based tweets.

In addition to the proposed traffic event detector, a limited traffic event description method is proposed that links tweets classified as traffic event-related directly to a possible traffic event location. In this method, the start and end point of the possible traffic event get derived by finding “Preposition” and “Place” combinations in the tweet. These points are used to find a corresponding location, by using them to query the place dictionary of the Ministry of transportation Thailand, or to query Google geocoding (road segment-based). If a tweet did not contain a start or endpoint, the road keyword is used to determine a location (road point-based). This method is able to classify traffic event-related tweets with 76.85% accuracy, 62.77% precision, and 95.36% recall in the road segment category, based on a dataset of 3311 tweets. And 93.23% accuracy, 81.72% precision, 92.20% recall in the road point category, based on a dataset of 2942 tweets.

Ribeiro Jr, Davis Jr, Oliveira, Meira Jr, Gonçalves, and Pappa (2012) propose a real-time Twitter-based traffic event and condition identification method for the city Belo Horizonte. Portuguese tweets are collected by following ten influential accounts that report on traffic situations. Traffic event detection is performed based on a static dictionary list of frequently used traffic event terms in tweets. They do not apply any way of automated traffic event categorization. Instead, they focus on location detection and mapping based on tweets. For this purpose, a geographic dictionary is formed of thoroughfare names and segments, and street crossings with their related thoroughfares. Additionally, this dictionary also provides for common traffic abbreviations. By using exact string matching on words in tweets combined with thoroughfare types, a traffic event location is determined. This location is then refined by using fuzzy string matching to find names of related roads.

In the work by Li, Lei, Kwadiwala, and Chang (2012) a Twitter-based event (crime and disaster-related, including traffic accidents) detection and analysis system are proposed. Data is collected by using a seed keyword set to query the Twitter REST API. From the resulted tweet set, word bi-grams are extracted as possible candidates to add to the keyword set. They get added to the keyword set if the ratio between event-related and non-event related is positive. These new keywords get validated by comparing the ratio of newly retrieved event-related tweets to newly retrieved non-event related tweets. If this proves positive the keyword set gets added to the initial keyword set. This process is repeated until no new keywords can be found. Their method uses a combination of Twitter-specific features (links, hashtags, and mentions) and event-specific features (time, location, and numbers). Based on these features they train a classification model which tested to have an accuracy of 80%. However, no specifics on feature extraction or the type of classifier are given.

Positively classified tweets are indexed by a text search engine and stored in a database, which is used to answer real-time queries and provide visualizations. A clustering model is used to group similar tweets into similar geographic regions and temporal ranges. However, no further details were provided on the workings of this clustering model. Another event description part includes the ranking of tweets according to their importance, which is done based on content features (e.g., important words or URL's), user features (e.g., verified account, number of followers/tweets, or the age of the account), usage features. Usage features are measured by the number of similar tweets, and tweets with the same hashtags within a time and location range to the current tweet.

Cui, Fu, Dong, and Zhang (2014) propose a method to extract traffic information from the Chinese social media platform equivalent of Twitter, called Sina Weibo. The paper does not contain a specified data collection and pre-processing approach. It detects traffic event-related posts into three categories (traffic flow, traffic accident, traffic control) by using a Bayesian classifier based on word n-gram features, however, no concrete results are provided. Moreover, temporal and locational features are extracted based on a custom natural language approach, which is not further elaborated. These two features are used to position the geosocial post on a geographic point or line. Besides a linking procedure, another novel idea has been implemented towards traffic event description, namely a QA system. When a geosocial post is labeled as traffic event-related but misses an incident category, temporal or locational aspect, the system sends a question to the user who posted the message to inquire additional missing information.

An automatic road hazard detection system based on tweets is proposed by Kumar, Jiang, and Fang (2014). Tweets are retrieved with the Twitter Streaming API based on a dictionary of terms related to hazardous events in the categories: animals, emergency, weather, special events, and traffic. Tweets without a geopoint (single latitude/longitude point) were discarded, stop words were removed and stemming was applied. The result set was manually labeled as hazardous or not hazardous. In this study, it is claimed that there is a relationship between negative sentiment and the mention of road hazards in a tweet. Therefore sentiment classification is applied to the labeled tweets based on word n-grams, with the help of three machine learning methods: *k*NN, NB, and Dynamic Language Model (DLM). NB proved to have the best precision of 77.5%, with a recall of 51.5% and accuracy of 81.2%, based on a dataset of 30,876 tweets.

An approach of detecting small-scale incidents (not limited to the traffic domain) based on spatial-temporal-type clustering is proposed by Schulz, Schmidt, and Strufe (2015). In their research, they try to solve the problem of clustering incident-related tweets based on incident-type, location and time. Data is collected through the Twitter Search API based on a geo-radius and is then preprocessed so it can be used for feature generation. Their preprocessing steps consist of: replacing abbreviations with words from slang dictionary, identification and replacement of locational and temporal expressions with a token, interlinking entities with types and categories in linked open data, and lastly tokenization. Next, non-alphanumeric characters are removed from the tokens and lemmatization is applied on the tokens. They select the following features to use for their classifiers: word-n-grams, char-n-grams, TF-IDF scores, syntactic features, number of locational and temporal mentions, and linked open data. The features get combined and evaluated by using the binary classifiers multinomial Naïve Bayes (NB) and Support Vector Machine (SVM), based on a sub-dataset of 2000 tweets collected over a period of 2.5 months. SVM with word-3-grams and binary weighting provided the best results with an accuracy of 90.1% and micro-avg. F1 of 90.05%.

The classified incident-related tweets are linked to a location, by using a custom location mapping technique. First, word-3-grams are created of location-tagged words (Stanford NER). Second, each n-gram gets mapped by using geocoding APIs (e.g., MapQuest Nominatim API). Third, based on the resulted sets of coordinate pairs for each n-gram a polygon is created. Last, the polygons get stacked and the highest area is used as an estimation of the incident location. In addition to location-based linking, tweets get linked to a time period by extracting temporal expressions (based on the HeidelTime framework, which uses regular expressions) and combining them with the creation date to calculate the most probable incident occurrence date. Based on the incident type, location and time period a rule-based clustering method is applied. Incident reports get clustered with each other when there is a corresponding incident type and the spatial and temporal extent falls within the extent of the defined in the rule. Their evaluation of the approach showed that 50% of real-world incidents published in an emergency management system could be detected. Furthermore, 32.14% of the incidents could be detected within a 500-meter radius and 10-minute interval around the actual event.

D'Andrea et al. (2015) propose a real-time traffic event detection system. Data is collected through the Twitter Search API based on a geo-radius and keyword list. Their preprocessing steps consist of: discarding hashtags, links, mentions, special characters, non-Italian tweets. Additionally, tweets get tokenized and stop-word filtering and stemming are applied. Next, to form a feature set, the weight of all stems is computed by using the IDF index. Then a method based on the computation of the Information Gain (IG) value between the feature set (stem set) and output set (traffic class labels) is applied, in which the set of relevant stems have a positive IG value. Based on the feature set of relevant stems, a multi-class classification is applied in which three traffic classes get distinct: non-traffic related, traffic congestion/crash and traffic due to an external (scheduled) event (e.g., sports match or concert). Several classification algorithms have been taken into account: SVM, Multinomial NB (MNB), C4.5 decision tree, k-nearest neighbor (*k*NN), and PART. When applying the classifiers on a 2-class (non-traffic, traffic based) dataset, SVM turned out to be best with an accuracy of 95.75%, precision of 95.3%, recall of 96.5%, and F1-score of 95.8%. The evaluation was performed on a dataset of 1330 tweets, collected over a time span of four evening hours of two weekend days. SVM also proved to be the best classifier with an 88.89% accuracy when applying the classifiers on a 3-class (non-traffic, traffic congestion/crash, traffic due to an external event) dataset of 999 tweets.

Nguyen, Liu, Rivera, and Chen (2016) developed a system that detects traffic incidents in real time by monitoring Twitter. Data collection is achieved by using a keyword-based query on the Twitter REST API. The result set of tweets is afterward filtered on a combination of geo-location, time zone, location and country from the user's profile in order to only obtain Australian tweets. Next, the following pre-processing techniques are applied: stop word filtering, special character filtering, and tokenization. In order to train the traffic incident classifier the following list of features are extracted: bag of words (each word gets a weight based on the accumulated TF-IDF score overall positive tweets); lemmatization and part-of-speech (POS) by applying the Stanford Twitter tagger; date, time and numbers by applying a custom pattern recognizer; bag of tags (custom NER, trained based on CRFs). Based on these features experiments with the following classifiers were executed: *k*NN, BN, SVN, C4.5 decision tree. The BN method based on a combination of all features delivered the best performance: precision of 94.2%, recall of 96.6%, and an F1-score of 95.4%, based on a dataset of 5000 tweets.

Even though the tweet sets have been annotated with a variety of location types (state, suburb, street, POI, place), entity types (people, vehicle, stationary object), incident types (queue, accident, breakdown, hazard, special event, police, roadwork) and properties (lane, direction, status), in order to train a custom NER, this information was not used to create a traffic event categorizer. However, tokens identified as one of the location types have been used to couple tweets to a location (no further details were given). Besides the 2.87% of tweets located based the device location, this custom geo-locator approach mapped an additional 19% of tweets. The final application consists of geo-located traffic event-related tweets mapped on a map in a real-time fashion.

In recent work by Gu et al. (2016) a methodology is proposed to crawl, process and filter tweets to extract incident information on highways and arterials. An adaptive data acquisition is used to collect tweets based on an iteratively composed keyword list used to query the Twitter REST API. First, an initial keyword list of traffic-related words that are to be included and words that are to be excluded from the query. In each iteration this keyword list gets expanded with a pair of synonyms per keyword derived from the WordNet database. Second, the resulted tweet set gets labeled traffic related or not. Third, for all tweets, the combinations of tokens and their labels (traffic event-related or not) get counted and aggregated. A set of tokens and their combinations with the highest positive and a set with the highest negative correlation, get added to the initial keyword set. This process is iterated until it is no longer cost-effective (adding new keywords does not yield enough new traffic-related tweets). This resulted in a final keyword set with 131 positive and 383 negative keywords. Additionally, tweets from 46 influential users got queried. The final keyword set of positively correlated keywords and combinations of keywords is used to form the feature space for a Semi-Naïve-Bayes classifier for detecting traffic incident related tweets. Tests with this classifier resulted in an accuracy of 98.94%, precision of 99.02%, and recall of 79.84%, based on a dataset of 5000 tweets.

In order to categorize traffic incident related tweets five categories are defined: accidents, road work, hazards & weather, events, and obstacle vehicles. Supervised Latent Dirichlet Allocation (sLDA) is used to assign a category label to the traffic incident related tweets. Its output is a vector containing a probability of a tweet falling in one of the five categories. 51% of the tweets proved to be categorized into the correct category by applying this method.

In this research, a geo-parser linking approach, based on a large set of regular expressions, is used as a way to describe traffic incidents. It contains rules for identifying roads and segments (based on markers and relational words). The resulted output is used as a query for a Gazetteer in order to identify the geo-location. The linking of the tweets has been validated by comparing it with the traffic incident data from the Road Condition Report System from the Department of Transportation Pennsylvania (RCRS). When comparing the traffic incidents based on tweets, allowing a 30-minute reporting time and a 1-mile distance discrepancy with the traffic incident data from RCRS, 71% of the incidents matched.

2.2.2 Geosocial Data based Related Work Evaluation

The research discussed in this section shows how geosocial data, which is almost always Twitter-based, can be utilized for traffic event detection, categorization, and description. However, traffic event categorization seems to be a less researched topic. Research involving traffic event detection shows that this can be divided into a data collection, pre-processing, and feature engineering stage. These stages are steppingstones to arrive at the classification stage, where machine learning and rule-based approaches are applied in order to create a traffic event detector. Table 2-1, provides an overview of the research that includes a geosocial data based traffic event detection approach. Traffic event categorization was neglected in most research, due to unknown reasons. Only one paper discusses a categorization approach based on sLDA, as is made visible in Table 2-2. Traffic event description approaches can be divided into linking, aggregation and visualization approaches. Just like with the traffic event detector approaches, linking and aggregation are often used as steppingstones towards a map visualization. Most of the research focuses on geocoding approaches based on a dictionary/gazetteer method, or rule-based methods. Temporal linking is mostly done based on the creation date of a geosocial post. Aggregation methods are not applied most of the times, resulting into a visualization approach that directly maps tweets based on the combination of geo-coordinates and creation time. Aggregation approaches that do have been used, cluster geosocial posts based on rules regarding incident types, geo-regions, and temporal ranges. An overview of all research that includes traffic event description approaches can be found in Table 2-3.
The overall weaknesses of only focusing on one source of geosocial data to detect, categorize, and describe traffic events are:

1. Reliability of the category assigned to a detected traffic event: Did the user use distinctive enough words to derive the correct event category?
2. Reliability of the spatial aspects of the detected traffic event: Was the user really on the location of the event at the time of posting the tweet, or did he post about an event he read or heard about? And did the user use accurate enough locational words to be able to derive the correct event location?
3. Reliability of the temporal aspects of the detected traffic event: Was the tweet composed directly after the traffic event, or does it refer to a historical or future event?

These disadvantages would mostly be non-existent if the number of social posts that refer to a single event would always be of a high quantity. That way, tweets could be aggregated together to improve the reliability of the detected event. However, the reviewed research

shows that the ratio of the number of traffic event-related tweets per location and time range proves to be very low.
An overview of the weaknesses in/what is missing from, current research based on geosocial data only, and how our work extents upon current work based on the opportunities these missing parts offer, is provided below.

1. **Weakness:** Research focusses on sub-parts of the traffic event domain, or mixes it with different incident domains. Including different incident domains could improve the overall results, while the sub-results related to traffic events are not as good.
   Opportunity: *Focus on a large range of possible non-recurrent traffic events.*
2. **Weakness:** Data collection approaches often seem biased due to a limited keyword selection approach. For example, by choosing traffic related keywords that are not ambiguous better results could be achieved, yet this limits the diversity and amount of collected traffic event-related geosocial posts.
   Opportunity: *Create a keyword based data collection approach that captures as many traffic event-related tweets as possible.*
3. **Weakness:** Data collection is performed over a too limited time range. This could cause bias towards traffic events that are time period bounded, e.g., rush hour traffic jams.
   Opportunity: *Perform data collection over a time range that is likely to include all types of traffic events.*
4. **Weakness:** Datasets are small in size, and likely to miss traffic event categories.
   Opportunity: *Collect larger datasets containing traffic events of every category.*
5. **Weakness:** Datasets include a mixture of traffic event-related geosocial posts from "real road-users", news agencies, bots, and emergency agencies. This affects the results, as traffic event-related geosocial posts from news agencies, bots and emergency agencies contain a different syntax than geosocial posts from "real road-users". This could have a biased positive result on the detection, categorization, and description methods.
   Opportunity: *Focus only on traffic event-related tweets from "real road-users".*
6. **Weakness:** Categorization approaches are almost non-existent or limited in scope.
   Opportunity: *Categorization of a large range of non-recurrent traffic event categories.*
7. **Weakness:** Geocoding approaches only take geopoint based linking into account. Geosocial posts, often contain multiple locational terms. By linking a geosocial posts to a single geopoint one creates conflicts based on location resolution.
   Opportunity: *Create a location linking approach that takes into account geopoint, -line, and –shape based linking. As well as takes into account the interrelationships between location terms in geosocial posts.*
8. **Weakness:** Visualization of traffic events (in an application) is static and only focuses on displaying the events on a map and showing their related descriptions.
   Opportunity: *Create an interactive map-based application that allows for the visualization of traffic events, but also contains interactive elements for data analysis.*

Geosocial Data - Traffic Event Detection					
Research	**Data Collection**	**Pre-processing**	**Feature Engineering**	**Classifier**	**Evaluation**
Wanichayapong et al. (2011)	Twitter Search API: keyword list	-Thai language filtering -Dictionary-based Tokenization	N/A	**Rule-based Dictionary approach**	Dataset size: 1249 tweets Accuracy: 91.75% Precision: 91.39% Recall: 87.53%
Ribeiro Jr et al. (2012)	Twitter: influential accounts	-Portuguese language filtering	N/A	N/A	N/A
Li et al. (2012)	Twitter REST API: adaptive keyword approach	N/A	-Twitter-specific (links, hashtags, mentions) - Traffic event-specific (time, location, numbers)	**Unspecified classifier**	N/A
Cui et al. (2014)	Sina Weibo	N/A	-Word n-grams	**Bayesian**	N/A
Kumar et al. (2014)	Twitter Streaming API: keyword list	-Discarding non-geo tweets -Stop word removal -Stemming	-Word n-grams -Sentiment	-*k*NN **-NB** -DLM	Dataset size: 30,876 tweets Accuracy: 81.2% Precision: 77.5% Recall: 51.5%
Schulz et al. (2015)	Twitter Search API: geo-radius	-Abbreviation replacement -Locational and temporal generalization -Linked open data -Tokenization	-Word n-grams -Char n-grams -TF-IDF scores -Syntactic features -Number of locational and temporal mentions -Linked open data	-Multinomial-NB **-SVM**	Dataset size: 2000 tweets Accuracy: 90.1% Micro-avg. F1: 90.05%
D'Andrea et al. (2015)	Twitter Search API: geo-radius and keyword list	-Italian language filtering -Discarding hashtags, link, mentions, special characters -Tokenization -Stop word removal -Stemming	-Set of relevant stems based on Information Gain between stem set and traffic class labels set	**-SVM** -NB -C4.5 decision tree -*k*NN -PART	Dataset size: 1330 tweets Accuracy: 95.75% Precision: 95.3% Recall: 96.5% F1-score: 95.8%
Nguyen et al. (2016)	Twitter REST API: keyword list	-Discarding special characters -Stop word removal -Tokenization	-Bag of words -Lemmatization -POS -Temporal, numerical features -Bag of tags (Custom NER)	-*k*NN **-Bayesian Network** -SVN -C4.5 decision tree	Dataset size: 5000 tweets Precision: 94.2% Recall: 96.6% F1-score: 95.4%
Gu et al. (2016)	Twitter REST API: adaptive keyword approach in combination with influential accounts	Tokenization	Word n-grams of positively correlated (towards traffic incidents) keyword tokens	**Semi-NB**	Dataset size: 5000 tweets Accuracy: 98.94% Precision: 99.02% Recall: 79.84%

Table 2-1: Traffic event detection based on geosocial data sources. The best performing classifiers have been made bold and relate to the evaluation metrics.

Geosocial Data - Traffic Event Categorization	
Gu et al, (2016)	-Categorization into 5 categories: accidents, road work, hazards & weather, events, and obstacle vehicles -sLDA is used to assign a category label, its output being a probability vector

Table 2-2: Traffic event categorization based on geosocial data sources.

Geosocial Data - Traffic Event Description				
Research	**Linking**	**Aggregation**	**Visualization/App**	**Evaluation**
Wanichayapong et al. (2011)	- Combination of Dictionary (Gazetteer) and Rule-based approach	N/A	N/A	Linking to road segment: -Dataset size: 3311 tweets -Accuracy: 76.85% -Precision: 62.77% -Recall: 95.36% Linking to road point: -Dataset size: 2942 tweets -Accuracy: 93.23% -Precision: 81.72% -Recall: 92.20%
Ribeiro Jr et al. (2012)	- Combination of Dictionary (Gazetteer) and Rule-based approach	N/A	Direct mapping of tweets based on geo-coordinates and creation time	N/A
Li et al. (2012)	Unspecified linking method	-Clustering on geo-regions and temporal range -Importance ranking based on content, user, and usage features	Direct mapping of tweets based on geo-coordinates and creation time	N/A
Cui et al. (2014)	- Unspecified temporal and locational NLP extraction approach - Unspecified geo-positioning approach - QA approach to complement missing labels	N/A	Direct mapping of tweets based on geo-coordinates, creation time or time specified through QA	N/A
Schulz et al. (2015)	- Geocoding approach based on word n-grams retrieved with an NER, geocoding APIs and polygon stacking - Temporal linking derived from a combination of regular expressions (HeidelTime framework) and creation date of the tweet	Rule-based clustering of event-related tweets based on incident type, location and time period.	N/A	Locational (500m)/Temporal (10min) linking: -Dataset size: 1271 tweets -Accuracy: 32.14%
Nguyen et al. (2016)	- A geocoding approach based on word types retrieved with a custom NER.	N/A	Direct mapping of tweets based on geo-coordinates and creation time	Geocoding: -Dataset size: 1056 tweets -Accuracy: 21%
Gu et al. (2016)	- Combination of Dictionary Gazetteer) and Rule-based approach	N/A	Direct mapping of tweets based on geo-coordinates and creation time	Geocoding: -Dataset size: 3776 tweets -Portion of geocodable tweets by influential users: 64.0% -Portion of geocodable tweets by individual users: 4.9%

Table 2-3: Traffic event description based on geosocial data sources.

2.3 Combination of Traffic and Geosocial Data

On the one hand, there exists a fair amount of research on traffic event detection based on traffic data in the Transport & Planning domain. On the other hand, there is a considerable quantity of research on traffic event detection, categorization, and description based on geosocial data in the Computer Science domain. However, research focusing on the combination of these two domains regarding the topics on traffic event detection, categorization, and detection, seems to be still in its infancy. In this section, the research conducted in this domain is discussed.

2.3.1 Combination of Traffic and Geosocial Data based Related Work

Daly et al. (2013) developed the Dub-STAR system that uses a mechanism that fuses traditional city traffic data sources with tweets in order to describe the underlying causes of traffic conditions. The system is able to infer links between traffic events and traffic congestion based on a traffic diagnosis method trained on historical traffic data within Dublin. In other words, it is able to explain anomalies such as congestion in real-time based on historic conditions. This is achieved by using Dublin Bus GPS speed data to define road segments as congested based on pre-defined rules. Other traffic data including Eventful (a web-based event sharing service) matched with DBPedia, Dublin Road Works, and LinkedGeoData is used to describe possible causes of the congestion. This is done on the basis of the semantic similarity, time window, and road network of the event. In order to describe additional aspects of these derived traffic events, a dataset of tweets is collected from three influential users who tweet about traffic in the Dublin area. In this research, it is presumed that these tweets are traffic event-related, so no traffic event detection classification is applied. In order to link the tweets to a geo-location, punctuation is removed and traffic abbreviations are expanded (e.g., rd to road). Each word is used to perform a dictionary lookup (Lucene index based on OpenStreetMaps), if no result is found a spelling checker is applied to check for any misspelling. All found words are extracted from the tweet and used to create word n-grams. These n-grams are again used to perform a dictionary lookup and removed if no results are found. The resulting n-grams are used to search for a location. An evaluation of their geocoding approach, based on a dataset of 719 tweets, showed that 50% of the tweets were matched accurately with an error range of 500 meters (100% when applying an error range of 2 km). Additionally, tweets get semantically annotated based on a simple dictionary approach based on the categories: delay, incident, event, closure, roadworks, obstruction, and weather. The tweets are matched to the traffic events derived from the traffic data, based on a similarity estimation. This estimation is computed by using the semantic description of the event and spatial and temporal connectivity of events. This all comes together in an application called Dub-STAR in which real-time traffic events are visualized on a map. The system supports free text queries, coordinate-based queries, and filtering on specific types of events. The system itself is evaluated based on the same dataset of 719 tweets, having a recall of 78% and a precision of 20%.

Dokter (2015), conducted his thesis research, as part of the WIS group at the TU Delft, on the characterization of traffic events using social media. In his thesis, geosocial data from Twitter and traffic data from the National Data Warehouse for Traffic Information (NDW) is used. Twitter data is collected by using a keyword list of traffic terms and a bounding box covering the Netherlands, in combination with the Twitter streaming API. A linkage method is developed to link tweets to traffic events from the traffic data set of the NDW. Within this method, geocoding is performed based on the similarities of textual location descriptors within NDW data and tweets. Temporal linking is performed based on the creation date of tweets and the creation date of events in the NDW data. Linked events are classified by matching the linked tweet tokens with a traffic dictionary. This way the most occurring cause type is used as the type of the traffic event. An evaluation of the linking method on a small subset of tweets gave a precision of 96% and a recall of 80%. An evaluation of the classifier showed that the system is able to identify the traffic cause type for 63% of the events, from which 33% was classified correctly. The proposed system proved to have difficulties with the classification of cause types due to: incorrect linking of tweets to NDW data, tweets of non-real-time nature, and the limited amount of traffic event-related tweets.

Giridhar et al. (2017) propose a traffic anomaly explanation service using Twitter data, named ClariSense+. This service is an extension on their previous system ClariSense (Giridhar, Amin, Abdelzaher, Kaplan, George, & Ganti, 2014) and enhances their base algorithm by considering the credibility of the tweets and the spatial locality of detected traffic anomalies. On the one hand, the system relies on traffic sensor data. It detects anomalies in sensor reports and clusters these sensors based on distance and time overlap. An anomaly gets defined as an unusual flow interruption on major freeways. A sensor anomaly detection algorithm called the Performance Management Systems (PEMS) analysis tool is used to report the start, end, duration and sensor IDs for each detected anomaly. Additionally, each anomaly is classified in the categories: Accidents, Hazards, Breakdowns, Weather, and Other events. It must be noted that this anomaly detection algorithm is not further explained and no information is given on how traffic events are categorized based on unusual traffic flow interruptions. Clustering of nearby sensor anomaly reports is used to remove redundant observations. Sensors are part of the same cluster when they are less than 2 miles apart from each other. On the other hand, the system relies on the data from the Twitter Search API, collected by using a bounding box for the chosen city and the keyword "traffic". The proposed method does not rely on semantic analysis of tweets but instead focusses on the question if there can automatically be found a set of keywords that has a one-to-one correspondence with a unique event. First, all words longer than four characters long get removed from the tweet set. Second, a POS tagger is applied to identify nouns, which serve as keywords. Last, bi-grams of keywords are formed and ranked by information gain. By comparing this information gain with a certain threshold it can be determined that these keyword pairs occur disproportionally more frequently compared to the historical normal. When this is the case the tweets get labeled as traffic event-related. After having created a traffic and geosocial data set, the geosocial data can get matched to the traffic data. Tweets get matched to a sensor anomaly based on location keywords that occur in the tweet and geo-keywords (e.g., highway number, exit names, landmarks) associated with each physical sensor. Tweets that match the location of (containing one or more corresponding keywords), and occur within 24 hours from the traffic event are sorted by information gain. The top tweets are then used as an explanation of the traffic events. Their service is evaluated based on a dataset of tweets over a 3-week period for the cities Los Angeles (avg. of 850 tweets per day), San Francisco (avg. of 300 tweets per day) and San Diego (avg. of 800 tweets per day). The evaluation of their service showed an average recall over the three cities (number of correctly explained events among all returned tweets) of 85.7% for Hazard, 83.5% for Accident related traffic events. The precision is measured by determining how good the algorithm is at picking the right traffic event category for the traffic data anomalies at hand. Their approach resulted in a precision over all three cities of 88.63%.

2.3.2 Combination of Traffic and Geosocial Data based Related Work Evaluation

Research based on the combination of traffic and geosocial data proved to be limited and distinguishes itself mostly in its traffic event description approach as summarized in Table 2-6. Herein, possible traffic events derived from traffic data anomalies can be used to link geosocial data with, towards improving the description of traffic events. By enforcing geosocial data to be linked to traffic data in order to describe a traffic event, the overall weakness (reliability of the category, spatial, temporal aspects of the detected traffic event) of using only a geosocial data source is partly solved. Besides containing the same weaknesses and opportunities as described in Section 2.2.2, these works bring another weakness and opportunity to the light.

1. **Weakness:** In these works, a combination of traffic data anomalies and geosocial data is always needed to infer a possible traffic event. However traffic events can happen without any traffic data anomalies appearing, e.g., road debris does not necessarily lead to traffic data anomalies, but is considered a traffic event. In this case, any geosocial data referring to this road debris would be discarded.
   Opportunity: *Use multiple geosocial data sources and if possible link these to roads containing traffic data. On the one hand, by aggregating geosocial data sources, the weakness related to the reliability of one geosocial data source is reduced. On the other hand, geosocial data is always linked to roads containing traffic data, so when anomalies appear no geosocial data is lost and it can be used to describe the anomalies.*

Traffic and geosocial data based research containing traffic event detection methods are further summarized in Table 2-4. The only research containing a traffic event categorization approach can be found in Table 2-5.

Traffic Data & Geosocial Data - Traffic Event Detection				
Research	**Data Collection**	**Pre-processing**	**Feature Engineering**	**Classifier**
Daly et al. (2013)	**Geosocial data:** Twitter: influential accounts **Traffic Data:** -Dublin Bus GPS speed data -Eventful (matched with DBPedia) -Dublin Road Works -LinkedGeoData	**Geosocial data:** -Abbreviation replacement -Discarding punctuation	N/A	**Traffic Data:** Rule based congestion classifier
Dokter (2015)	**Geosocial data:** Twitter streaming API: keywordset and boundingbox **Traffic Data:** -NDW dataset	**Geosocial data:** -Bot filter -Ban word filter -Discarding special characters and punctuation -Tokenization -Remove tokens with < 4 characters	N/A	**Geosocial data:** Linked tweets to NDW data get a matching percentage based on term similarity
Giridhar et al. (2017)	**Geosocial data:** Twitter Search API: geo-radius and keyword **Traffic Data:** Roadway-based sensors flow data	**Geosocial data:** -Discarding words containing less than 5 characters -Discarding non nouns based on POS tagger	**Geosocial data:** -Set of word bi-grams based on Information Gain	**Geosocial data:** Classifier based on comparison of Information Gain with threshold **Traffic Data:** Performance Management System classifies flow anomalies into: accidents, hazards, breakdowns, weather, and 'other' events

Table 2-4: Traffic event detection based on traffic data & geosocial data

Traffic Data & Geosocial Data - Traffic Event Categorization	
Daly et al. (2013)	-Categorization into 7 categories: delays, incidents, social events, closure, roadworks, obstruction, weather. -Dictionary approach
Dokter (2015)	-Categorization into 11 categories: rush-hour, accident, event, non-technical, technical, construction, weather, breakdown, other, unknown. -Dictionary approach

Table 2-5: Traffic event categorization based on traffic data & geosocial data

Traffic Data & Geosocial Data - Traffic Event Description				
Research	**Linking**	**Aggregation**	**Visualization/App**	**Evaluation**
Daly et al. (2013)	- Dictionary approach applied first on all words in a tweet (spelling correction is applied when necessary) and followed by n-grams if a word matched a term from the dictionary - Gazetteer	Confidence ranking, based on spatial-temporal relationship between congestion and potential causes	Direct mapping of tweets based on geo-coordinates and creation time Mapping of traffic data based on semantic similarity to historical events, time window, and road network.	Geocoding: -Dataset size: 719 tweets -Accuracy: 50% (<500m) -Accuracy: 100%(<2000m) System: -Dataset size: 719 tweets -Precision: 20% -Recall: 78%
Dokter (2015)	- Geocoding approach based on the similarities of textual location descriptors within NDW data and tweets. -Temporal linking approach based on the creation date of tweets and the creation date of events in the NDW data.	Tweets get aggregated to NDW events based on locational and temporal similarities.	Direct mapping of clusters based on geo-coordinates and creation time of NDW events. Tweets and NDW events also get mapped by themselves if geo-coordinates are available. Additionally, an experiments part is provided where users can interact with the system to test different linkage strategies.	Event linking: -Dataset size: 100 events -Precision: 96% -Recall: 80%. Classifier: -Recall: 63% -Precision: 33%
Giridhar et al. (2017)	-Dictionary approach, matching location keywords in tweets to location keywords associated with physical roadway –based sensors -Temporal linking based on creation time of tweets	N/A	Mapping of tweets based on geo-coordinates and creation time to a physical sensor cluster that indicates a possible traffic event	Event category: -Dataset size: approx. 40.950 tweets -Recall: 85.7% Hazard event, 83.5% Accident event -Precision: 88.63%

Table 2-6: Traffic event description based on traffic data & geosocial data

2.4 Additional Related Work

As the research on traffic event detection, categorization, and detection, by utilizing a combination of traffic and geosocial data seems to be still in its infancy, this section provides a short overview on closely relevant topics that do use this combination. These topics include traffic prediction, traffic and geosocial data correlation, and traffic congestion monitoring.

He, Shen, Divakaruni, Wynter, and Lawrence (2013) examine the possibilities of using Twitter data to improve long-term traffic prediction. The Twitter Streaming API is used to collect tweets based on a geo-bounding box, and stop word removal and stemming is applied. For this same location, a traffic dataset is generated by collecting measurements of loop detectors. By applying a correlation technique they establish that there is indeed a significant correlation between the intensity of traffic and social activity (tweet counts). Next, a general optimization framework to extract traffic indicators based on traffic intensity and tweet semantics is proposed. The evaluation of the model shows that the additional information in tweets indeed helps to improve the performance of traffic prediction, in terms of mean absolute percentage error and root mean square error.

Tostes, Silva, Duarte-Figueiredo, and Loureiro (2014) study the correlation between Foursquare and Instagram posts and congested traffic flows from Bing Maps. Their goal is to verify if these geosocial posts can be used as an indicator of traffic condition changes within Manhattan, New York City. Their method to evaluate the correlation consists of five steps. First, the geosocial posts are aggregated into 3-hour periods due to the long time intervals between posts. Additionally, the traffic flow for streets gets categorized into three groups representing fast, moderate and slow traffic. Second, the mean and standard deviation of the number of geosocial posts per street segment is calculated. Third, based on the relation between the number of posts and the mean and standard deviation, the geosocial post gets assigned to one of the three traffic flow categories. Fourth, five groups are created to analyze the correlation between the geosocial posts categories and traffic flow categories (e.g., geosocial posts category is less than the traffic flow category, in other words when the number of geosocial posts is low, the traffic flow is more congested). Last, for both categories, a distribution of the frequency of geosocial posts during 24 hours, and the frequency of congested average traffic flow is created. Based on a temporal and spatial analysis the authors were able to show that the distribution of geosocial posts is equal to that of congested traffic flows (with a discrepancy error). However, due to the time difference between an occurred congestion and a geosocial post, the signal time of traffic flow congestions has a 36 minutes delay.

Silva, de Melo, Viana, Almeida, Salles, Loureiro (2013) research how the geosocial data from Waze can be used to derive a participatory sensor network (PSN), to gain a better understanding of traffic problems, city dynamics and urban behavioral patterns of users. Waze data is collected through tweets containing Waze alerts, which means that only a fraction of all Waze data is used. They found that spatial coverage of Waze alerts is greatly influenced and correlated by the number of circulating vehicles per region. Additionally, Waze alert sharing happens at specific intervals, where alerts cluster towards specific events. When looking at the user activity, a great variability of user participation was measured. The routines of the users proved to correlate with rush hour peaks and proved much lower during late night hours and dawn.

P. Chen, F. Chen, and Qian (2014) propose an approach towards traffic congestion monitoring, by combining a language model and Hinge-loss Markov Random Fields based on Twitter data on traffic events. In this approach, two datasets containing Twitter data and traffic speed data are used. Tweets are collected by using the Twitter REST API and a selection of traffic-related keywords. These tweets are categorized based on if they report on traffic accidents or not by using an SVM classifier. Traffic data is taken from the INRIX database, which provides traffic speed and reference speed information for road links at a 5-minute rate. Next, a custom traffic language model is applied to model tweet descriptions that describe free and congested traffic conditions. Furthermore, a probabilistic soft logic (PSL) model (based on 11 PSL rules) is used to detect traffic congestions and includes tweet geocoding. These two models are integrated into a newly proposed Language enhanced Hinge-Loss Markov Random Fields model. In an evaluation of the model, an average recall value of 70.4% and a precision of 48.7% was measured.

Another traffic congestion estimation model is proposed by Wang, He, Stenneth, Yu, and Li (2015). In this study, a coupled matrix and tensor factorization algorithm is proposed to combine data from Twitter, road features, and social events, in order to create a traffic congestion estimation model. Twitter data is collected from 11 influential accounts that focus on traffic, as well as from users that have Chicago registered as their hometown in their Twitter profiles. These tweets are divided into traffic event categories based on a dictionary based word matching approach. Additionally, social events are extracted from Twitter accounts focusing on social events. These tweets are all geocoded to road segments by matching them to terms from a gazetteer. The traffic data set consists of road features including the segment length, number of lanes, one-way road, road heading, and number of intersections. And eight types of places of interest (POI), e.g., schools and hospitals. Based on this real-time data a real-time congestion matrix and event tensor are constructed. Besides real-time data, historical data is used to conduct congestion probability summarization of road segments, and road segment congestion co-occurrence pattern mining. Evaluations of these methods on real data in Chicago showed that the proposed method is effective.

2.5 Evaluation of Related Work

In this chapter, we looked at the related work regarding traffic event detection, classification, and description methods. In this section we summarize the most important takeaways from this literature review.

2.5.1 Related Work Key Points

1. **Traffic Data:** traffic event detection proved to be the only focus of research based on traffic data. Traffic event detection is based on algorithms that depend on data from roadway-based sensors. Weaknesses of this approach include:
   a. Quality of measurements depend on the density of the sensor network
   b. Noise corruption of sensor network.
   c. Algorithms are road-type dependent, i.e. algorithms that can be applied on freeways are often not suitable for arterial situations which are much more complex.
2. **Geosocial data:** traffic event detection, categorization, and description were all part of research based on geosocial data. Weaknesses of this approach include:
   a. Reliability of the categorical, spatial, and temporal aspects of the derived traffic event, by only using one geosocial data report as basis to derive a traffic event.
   b. Focus lays on sub-parts of the traffic event domain, or is mixed with different incident domains unrelated to traffic.
   c. Biased data collection approaches, due to limited keyword selection and time ranges, dataset size, and mixed geosocial post authors (e.g., "real road-users", news agencies, bots).
   d. Categorization approaches are non-existent or limited.
   e. Geocoding is limited.
   f. Visualization of traffic events is static.
3. **Combination of Traffic Data and Geosocial data:** Research based on the combination of traffic and geosocial data proved to be limited and distinguished itself mostly in its traffic event description approach. Weaknesses of this approach include:
   a. A mandatory combination of traffic data anomalies and geosocial data can lead to a loss of important semantic data.
4. **Additional Related Work:** Other related research showed us that geosocial data can contribute to traffic data for other means than traffic event detection, categorization, and description. Nevertheless, these works contain many overlapping approaches, and therefore gave us a better understanding of our own research domain.

2.5.2 This Work versus Related Work

In this thesis, we aim to mitigate the weaknesses and fill in a selection of open research gaps found in the related work. First, in this work, we do not rely on traffic data alone to derive traffic events. Instead, we enrich traffic data with geosocial data and use the combination to derive traffic events. This way, even though the traffic data on its own would have been inconclusive to derive a traffic event, the inclusion of supporting geosocial data can help make this conclusive. Additionally, this work provides a traffic event categorization and description approach based on geosocial data, which is not possible when relying on traffic data alone. Second, in this thesis, we aim to mitigate the reliability issues related to the categorical, spatial and temporal aspects of geosocial data reports, by not relying on a single geosocial data report to derive a traffic event. Instead, we combine multiple geosocial data reports from different sources, by clustering them based on categorical, spatial and temporal similarities, to derive a traffic event. Third, a keyword-based data collection approach is created that mitigates the bias formed due to a limited keyword selection approach. Fourth, in this work, the focus lays on traffic event derivation through geosocial data from "real road-users", instead of using a mixture of different user-types including news agencies, bots, and emergency agencies. Thus filling in the gap of specialization towards understanding traffic event through "real road-user" geosocial data. Fifth, a rule-based traffic domain annotator is created to annotate tweets and to assign a wide range of traffic event categories. This fills in the open research gap towards categorization, as these are limited in scope in related work. Sixth, related work only uses geocoding approaches based on locational terms and their derived geopoint, within geosocial posts. In this work, we fill this gap by taking into account that tweets can contain multiple locational terms that can correlate, contradict, and confirm each other. Additionally, our approach takes into account that spatial indicators can relate to different geographic forms and scales, and as well can be ambiguous (locational term matches multiple locations). Seventh, related work that combines traffic data and geosocial data work from the perspective of traffic data and try to describe this data by linking traffic event-related geosocial data to it. In this work, a less limited approach is taken, by working from the perspective of clustering geosocial data to derive a traffic event and link traffic data to this event based on temporal and locational features. This way we reduce the loss of valuable data caused due to mandatory linking of traffic data and geosocial data. This as, traffic events can happen without any traffic data anomalies appearing, e.g., road debris does not necessarily lead to traffic data anomalies but is considered a traffic event. Lastly, related work leaves a gap when it comes to using the implementation of the pipeline that detects, categories and describes traffic events, in an interactive map-based application. We fill this gap by creating this application, allowing for the visualization of traffic events, and providing interactive elements for data analysis.

3 Experiment Design

This chapter focuses on the design of a pipeline for extracting knowledge on traffic events from geosocial and traffic data sources. The literature study as described in 2, showed us what current research on traffic event detection, categorization and description models based on a combination of traffic and geosocial data lacks and where the opportunities lay for our work. This enables us to design our own pipeline containing the detection, categorization, and description parts, as depicted in Figure 3-1. This chapter discusses the experiment design setup of each pipeline part, leading up to the answering of the following research sub-questions:

- ***RQ2:*** *How can non-recurrent traffic event-related geosocial posts automatically be detected?*
- ***RQ3:*** *How can detected non-recurrent traffic event-related geosocial posts be categorized by event type?*
- ***RQ4:*** *How can categorized geosocial posts be used to describe non-recurrent traffic events?*
- ***RQ5:*** *How to develop a software system that is able to perform the detection, categorization, and description of non-recurrent traffic events?*

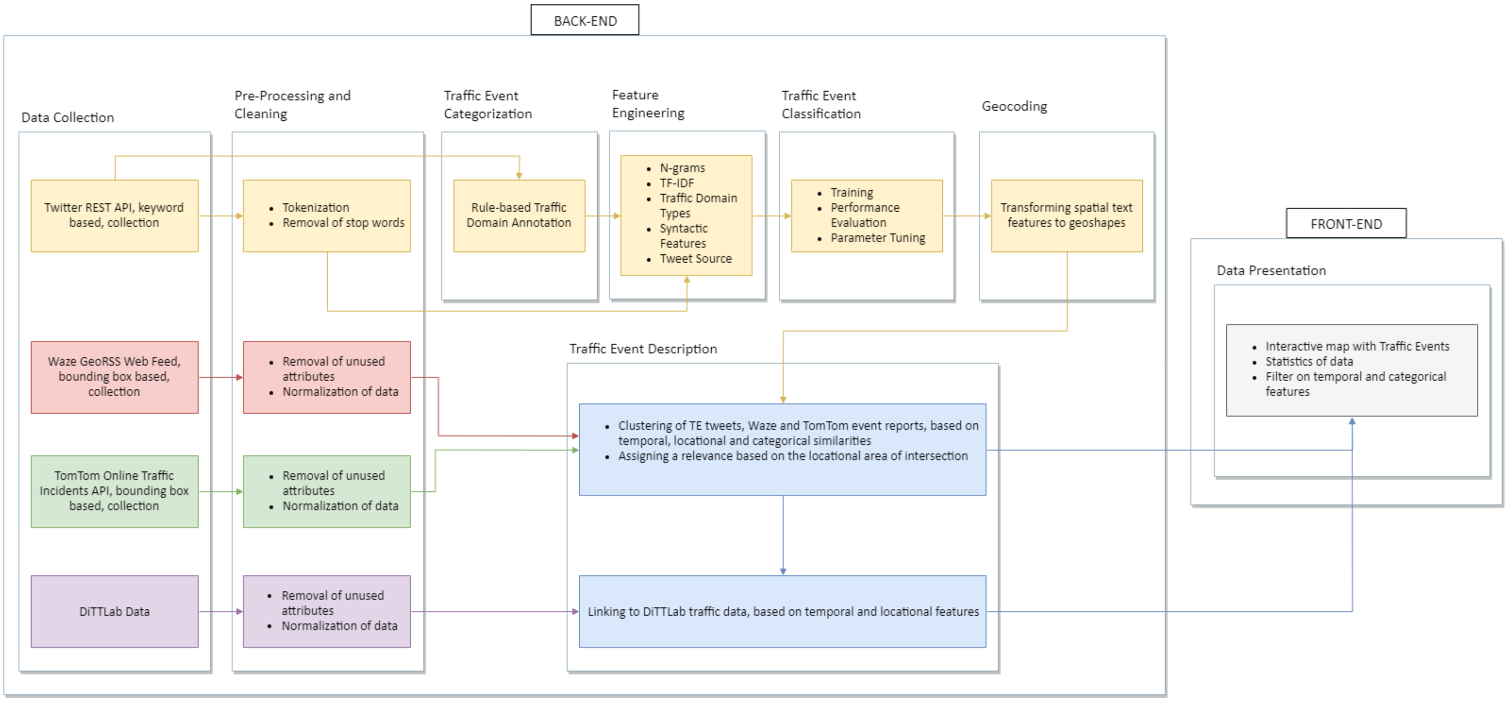

BACK-END
Data Collection
Pre-Processing and Cleaning
Traffic Event Categorization
Feature Engineering
Traffic Event Classification
Geocoding
Twitter REST API, keyword based, collection
Tokenization
Removal of stop words
Rule-based Traffic Domain Annotation
N-grams
TF-IDF
Traffic Domain Types
Syntactic Features
Tweet Source
Training
Performance Evaluation
Parameter Tuning
Transforming spatial text features to geoshapes
Waze GeoRSS Web Feed, bounding box based, collection
Removal of unused attributes
Normalization of data
TomTom Online Traffic Incidents API, bounding box based, collection
Removal of unused attributes
Normalization of data
DiTTLab Data
Removal of unused attributes
Normalization of data
Traffic Event Description
Clustering of TE tweets, Waze and TomTom event reports, based on temporal, locational and categorical similarities
Assigning a relevance based on the locational area of intersection
Linking to DiTTLab traffic data, based on temporal and locational features
FRONT-END
Data Presentation
Interactive map with Traffic Events
Statistics of data
Filter on temporal and categorical features

Figure 3-1: Overview of experiment design methods

3.1 Data Collection

The data collection part is the starting point of our pipeline. In this study data sources containing social and traffic data are used. For a data source to be used in this study, it needs to provide a significant enough amount of possible traffic event-related data in order to be able to conduct an analysis on it. Additionally, it must contain data within the targeted geographical area, as well as temporal period. The initially chosen data sources include Twitter, Instagram, Waze, TomTom, and DiTTLab. We describe the data collection approach for each data source, aligning the data towards the subject, temporal, and locational scope of the study.

3.1.1 Twitter and Instagram

The main goal that we want to achieve when collecting traffic event-related (TE) geosocial posts, is to create a keyword set that maximizes the percentage of TE geosocial posts over all acquired geosocial posts and maximizes the amount of acquired TE geosocial posts in the pool (all Dutch geosocial posts in the Netherlands within a specific time range). This means that at the same time we want to achieve as much recall and precision as possible. Recall and precision are defined as follows:

$$Recall = A \cap B \;/\; A$$
$$Precision = A \cap B \;/\; B$$

Where A is the set of all TE geosocial posts within a specified time period and B is the set of all geosocial posts within the same specified time period. The recall is defined as the number of true positives, denoted as the intersection between A and B divided by the number of relevant geosocial posts. Whereas precision is defined as the number of true positives divided by the number of retrieved tweets (combination of true and false positives). Even though it is known how to calculate recall, doing so is complicated in the case of data mining Twitter, as there is no ground truth available that describes a full set of TE geosocial posts. Obtaining a 100% precision is also impossible as the used keyword set will always contain ambiguous keywords, resulting in geosocial posts that are not traffic event-related (NTE).

Twitter is offering two suitable options for data mining to developers. The first one is their REST API, which enables developers to search for tweets based on keywords and location radius. It is also possible to define a set of keywords with operators including OR, AND, and EXCLUDE, as well as pre-filtering out languages, specific user accounts, retweets, links, replies, and mentions. However, the free and standard version comes with a number of limitations. It only enables developers to search up to ten days back in time and API calls are limited to 180 calls every 15 minutes. The second option is their streaming API, which allows for real-time tweet collection. A single HTTP connection is opened between the app and the API, resulting in new results whenever matches occur. Compared to the REST API, which enables the app to obtain data in batches through multiple requests, the streaming API has a lower latency and supports a very high throughput. This, however, comes with a number of limitations. Only 1% of all public tweets can be obtained from the stream. The streaming API does not support keyword operators or pre-filtering operations. The original idea was to use the SocialGlass[12] application to obtain the TE-related Twitter data. But as SocialGlass makes use of the streaming API this option is not feasible, as this would lead to a massive loss of possible relevant tweets. Take for example a query containing the Dutch keyword “file” (EN: traffic jam). As the streaming API has no pre-filter option on language, it returns 1% of all public tweets containing the keyword “file”. This causes us to miss out on a lot of possible relevant TE tweets, as the returned set is “contaminated” with irrelevant English tweets. Therefore, we decided to use the REST API as it is more important to obtain the complete set of TE tweets than to get a large stream of public tweets in real time.

The REST API can be used with a keyword set, a geocoordinates radius or a combination of the two. As we aim to obtain an as large as possible set of TE tweets, using only a keyword set is the most logical option. By only querying based on a geocoordinates radius, one collects many irrelevant tweets resulting in a possible loss of TE tweets, besides these tweets have to be filtered afterward to obtain a set of possible TE-related tweets. The initial set of keywords is based on the keywords used in the thesis by Dokter (2015). In this thesis a set of suitable keywords is defined to find TE-related tweets, consisting of: file (EN: traffic jam), ongeluk (EN: accident), pech (EN: breakdown), brug (EN: bridge), langzaam rijden (EN: drive slowly), traag rijden (EN: slow moving), km, spits (EN: rush hour), verkeer (EN: traffic), gekanteld (tilted), gekantelde (EN: overturned), aanrijding (EN: collision).
It is our assumption however that this initial keyword set is just a subset of often used traffic event-related keywords within tweets. In addition, this initial keyword set could lead to too many NTE tweets, due to the ambiguity of some keywords. Therefore, a method is created to improve the quality and quantity of TE tweets that can be acquired through a keyword set.

We extend the initial keyword set with the road numbers of the Dutch road network. Next, a Dutch language, retweet and replies filter is applied to the query, as we are only interested in Dutch tweets directly posted by road users. Initial results showed us that tweets containing URLs that link to external websites are never TE-related. Based on this discovery we decided to also filter out any tweets that contain a URL. Note that this does not include tweets that contain an embedded media link (containing a photo or video), as these are relevant. To achieve this, we use the “filter on links” option the API provides us. Furthermore, a rule-based filtering method is applied to filter out the majority of non-real road user accounts. We define a real road user as follows: a natural person that tweets on his/her own account and is a road user, therefore excluding all legal person entity accounts such as public organizations (government agencies, police, and infrastructure agencies), private organizations, and bots. Tweets from these accounts are filtered out, based on suspicious terms in their name or username, as well as a manually composed list of non-real road user accounts. The result set is manually labeled, as TE or NTE posted by a real road user account as well as TE or NTE related but posted by a non-real road user account. The entire set of tweets is then processed as follows:

1. Filter tweets on stop words, based on a Dutch Twitter stop word list.
2. Strip the tweets from any URL links.
3. Transform tokens of road names to a general road number tag.
4. Tokenization of tweets through the tokenizer from the Frog NLP library (Bosch, Busser, Canisius, & Daelemans, 2007).

[12] social-glass.tudelft.nl/

5. For each token and its bigram, compute how many times it appears in TE and NTE tweets posted by a real road user account.
6. For each token and its bigram, compute how many times it appears in TE and NTE tweets posted by a non-real road user account.
7. For each token and its bigram, compute how many times it appears in combination with other tokens in TE tweets (positive co-occurrence) and NTE tweets (negative co-occurrence).

By following this process, we are able to identify keywords with their positive and negative correlation towards TE tweets. Tokens are manually added to the keyword set when they are not too ambiguous and appear in more than one TE-related tweets and the following is rule holds:

# of tokens in TE tweets/ (# of tokens in NTE tweets by non-real road user accounts) > 0.05

In this rule, we divide the number of appearances of a token in TE tweets through the number of times it appears in NTE tweets by non-real road user accounts, as these will get filtered out in the next iteration and should not result in a lower ratio. The five percent threshold has been chosen, based on initial tests with multiple thresholds. This threshold, proved to bring the best balance between accepting keywords indicating TE tweets and rejecting keywords indicating NTE tweets. The idea was to automatically extend these tokens with synonyms retrieved from ConceptNet (Speer & Havasi, 2012). However initial results showed that this adds too many ambiguous new terms, and therefore cannot be used as an automated process. Therefore, for this experiment we choose not to add synonyms. Additionally, we experimented with adding tokens to a negative keyword list when they are not too ambiguous, and if they do not appear in TE tweets and appear more than 20 times in NTE tweets. Tweets that contain a token from the negative keyword list would not have been collected. This however, resulted in the loss of too many TE tweets, due to ambiguity problems and was therefore left out. This entire process is iterative and ends when no more new positive keywords are found. An overview of the process can be found in Figure 3-2. Table 3-1 shows the properties of each collected tweet. The final keyword set is used to collect geosocial data from Twitter and Instagram.

For collecting data from Instagram we follow the same approach as with Twitter. The main difference however, is that Instagram offers an API that only allows for keyword queries on tag objects instead of text objects. This causes a significant decrease in results, as tags have to be manually added to an Instagram post by a user in contrast to Twitter where tags are part of the tweet text itself. Therefore, based on initial experiments, we made a well-substantiated decision to no longer include Instagram in our setup.

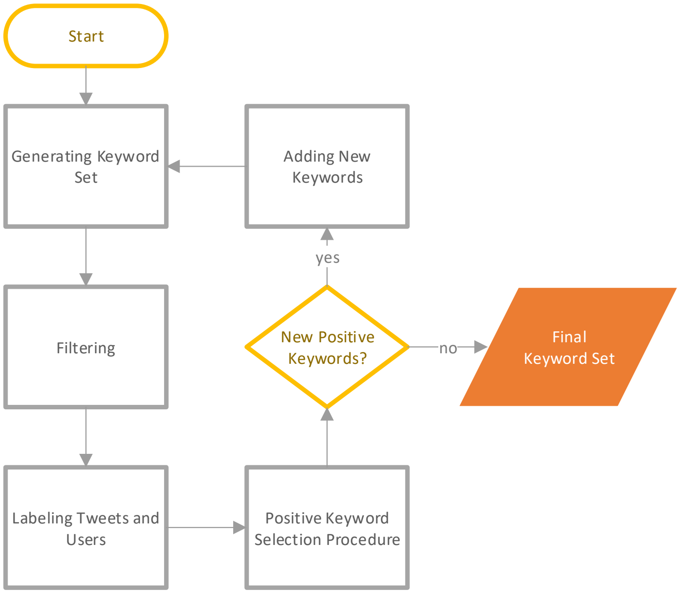


Figure 3-2: Adaptive keyword selection flowchart

Tweet Properties	
Tweet	**User**
Id	Name
Text	Screen name
Creation date	Description
Geocoordinates	# Followers
# Retweets	# Friends
Language	Language
In reply to status id	Profile image URL
In reply to user id	User home location (as defined by the user in its profile)
Source	Profile creation date
# Favorited	
URLs	
Hashtags	
User mentions	
Symbols	
Media	

Table 3-1: Tweet properties

3.1.2 Waze

Waze is a community-based traffic and navigation application, which integrates data provided by users to inform other users on all sorts of traffic events. Contrary to the other data sources that we use, Waze does not have an API for data collection. Therefore, an alternative data collection approach is set up. In this approach, the web-based live map from Waze is monitored, as depicted in Figure 3-3. Through this map we extract data in the form of a GeoRSS web feed. By specifying a geo bounding box all Waze live map data within that region can be extracted, up to a limit of 200 alerts and 100 jams (two parent categories under which all data is grouped). Because of this limitation, the initial bounding box covering the Netherlands is automatically split into sub bounding boxes until we collect less than 200 alerts and less than 100 jams. As the live map is updated every two minutes, our method downloads the JSON files in two-minute intervals.

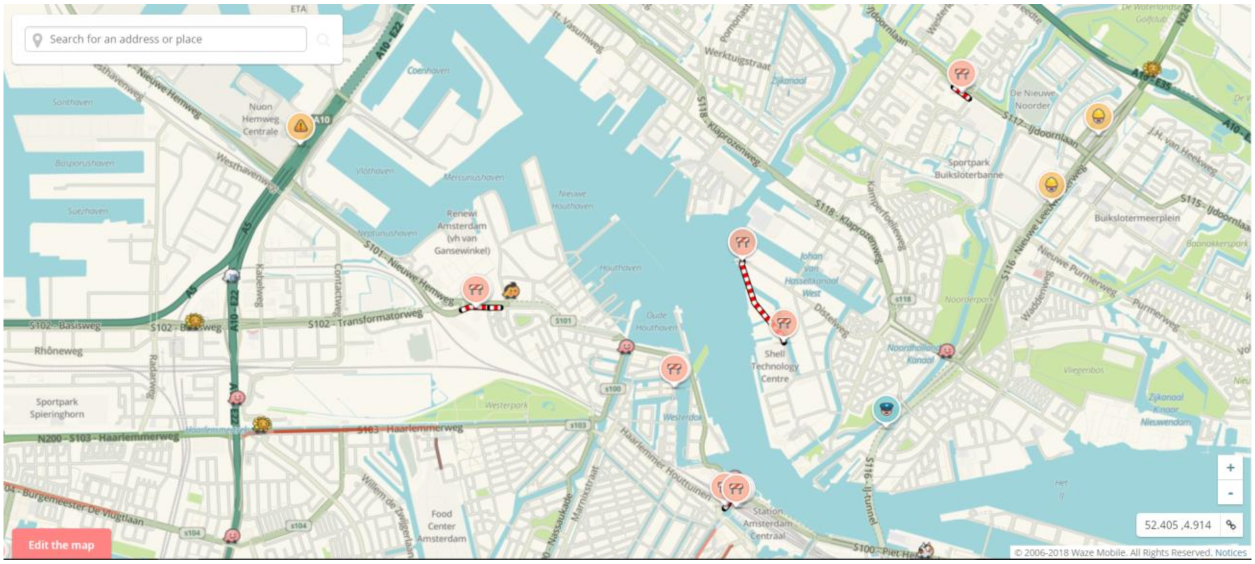


Figure 3-3: Waze Live Map, where the icons represent users and traffic events

Waze subdivides its data into three main categories, each with its own set of attributes as shown in Table 3-2. The Alerts category, contains a wide selection of all sorts of traffic events. The Jams category, extends upon closed road types, including construction types from the alerts category. The category name "Jams" proved to be misleading as it includes no information on traffic jams. The Users category, contains anonymous information on active users. When a Waze user reports a traffic event he chooses from a set of inheritance based traffic event types. This way it is possible for a user to choose a more abstract parent type (e.g., Hazard) or a more specified child type (e.g., Hazard On Shoulder Car Stopped). The following event types can be used in Waze to categorize traffic events:

- Accident: Minor, Major
- Hazard:
  - On Shoulder: Animals, Car Stopped, Missing Sign
  - On Road: Car Stopped, Construction, Ice, Lane Closed, Object, Oil
  - Weather: Fog, Hail, Heavy Rain, Heavy Snow, Flood, Freezing Rain
- Police: Visible, Hiding
- Jam: Stand Still Traffic, Moderate Traffic, Heavy Traffic
- Road Closed: Event Construction

Alerts		**Jams**		**Users**	
Attributes	**Example**	**Attributes**	**Example**	**Attributes**	**Example**
Country	NL	*Country*	NL	*Speed*	25.77
City	Delft	*City*	Delft	*GeoPoint*	[4.877815, 51.823144]
# of Thumbs Up	1	*Description*	Werkzaamheden		
Report Rating	3	*GeoLine*	[4.877815, 51.823144], [4.877817, 51.823145]		
Reliability[13]	7	*Length*	37		
Type	HAZARD	*Type*	NONE		
Speed	0	*Block Type*	ROAD_CLOSED_EVENT		
Subtype	HAZARD_ON_ROAD_OIL	*Speed*	0		
Street	Oostplantsoen	*Street*	A13		
Image URL	https://s3.amazonaws.com/waze.photos/36408761-0c20-4a2a-8730-2fcf847db845	*Severity*	2		
Reported by	BasdeBock	*Level*	5		
Comments	-	*Delay*	-1		
Confidence[14]	2	*Published Millis*	1512322497351.0		
Description	Olie op weg	*Last updated Millis*	1512322497352.0		
GeoPoint	[4.877815, 51.823144]				
Published Millis	1512322497351.0				

Table 3-2: Waze attributes overview with examples

As Waze users are able to link their Twitter account to their Waze account, we decided to also collect all Dutch Waze related data on Twitter. Automated Waze tweets contain either information on traffic events posted by the user or a summary of the car ride of the user. Only tweets containing information on traffic events are useful for this study and are therefore collected based on the bold tweet format as shown below.

Automated Waze tweet:

> ***Hielp chauffeurs in de omgeving door het melden*** *van wegwerkzaamheden op de N209 - Nieuwe Hoefweg, Bleiswijk* ***via @waze - social navigation.*** *(EN:* ***Helped nearby drivers by reporting*** *roadworks on the N209 – Nieuwe Hoefweg, Bleiswijk* ***on @waze – Drive Social.****)*

[13] Reliability score based on the experience level of the user. Users gain experience levels by contributing to the map, from level 1 to level 6. The higher the level, the more experienced and trustworthy the user. The score ranges between 0 and 10, with 10 being the most reliable.

[14] Confidence score based on how other users react to the report - either with a "Thumbs up" to indicate the alert is accurate or "Not there" if the report is irrelevant. The score ranges between 0 and 10, and a higher score indicates more positive feedback from Waze users.

3.1.3 TomTom

TomTom is a company that produces traffic, navigation and mapping products. It enables developers to use their APIs and SDKs to enhance applications with search, routing, mapping, traffic and navigation features. For our research, the TomTom data should either contribute to geosocial data or traffic data. This leaves us with two possible useful data sets that they offer: Online Traffic Incidents[15] and Online Traffic Flow[16].

The Online Traffic Incidents data can enrich our geosocial data set, in the same way Waze does. This service provides information on traffic incidents inside a given bounding box, updated every minute. This information is generated from anonymous real-time location trace information from connected GPS devices in vehicles, including personal navigation devices, in-dash navigation systems, smartphones and fleet management devices. This technique, called Floating Car Data (FCD), is able to measure traffic conditions on the road by using the previously described location-aware devices[17]. We contacted the developer relations helpdesk from TomTom to get more information regarding how traffic incidents are derived from their FCD technique. Unfortunately, they were not allowed to provide any specifics. We therefore make the assumption that TomTom incident data is provided by "real road-users", and thus could provide a valuable geosocial data source for this work.

The Online Traffic Flow data service, provides information about the speeds and travel times of the road fragment closest to the given coordinates. It is designed to work alongside the Flow layer of the Maps API, in order to support clickable flow data visualizations. After experimenting with this data source, we came to the conclusion that it could not be used in the way we would have liked. This because we can only provide one set of coordinates, which their system uses to map to the closest road and based on the provided zoom level it returns the values for a road section with a maximum of 1 kilometer. It is therefore impossible to scale this to get the flow data for the entire Netherlands, which is why we decided to only use the Online Traffic Incidents data from TomTom.

An overview of the most important properties and incident categories from the TomTom data we collect in JSON format is shown in Table 3-3. On first sight, compared to Waze, TomTom seems to have a more limited non-inheritance based traffic event typing (incident category) system, containing the following types: *Fog, Rain, Ice, Wind, Flooding, Accident, Dangerous Conditions, Jam, Lane Closed, Road Closed, Roadworks,* and *Detour.* However, we noticed after some experimenting with the data that TomTom incidents sometimes also contain descriptions and causes of the incident. By contacting the developer relations helpdesk, we learned that these incident descriptions and causes are part of a set of 443 incident categories (note that these can be used interchangeably as description and cause). These are nowhere mentioned in the documentation of this service, but have been provided to us by the helpdesk. We can therefore see the incident category as the main category and the description/cause as a subcategory of the event.

[15] developer.tomtom.com/online-traffic/online-traffic-documentation/online-traffic-incidents

[16] developer.tomtom.com/online-traffic/online-traffic-documentation/online-traffic-flow

[17] https://www.tomtom.com/en_gb/traffic-news/traffic-incidents

TomTom Online Traffic Flow	
Attributes	**Example**
GeoPoint	[4.877815, 51.823144]
Incident category	6
Magnitude of delay	5
Description of incident	Slow traffic
Cause of incident	Accident
Start point	Deil (N327)
End point	A2: Geldermalsen - A2 (N327)
Caused time delay in seconds	231
Affected road numbers by the incident	N327
Retrieval date	2017-12-05T13:05:18.179Z

Table 3-3: TomTom Online Traffic Flow properties

3.1.4 DiTTLab

The Delft integrated Traffic & Travel Laboratory (DiTTLab), is a research lab at the TU Delft that works with traffic data and simulation models to develop knowledge and tools for the international traffic and transport community. This lab provides us with traffic data that contains raw and interpolated speed and flow values per 100 meter segments for each motorway (e.g., A10) in the Netherlands. The raw traffic data is collected by roadway-based sensors. As these are irregularly spread along the highways, the data is interpolated to cover consistent 100 meter segments. The data is collected within the DiTTLab and distributed to us in JSON formatted files.

3.2 Data Pre-processing

In this section, we show how each data set is cleansed and transformed so that it is suitable to extract information out of it in the next phases of the pipeline.

3.2.1 Twitter

Before we start our pre-processing approach, we label the Twitter data as traffic event-related (TE) and non-traffic event-related (NTE). Based on the many pre-processing techniques used in previous work, we make a selection of pre-processing techniques for the Twitter data. Table 3-4 shows the pre-processing techniques used in previous work and shows the arguments on why a technique is applied, replaced or rejected in our work.

Technique	Apply/Reject/ Replace	Reason
Tokenization	Apply	Demarcation of sections of a string of input characters is needed for all other forms of processing.
Stop word removal	Apply	Removing the most common words in a language is needed to improve performance while keeping the words with the highest importance.
Stemming	Replace with Lemmatization	The goal of stemming and lemmatization[18] is to reduce inflectional forms and derivationally related forms of a word to a common base form. We replace stemming with lemmatization, as stemming chops off the ends of words and often includes the removal of derivational affixes. Lemmatization uses a vocabulary and morphological analysis of words, to remove inflectional endings only and to return the base or dictionary form of a word, called a lemma.
Discarding non-geo tweets	Reject	Approximately 1% of the tweet dataset is geo-tagged, discarding all other tweets leaves us with a too small data set to work with.
Abbreviation replacement	Reject	Automatic correction of abbreviations could lead to incorrect words, increasing possible false positives.
Locational/ Temporal generalization	Reject	Locational/temporal features are used later on in the pipeline.
Discarding hashtags, links, mentions, special characters, words based on length	Reject	Links are already removed during the data collection process. Hashtags, mentions, special characters, and words of all lengths are able to provide us valuable information and are therefore not discarded.
Discarding non-nouns based on POS Tagger	Reject	Words other than nouns can still provide valuable information and are therefore not discarded.

Table 3-4: Pre-processing techniques selection

[18] https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html

3.2.2 Waze and TomTom

Both data sources from Waze and TomTom have little pre-processing needs. Attributes that do not contain any descriptive value will be omitted and attribute terms are made uniform between the datasets.

3.2.3 DiTTLab

DiTTLab provides us with raw and interpolated traffic speed and flow data. In this study, only the interpolated data is used, as this provides us with a higher possible anomaly coverage. This data could be used as a source for traffic event detection algorithms. However, as traffic event detection algorithms greatly depend on the type and properties of the road, it is not feasible to implement this for every motorway in the Netherlands. Besides, this would fall out of the scope of this research. Therefore, we only process the data in such way that it can be stored and accessed in and from our document database. In order to achieve this we store the data by road number, road side, road location and date.
Table 3-5, shows an example of how DiTTLab data will be stored in our database.

Key	Value	Type
_id	5afc2974e8b2900e404e9ce6	ObjectId
roadNumber	A1	String
roadSide	R	String
roadLocation	{ 2 fields }	Object
x	4.959109	Double
y	52.346883	Double
roadData	[2879 elements]	Array
0	{ 3 fields }	Object
date	2017-12-05T00:00:00.000Z	Date
speed	101	Int32
flow	398	Int32

Table 3-5: DiTTLabReportCollection

3.3 Traffic Event Categorization

In this section, we discuss the design of our rule-based traffic domain annotator, inspired by the work of Oostdijk, Hürriyetoglu, Puts, Daas, and van den Bosch (2016). This annotator is used for extracting relevant traffic domain information from tweet text data. This enables us to categorize a tweet to one or multiple traffic event categories.

3.3.1 Category Composition

This annotator enables us to automatically identify tokens belonging to multiple traffic domain related categories within tweets. The categories that are used to label the tweets are based on the event categories from Waze and TomTom, the categories in the police accident reporting dossier (Bestand geRegistreerde Ongevallen Nederland (BRON[19])), and acquired knowledge from reviewing literature and annotating tweets. The event typing method from Waze and TomTom can be found in Sections 3.1.2 and 3.1.3. BRON contains information on causes and effects of traffic events. Table 3-6, describes the categories within BRON that are relevant to this study.

Categories	Description
Lane subtypes	All types of lanes within the road network, e.g., entry, service lane, bus lane.
Vehicle details	All types of vehicle brand, and sub brand names and their measurements, e.g., Volkswagen, Volkswagen Polo, and Skoda.
Points of collision	Point of impact from a collision on a vehicle, e.g., left side, center rear.
Nature of accident	Anything that could cause the accident, e.g., animal, parked vehicle, fixed object.
Outcomes	Outcomes for people and vehicles involved in the accident, e.g., injury, material damage.
Movements	Movement of involved vehicles during the accident, e.g., rollover, overturning, skidding.
Particulars	Any particulars during the accident such as nearby infrastructure or road types, e.g., bridge, speed bump, overtake prohibition.
Devices	Any type of vehicle involved in the accident, e.g., tipping wagon, taxy, caravan.
Light conditions	Conditions of light that could have been the cause of the accident, e.g., daylight, darkness, twilight.
Manoeuvres	Any type of movement of vehicle or person that caused the collision, e.g., collision with lose object, head-tail collision when turning to the right.
Object types	Any type of object that could cause the accident, e.g., tree, bike, bus.
Circumstances	Any circumstance related to driving or the driver that caused the accident, e.g., not giving right of way, ignoring a red traffic light, high-speed.
Road surfaces	Status of the road surface that could have caused the accident, e.g., dry, wet, snowy.
Road Situations	Type of road on which the accident occurred, e.g., crossroads, roundabout.
Road Surfacing	Type of road surface at the place of the accident, e.g., concrete, asphalt.

Table 3-6: BRON relevant categories

[19] https://www.rijkswaterstaat.nl/apps/geoservices/geodata/dmc/bron/

The following procedure is followed to compose the list of traffic domain categories. Firstly, a category from Waze is selected, as Waze has the most coherent traffic event taxonomy. Secondly, a related TomTom category is selected. Thirdly, categories from BRON that are related to the category are selected. Finally, a custom category with related sub categories is formed based on the categories from Waze, TomTom and BRON. Categories that are not described in Waze, TomTom or BRON, but are deemed relevant are added, and any overlapping categories are merged. Note that events within Waze and TomTom depend on a location and datetime, while approximately, only 1% of the traffic event-related tweets contains a geotag. Therefore, additional place based categories are defined that could be mapped to a location. In order to infer a datetime, a combination of the creation date of a tweet and temporal expressions in a tweet is used, as the creation date by itself is not necessarily a reflection of the date a traffic event occurred. By following this category composing procedure a set of 27 (not counting the not applicable, temporal, and media attachment categories) unique traffic related categories has been composed. For each category we explain the idea behind it, were it derived from (Waze, TomTom, BRON, literature), and provide an example of the sort of tokens it should describe. Note, that we do not make a distinction between positive/confirming categories and negative/disconfirming categories. For example, the token set "A man was injured" and "No man was injured" both get assigned the category Road User Casualty.

1. **No Applicable Category (N/A):** Describes tokens that are not matched by the other categories.
   Derived from: Literature.
   Example: @joopb68flc we staan bij brug Zaltbommel in file. (EN: @joopb68flc we stand near bridge Zaltbommel in traffic jam.)
2. **Media Attachment:** An indication of a media link.
   Derived from: Literature.
   Example: Ongeval 2 personenwagens . Snel bergen . #A16 Li 16,9 https://t.co/ovmSUIHLMv (EN: Accident 2 passenger cars. Quick salvage. #A16 Le 16,9 https://t.co/ovmSUIHLMv)
3. **Temporal (Timex):** An indication of time, a point in time, a time duration, or a time frequency.
   Derived from: Waze, TomTom, BRON, Literature.
   Example: Ik sta al 30 minuten in de file richting Den Haag. (EN: Standing in a traffic jam for 30 minutes in the direction of Den Haag.)
4. **Advice:** A mention of an announcement or guidance.
   Derived from: TomTom, Literature.
   Example: Pas je snelheid aan er heeft net een ongeluk plaatsgevonden op de A10. (EN: Adjust your speed, an accident just happened on the A10.)
5. **Road User Transport:** Various types of groups of traffic.
   Derived from: Literature.
   Example: Veel vakantieverkeer richting Amsterdam vandaag. (EN: Lots of holiday traffic in the direction of Amsterdam today.)
6. **Road User Casualty:** Various types of injuries and casualties.
   Derived from: BRON.
   Example: Meerdere inzittenden ernstig gewond bij kettingbotsing op de A10. (EN: Multiple passengers seriously injured at chain collision on the A10.)
7. **Road User Traffic:** Information that describes traffic related persons and their road user role (e.g., driver, passenger).
   Derived from: Literature.
   Example: Bestuurder onwel geworden achter het stuur, politie is gearriveerd #A2. (EN: Driver unwell behind the wheel, police has arrived #A2).
8. **Road User General:** Information that describes general persons.
   Derived from: Literature.

Example: Persoon onwel geworden achter het stuur, politie is gearriveerd #A2. (EN: Person unwell behind the wheel, police has arrived #A2).

9. **Road User Vehicle:** Various types of vehicle names and their brands.
   Derived from: BRON.
   Example: Reed net voorbij een ongeluk met een Audi en een vrachtwagen op de A1. (EN: Just drove past an accident between an Audi and a lorry on the A1.)
10. **Road User Emergency Service:** Various types of emergency services and their status.
    Derived from: TomTom, Literature
    Example: Ongeval bij knooppunt Amstel politie is ter plaatse. (EN: Accident at junction Amstel police is on location.)
11. **Place Location:** Exact locations in the Netherlands that contain a geopoint, geoline, or geoshape.
    Derived from: TomTom, Waze, BRON, Literature.
    Example: @RWS_verkeer En we rijden weer zeeburgertunnel is weer open. (EN: @RWS_verkeer And we're driving again zeeburgertunnel is weer open.)
12. **Place Location Combination:** Combination of areas having unique physical and human characteristics, and locations.
    Derived from: TomTom, Waze, BRON, Literature.
    Example: Ongeluk voor de rotonde Vliegveldweg. (EN: Accident in front of the roundabout Vliegveldweg.)
13. **Place Road Section:** Section of a road containing a start and end point, indicated by places and locations.
    Derived from: TomTom, Waze, BRON, Literature.
    Example: Er staat een file van knooppunt Coenplein tot Zaandam. (EN: There's a traffic jam from junction Coenplein to Zaandam.)
14. **Place Road Direction:** Combination of directional terms and a location.
    Derived from: TomTom, Waze, BRON, Literature.
    Example: 5km file Delft richting Den Haag. (EN: 5km traffic jam Delft in the direction of Den Haag.)
15. **Place Road Mile Marker:** Place on the road denoted with a mile marker.
    Derived from: TomTom, Waze, BRON, Literature.
    Example: Gat in wegdek #A58 re 13.4 afrit Middelburg. (EN: Pothole #A58 ri 13.4 exit Middelburg.)
16. **Place Infrastructure Type:** Various types of road infrastructures.
    Derived from: TomTom, Waze, BRON, Literature.
    Example: File voor de brug, heb ik weer #Delft. (EN: Traffic jam in front of the bridge, just my luck #Delft.)
17. **Place Road Lane:** Further specification of road strips.
    Derived from: TomTom, Waze, BRON, Literature.
    Example: Olie op de vluchtstrook nabij Utrecht. (EN: Oil on emergency lane near Utrecht.)
18. **Event Accident:** Anything related to traffic collisions (including consequences) between vehicles and other vehicles, pedestrians, animals, road debris, or other stationary obstructions.
    Derived from: TomTom, Waze, BRON
    Example: Mercedes van de weg geraakt bij knooppunt Amstel. (EN: Mercedes of the road at junction Amstel.)
19. **Event Traffic Jam:** Traffic jam terms and indicators of a traffic jam, e.g., traffic flow/ intensities and durations.
    Derived from: TomTom, Waze.
    Example: Korte file van 10 minuten voor de Kuip. (EN: Short jam of 10 minutes in front of de Kuip.)
20. **Event Closure:** Anything related to a road being closed off.
    Derived from: TomTom, Waze.
    Example: Doorgaand rijverkeer gestremd in de richting van Delft-Noord. (EN: Through traffic obstructed in the direction of Delft-Noord.)

21. **Event Enforcement:** Activities held by traffic enforcement agencies.
Derived from: TomTom, Waze, Literature.
Example: Alcoholcontrole op de A2 richting Den Bosch. (EN: D.U.I. checkpoint on the A2 in the direction of Den Bosch.)
22. **Event Hazard Violation:** Road activities that violate the law.
Derived from: TomTom, Literature.
Example: Auto achter me loopt irritant te bumperkleven #A5. (EN: Car behind me is annoyingly tailgating me #A5.)
23. **Event Hazard Traffic Sign:** Indicators of broken, unreadable, or missing traffic signs.
Derived from: TomTom, Waze.
Example: @VID Defect matrixbord boven de rechter rijbaan hmp 56.1. (EN: @VID Defect matrix signal above the right lane hmp 56.1.)
24. **Event Hazard Traffic Light:** Indicators of malfunctioning or broken traffic lights.
Derived from: TomTom, Literature.
Example: Stoplichten op hol geslagen bij kruispunt Sloeweg. (EN: Trafficlights out of control at crossroads Sloeweg.)
25. **Event Hazard Weather:** Bad weather conditions that could have an effect on the traffic speed and flow, and sight of road users.
Derived from: TomTom, Waze.
Example: Dichte mist op de A2 zie geen hand voor ogen! (EN: Dense fog on the A2 can't see a thing!)
26. **Event Hazard Stopped Vehicle:** Indicators of a stopped vehicle, due to a breakdown.
Derived from: TomTom, Waze.
Example: Stilstaande auto met rookontwikkeling op de vluchtstrook bij afrit Goes. (EN: Stopped car with smoke on the emergency lane at exit Goes.)
27. **Event Hazard Roadwork:** Indicators of unplanned roadwork activities.
Derived from: TomTom, Waze.
Example: Rechterrijbaan tussen Souburg en Vlissingen afgesloten vanwege spoedreparatie aan het wegdek. (EN: Right lane between Souburg and Vlissingen closed off because of emergency repair on the road surface.)
28. **Event Hazard Object:** Foreign objects and road debris that could cause dangerous situations.
Derived from: TomTom, Waze.
Example: Boom omgewaaid op de rechterrijbaan bij station Delft. (EN: Tree blown down on right traffic lane near station Delft.)
29. **Event Hazard Animal:** Stray animals or roadkill that could cause dangerous situations.
Derived from: TomTom, Waze.
Example: Tussen Nijkerk en Amersfoort ligt langs de A28 een overreden kat. (EN: Between Nijkerk and Amersfoort lays a run over cat besides the A28.)
30. **Event Hazard Road Condition:** A hazardous condition to the road surface.
Derived from: TomTom, Waze.
Example: Gaten in wegdek #A5 li 13.24 afrit Middelburg. (EN: Potholes in road surface #A5 le 13.24 exit Middelburg.)

3.3.2 Grammar

Having established this set of categories, we can focus on the characteristics of the grammar and dictionaries behind each category. Our method uses a combination of place names, temporal expressions, traffic domain knowledge, and lexical pattern dictionaries. First, the place names dictionary, which is composed out of place names for the Netherlands from the GeoNames database, and the tagging system from OpenStreetMap[20] . Note that the GeoNames database is only used as a source for place names, and is not used to derive any locations (coordinates). The system from OpenStreetMap enables us retrieve the names of the following features:

- Highway: Names of roads for the entire road network of the Netherlands, e.g., bridges and residential roads.
- Amenity: Names of facilities used by visitors and residents, e.g., colleges and parking.
- Building: Names of buildings, e.g., warehouses and churches.
- Leisure: Names of leisure and sports facilities, e.g., parks and sport stadiums.
- Place: Names of settlements, e.g., suburbs and towns.

The advantage of using such a comprehensive dictionary is that we significantly increase our possibilities of finding a locational term in a tweet. However, we do acknowledge that this library contains a lot of ambiguous terms that could be either a location name or a common word used in the Dutch language. Additionally, using such a large dictionary will most certainly also significantly increase the computation time of the annotator. We decided however that being able to map a traffic event-related tweet, with the risk of mapping it to a wrong location, is more important than not being able to map it due to a too limited dictionary. In order to mitigate false positives due to ambiguous location terms, a dictionary is manually composed based on these types of encounters. Second, the temporal expressions dictionary, which is composed of the temporal expressions dataset from the work by Hürriyetoglu, Oostdijk, & van den Bosch (2014). This dictionary contains tokens and phrases that serve to identify time intervals, e.g., "vanmiddag om 16.20 uur" (EN: this afternoon at 16.20). Third, multiple traffic domain knowledge based dictionaries, e.g., vehicle names/brands, road debris, and emergency services. These dictionaries are composed of terms from the BRON dataset, the national scientific institute for road safety research in the Netherlands (SWOV) traffic terms set[21], and synonyms/ colloquial speech derived from these sets. Additionally, this dictionary is updated manually, based on relevant terms encountered in traffic event-related tweets. Finally, lexical pattern dictionaries, that consist of non-traffic related terms that occur within traffic event-related tweets with high frequency.

These resources are used to create a Backus-Naur form (BNF) grammar, allowing for partial matching of tokens. The grammar allows for case-insensitive matching as tweets are known for having an inconsistent usage of capitals. Additionally, the grammar includes rules to recognize spelling variations of domain terms, by matching on suffixes. For each of the 27 defined traffic categories, key terms and syntactic knowledge for that particular category are defined. We include optional term matching and linguistic structures such as adjectives to restrict ambiguous terms from matching excessively. In practice this means that some traffic domain knowledge terms can be used by themselves, while other terms need to have preceding/succeeding terms. For instance, for the category Event Hazard Weather the term

[20] https://wiki.openstreetmap.org/wiki/Map_Features

[21] https://www.swov.nl/publicatie/verkeersveiligheidstermen-nederlands-engels-en-engels-nederlands

"weer", which translates to both "weather" and "again", needs to be preceded with a weather type (e.g., "misty"), as it is too ambiguous by itself. The documentation of each category, with rules and examples can be found in Rule-based Traffic Domain Annotator Grammar. Below, we provide an example of one of the grammar modules for the "Event Hazard Roadwork" category. In order to understand the grammar rules we first explain the subparts each rule consists of. Rules are build out of operators, predefined grammar methods, and tokens from the resources.

Some of the commonly used operators are:

- **Token + Token:** concatenates two tokens with an interspace.
- **Token | Token:** exclusive or.
- **~Token:** disallows matching of this token.

Some of the commonly used grammar methods are:

- **WordStart:** matches if the current position is at the beginning of a word, and is not preceded by any character in a given set of characters, as well as at the beginning of a line.
- **WordEnd:** does the exact opposite of WordStart.
- **Optional(Token):** makes all tokens within the parentheses optional.

The grammar module in this example uses the following traffic domain related dictionaries:

- **Roadwork:** e.g., opruimingswerkzaamheden (clearing-up operations), spoedreparatie (emergency repair), onderzoek (examination)
- **Vehicle names:** e.g., auto (car), vrachtwagen (truck), caravan.
- **Vehicle brands:** e.g., Nissan, Volvo, Hobby.
- **Traffic lights:** e.g., stoplicht (traffic light), verlichting (lighting), lantaarnpaal (lamppost)
- **Traffic signs:** e.g., bewegwijzering (signage), matrixbord (matrix sign), wegmarkering (road marking)
- **Road lanes:** e.g., rijbaan (lane), vluchtstrook (shoulder), parallelrijbaan (parallel lane)

By establishing the definitions of all sub parts, the grammar rules become basically self-explanatory. We colored the tokens within the example to reflect the subparts of the grammar rules:

- **R1:** Optional(vanwegeLit | ivmLit | doorLit | tgvLit | alsgevolgvanLit |alsgevolgvanLit | metalsgevolgvanLit | naLit) + Optional(~Roadwork token + Arbitrary token) + Roadwork token + Optional(aanLit | vanLit | opLit | nabijLit | langsLit | bijLit | vlakLit| naastLit) + Optional(~(Vehicle names token | Vehicle brands token | Road lanes token) + Arbitrary token) + (Vehicles names token | Vehicle brands token | Road lanes token)

Example:

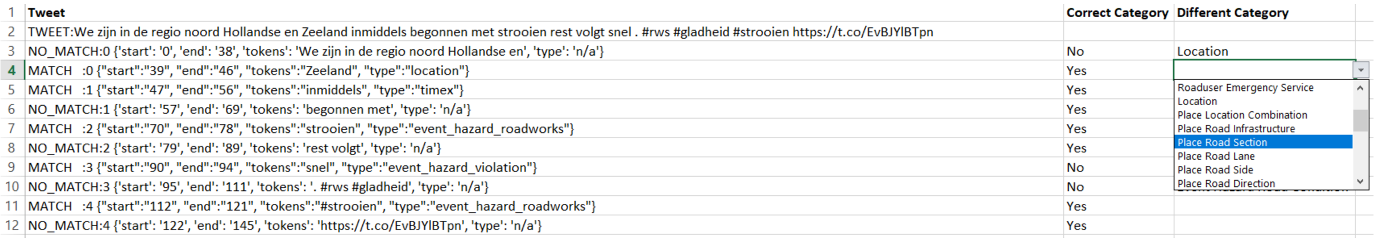
- **E1:** Vanwege langdurige spoedreparatie aan het wegdek bij afrit Delft-Noord. (EN: Because of prolonged repairs on the road surface at exit Delft-Noord.)

3.3.3 Evaluation

We evaluate our rule-based traffic domain annotator on a randomly selected sample of 200 annotated traffic event-related tweets. This set is omitted from the training phase of the traffic domain annotator in order to obtain an honest and unbiased assessment of the performance of the annotator. To prevent any personal bias in this evaluation phase, the evaluation is performed by two fellow ex-master students from our Web Information Systems research group, namely ir. Jan Zegers and ir. Alexander Grooff. Both will be assigned with an Excel file containing 100 annotated tweets, as partly depicted in Figure 3-4. For each annotated set of tokens within a tweet the following questions have to be answered:

1. Is the correct category assigned to the token set?
2. If a wrong category is assigned, what other category (from the predefined category list) should have been assigned to the token set instead.

By answering these two questions, we will be able obtain the number of false positives/negatives and true positives/negatives for each category. Thereby, we can find out what categories should had been assigned in the case a false positive/negative is found. As natural language is ambiguous and thereby open to interpretation, the assessor has a third option "Not Sure", besides "Yes/No" when answering if the correct category has been assigned to the set of tokens.



1	Tweet	Correct Category	Different Category
2	TWEET:We zijn in de regio noord Hollandse en Zeeland inmiddels begonnen met strooien rest volgt snel . #rws #gladheid #strooien https://t.co/EvBJYlBTpn		
3	NO_MATCH:0 {'start': '0', 'end': '38', 'tokens': 'We zijn in de regio noord Hollandse en', 'type': 'n/a'}	No	Location
4	MATCH :0 {"start":"39", "end":"46", "tokens":"Zeeland", "type":"location"}	Yes	
5	MATCH :1 {"start":"47", "end":"56", "tokens":"inmiddels", "type":"timex"}	Yes	
6	NO_MATCH:1 {'start': '57', 'end': '69', 'tokens': 'begonnen met', 'type': 'n/a'}	Yes	
7	MATCH :2 {"start":"70", "end":"78", "tokens":"strooien", "type":"event_hazard_roadworks"}	Yes	
8	NO_MATCH:2 {'start': '79', 'end': '89', 'tokens': 'rest volgt', 'type': 'n/a'}	Yes	
9	MATCH :3 {"start":"90", "end":"94", "tokens":"snel", "type":"event_hazard_violation"}	No	
10	NO_MATCH:3 {'start': '95', 'end': '111', 'tokens': '. #rws #gladheid', 'type': 'n/a'}	No	
11	MATCH :4 {"start":"112", "end":"121", "tokens":"#strooien", "type":"event_hazard_roadworks"}	Yes	
12	NO_MATCH:4 {'start': '122', 'end': '145', 'tokens': 'https://t.co/EvBJYlBTpn', 'type': 'n/a'}	Yes	

Roaduser Emergency Service
Location
Place Location Combination
Place Road Infrastructure
Place Road Section
Place Road Lane
Place Road Side
Place Road Direction

Figure 3-4: Rule-based Traffic Domain Annotator Evaluation

3.4 Feature Engineering

Feature selection is applied to the Twitter data. Based on the many feature engineering techniques used in previous work, we make a selection of the most relevant and useful features, as shown in Table 3-7. The selected features will be used by our traffic event classifier as described in the next step of the pipeline, to indicate if a tweet is traffic event-related or not. The main objective of our feature selection approach is to improve the prediction performance of the classifier, providing faster and more cost-effective predictors, and to better understand the underlying process that generated the data (Guyon & Elisseeff, 2003). For this process, we will use Scikit-learn[22], a free software machine learning library for Python in combination with the Frog NLP library.

Technique	Reason
Syntactic features	Exclamation/question marks, emoticons, and the total number of capital characters are part of the syntactic features. These features could indicate sentiment characteristics about the tweet.
Bag of words/n-grams	This process turns a collection of tweets into numerical feature vectors and is part of vectorization. Tweets are described by their word occurrences while ignoring the relative position information of words in the tweet. We will extract 1-grams (bag of words). However, this bag of words approach has its limitations. It is not able to capture phrases and multi-word expressions, thus effectively disregarding any word order dependence. This approach also does not account for misspellings/ word derivations. Therefore, we additionally use word bigrams to preserve some of the local ordering of information, as well as character bigrams as a solution against misspellings and derivations.
TF-IDF term weighting	Term frequency-inverse document frequency shows the importance of a term to a document in the corpus. The term-frequency (TF) resembles the amount of times a term is located in a document. The document frequency (DF) denotes the number of documents that contain this term. To measure the uniqueness of a term, the infrequency of the term occurring across documents is needed, in other words, the inverse document frequency (IDF). A high result of the product of TF and IDF shows that the term occurs frequently in the document and provides the most information about that document.
Traffic domain types	In the previous section, we showed how our traffic domain annotator is able to tag 27 different categories. These word categories thus appear as word n-grams or character n-grams in our model and can all be regarded as features. This includes temporal and locational tags that were often used as features in previous work.
POS tagging	POS features, e.g., nouns and verbs, could be used to extract syntactic and linguistic representation out of the tweets. A POS tagger is able to tag words with different part of speech labels. This feature will be used in the same way as the traffic domain types. We must state, that due to the limited amount of characters in tweets, the effectivity of a POS feature is uncertain.

Table 3-7: Feature selection

[22] Scikit-learn.org

3.5 Traffic Event Classification

In order to predict if a tweet is related to a traffic event, supervised binary classification is applied, in which we classify a tweet to either the traffic event-related (TE) or non-traffic event-related (NTE) group. We first manually label each tweet in our dataset, collected with the data collection approach from Section 3.1.1, with a TE or NTE label. The next step is to choose a classification method based on the size of the dataset, number of features, and previous work related to supervised binary classification of tweets. Commonly used methods for binary classification consist of decision trees, random forests, Bayesian networks, support vector machines, neural networks, and logistic regression. When trying to predict a category and working with a dataset of less than 100-thousand text-based tweets, it is advised by the documentation of sci-kit learn (the machine learning library that we will be using), to use either a Support Vector Machine (SVM) or Naïve Bayes (NB) based method. During our literature review, we also found that methods based on SVM and NB were the most used and proved to deliver the best results when working with tweets. Based on this information we opt to start the traffic event classification approach with these two types of algorithms.

3.5.1 Support Vector Machine Theory

SVMs consist of multiple supervised learning algorithms for classification, regression and outlier detection. In this work, we focus on the classification methods SVM offers. The idea behind SVM is finding a hyperplane that best divides a data set into two classes. This is achieved by plotting the data items from the dataset as a point into an *n*-dimensional space, where *n* represents the number of features. The coordinates of the points correspond to the value of each feature. The further away data points (support vectors) lie from the hyperplane the more confident we can be about the classifier performance. In SVM theory the term margin is used for the distance between the hyperplane and the nearest support vector from either set. Therefore, the goal is to find the hyperplane with the greatest possible margin between the hyperplane and any support vector within the training set (Joachims, 1998).

3.5.2 Naïve Bayes Theory

Naïve Bayes typed classifiers apply the Bayes' theorem, where the naïve part describes the independence assumptions between the feature values. So for example, a vehicle may be considered to be a school bus if it is yellow, longer than 5 meters and has 4 wheels. A NB classifier considers each of these features to contribute independently to the probability that the vehicle is a school bus, regardless of any correlations between the features. Which is why this algorithm is called naïve, as features are not always independent of each other. The Bayes' theorem itself describes how to update the probabilities of hypotheses when provided data. Given a hypothesis H and data D, the theorem states that the relationship between the probability of the evidence $P(D)$ and the probability of the hypothesis after getting the evidence $P(H/D)$ is:

$$P(H|D) = \frac{P(D|H)}{P(D)}\, P(H)$$

Where:

- *P(H/D)* is the probability of hypothesis *H* given the data *D*, called the posterior probability.
- *P(D/H)* is the probability of data *D* given that the hypothesis H was true.
- *P(D)* is the probability of the data.
- *P(H)* is the probability of hypothesis *H* being true, the so called prior probability of *H*.

The NB classifier combines the probability model of the Bayes' theorem with the maximum a posteriori (MAP) decision rule:

$$MAP(H) = \max(P(D|H) * P(H))$$

This provides us with the numerator and the class giving the largest response, being the predicted output (Murphy, 2006).

3.5.3 Model Selection and Evaluation

To determine which of our models has the best performance, we have to compare them based on the same evaluation techniques, which are discussed in this section.

3.5.3.1 Resampling of Dataset

The dataset to be used in the experiment is very likely to be imbalanced, as the number of collected NTE tweets will always outweigh the TE tweets. Therefore, we have to consider that some metrics can give a misleading picture. Take for example a dataset with a TE/NTE tweet ratio of 1:9. In this case, the accuracy score is misleading because when the classifier always predicts the most common class without performing any analysis of the features, it will still have a high accuracy rate of 90%. We therefore, use multiple types of metric scores to evaluate our model. Additionally, resampling is applied, which is a widely adopted technique for dealing with highly unbalanced datasets. Resampling consists of under-sampling and over-sampling techniques. With under-sampling, records from the over-represented class are removed, while with over-sampling copies of records from the under-represented class are added. We apply over-sampling based on three popular methods: random over-sampling technique, SMOTE (Synthetic Minority Oversampling Technique), and ADASYN (Adaptive Synthetic sampling method). Under-sampling is applied based on a random under-sampling technique, and Cluster Centroids (Chawla, 2009).

3.5.3.2 Cross-validation

K-fold cross-validation is applied to estimate how accurately the model performs in practice (on out-of-sample data) and to prevent overfitting. Overfitting is the situation in which a model is trained and tested on the same data (or closely related data), and therefore fails to fit additional data or provide reliable future observations. A common solution to this problem is to split the dataset into a train and a test set, so that the model can be trained and tested on different data. In the case of cross-validation, a bunch of these train/split sets is created. The training set is split into *k* smaller equal sets called folds. For each of these folds a model is trained using *k-1* of the folds as training data, while the union of the other folds is used as the training set. The average of the values computed in the loop is used as a performance measure. This is a way more accurate estimate of out-of-sample performance can be gained, and we use our data more efficiently as every observation is used for both training and testing. Initial findings show that our dataset exhibits a large imbalance in the distribution of the target classes. We therefore, have to ensure that relative class frequencies are approximately preserved in each fold. To this end, we apply a variation of k-fold cross validation called stratified k-fold. This ensures that each fold contains approximately the same percentage of each target class as the complete set.

3.5.3.3 Hyper-parameter Tuning

Now that we have compensated for any possible overfitting, we can focus on tuning the hyper-parameters of the classifiers we use. Hyper-parameters, express properties of the model that cannot be directly learned from the regular training process. These types of parameters define higher level concepts and influence the predictive of computation performance of the model. Examples of important hyper-parameters, which we will tune, for SVM based classifiers are:

- **Kernel:** The used kernel method for pattern analysis. A kernel function returns the inner product between two points in a suitable feature space. Examples of kernels are: linear, polynomial, rbf, and sigmoid.
- **C:** Penalty parameter of the error term. A large *C* corresponds to a smaller-margin separating hyperplane in the case that hyperplane does a better job of getting all the training points classified correctly. A small *C* corresponds to a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.
- **Loss:** The hinge loss is used to determine the maximum margin classification.

In order to obtain the best combination of hyper-parameters, an exhaustive grid search will be applied. This grid search exhaustively generates candidates from a grid of user-specified parameter values, and evaluates all the possible combinations of these values.

3.5.3.4 Model Evaluation

A range of different evaluation metrics will be used to evaluate our models:

- **Precision:** the ability of the classifier not to label negative samples as positive.
- **Recall:** the ability of the classifier to find all the positive samples.
- **Accuracy:** the proportion of true results among the total number of examined cases.
- **F1:** the weighted harmonic mean of precision and recall, between 0 (worst score) and 1 (best score).

$$PPV = \frac{TP}{(TP + FP)}$$

$$NPV = \frac{TN}{(TN + FN)}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F1 = \frac{2TP}{(2TP + FP + FN)}$$

- **ROC AUC:** Area Under the Curve (AUC) of Receiver Operating Characteristic (ROC). In a ROC curve true positives are plotted against false positives at various threshold settings. A perfect classifier would have a ROC curve that goes straight up the y-axis and then along the x-axis. While a classifier with no power (by random guessing) will sit on the diagonal, and other classifiers falling in between, as illustrated in Figure 3-5. Therefore, the area under the curve shows a classifier with no power when its value is 0.5, and a perfect classifier at 1.0.
- **Precision-Recall curve:** this curve shows the tradeoff between precision and recall for different threshold values, as depicted in Figure 3-6. By changing the threshold values different precision-recall ratios can gained.

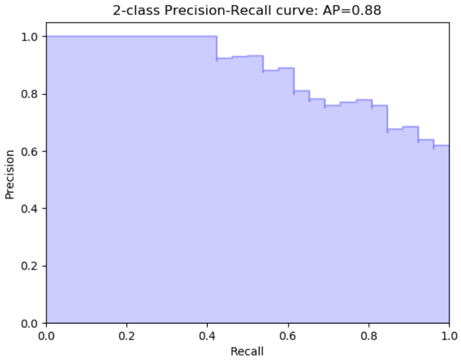


Figure 3-6: Precision-Recall Curve

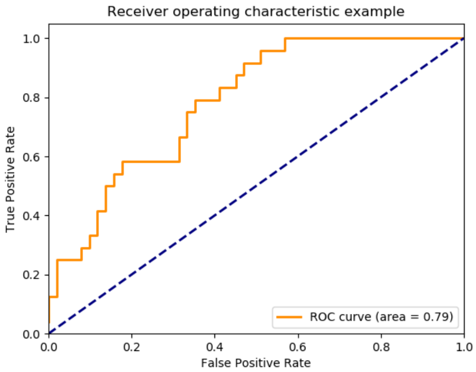


Figure 3-5: ROC Curve

3.6 Geocoding

In this section, we discuss how traffic event-related tweets can be linked to a location. A geocoding method is needed, as even though tweets can have their own location attribute (device location) in the form of a geopoint, this only occurs in very rare cases. Based on our initial findings, only 0.2% of the traffic event-related tweets contains a geopoint.

3.6.1 Approach

With the help of our annotator, as described in Section 3.3, we collect one or a multitude of spatial indicators from tweets. These spatial indicators are tagged with the following categories:

- **Place Location:** Exact locations in the Netherlands that contain a geopoint, geoline, or geoshape.
- **Place Location Combination:** Combination of "unnamed" areas, e.g., infrastructure and natural environments, and "named" locations, e.g., cities and street names.
- **Place Road Section:** Section of a road containing a start and end point, indicated by places and locations.
- **Place Road Direction:** Combination of directional terms and a location.
- **Place Road Mile Marker:** Right or left side of the road, in combination with mile marker number or road number.

These spatial indicators bring the following challenges with them:

- **Contradiction:** Spatial indicators can contradict each other, as they describe multiple places (e.g., "#N247 near Edam accident, traffic redirected from Monnickendam via A7").
- **Confirmation:** Spatial indicators can coincide and provide a more precise description of the event location (e.g., "Accident with multiple cars on A13 near exit Ikea").
- **Scale:** Spatial indicators can relate to different forms and scales. An indicator can be a geopoint, geoline or geoshape. And it can vastly differ in size, e.g., Amsterdam (city level) and Noord-Holland (province level).
- **Ambiguity:** Spatial indicators with the same name can be matched to different locations, e.g., "Michiel de Ruyterstraat" is a street name that appears in five different cities.

Therefore, we design a model that computes the intersections of a multitude of spatial indicators in a tweet. Figure 3-7, depicts a high-level view of the geocoding model. We will go through each step of the model, with the help of the following traffic event-related tweet example:

> **TE Tweet:** *"@vid vast op de #A4 thv McDonald's #Delft. Vermijd A4 richting Den Haag #File" (EN: @vid stuck on the #A4 near McDonald's #Delft. Avoid A4 in the direction of Den Haag #Trafficjam)*

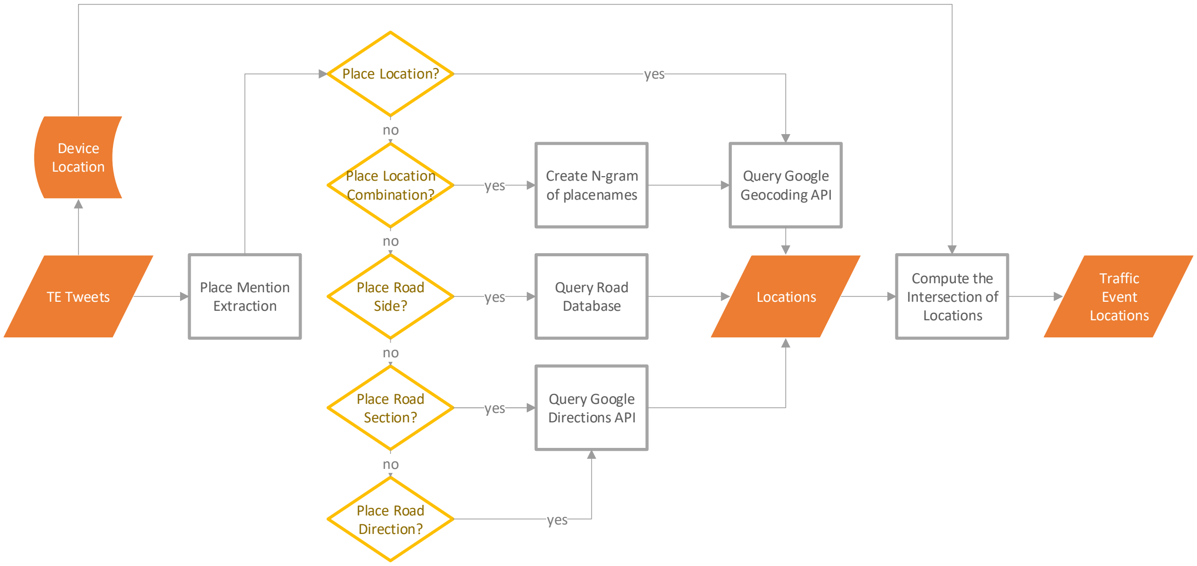


Figure 3-7: High-Level Geocoding Model

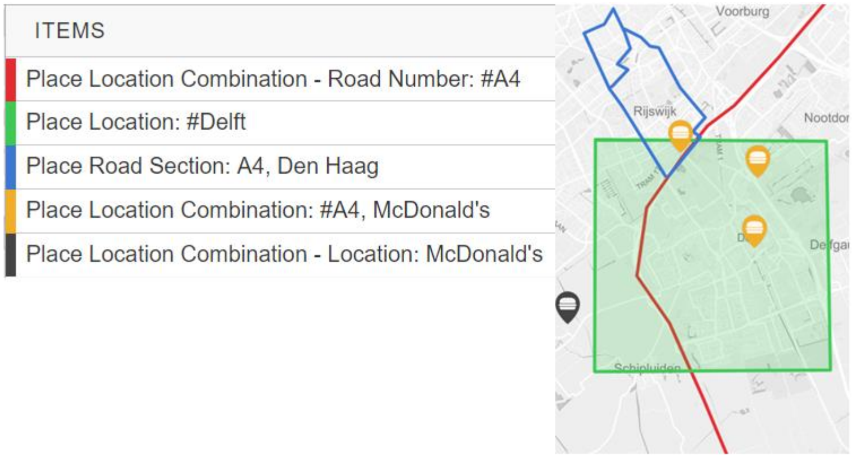
1. **Place mention extraction:** place related tokens are categorized into our five predefined place categories. Each token within a place category can also relate to one of the following place sub categories: Location, Road Number, Mile Marker Number, and Road Side. Place labels:
    - Place Location Combination: #A4 thv McDonald's
        - Road Number: A4
        - Location: McDonald's
    - Place Location: #Delft
        - Location: Delft
    - Place Road Section: A4 richting Den Haag
        - Road Number: A4
        - Location: Den Haag

2. **Geocoding**: a geocoding approach is defined for each place category. Note, that queries are restricted to be within the borders of the Netherlands. Any locational mentions of places outside of the Netherlands are therefore being discarded.
    a. **Place Location**: a place that per definition must have a location, and therefore can be used to query the Google Places API to retrieve a bounding box, if its subcategory is "Location". When its subcategory is "Road Number" or "Mile Marker Number", a query is made to our road database, as the Google Places API does not provide road number based geolines. Our road database consists of road numbers, mile marker numbers, road sides, and geopoints, based on data from Rijkswaterstaat Ministry of Infrastructure and Water Management[23].
    Example: #Delft = bounding box.
    b. **Place Location Combination**: a place and location that have some sort of relation to each other. Therefore, a combination of place tokens is used to query the Google Places API, except when one of the tokens is a "Mile Marker Number", as the Google Places API cannot work with this category. In that case the tokens are to query either the Google Places API or road database by themselves.
    Example: #A4 McDonald's = list of bounding boxes. #A4 = geolines. McDonald's = list of bounding boxes.
    c. **Place Road Mile Marker:** these token combinations can be used directly to query our road database, to retrieve one or multiple geopoints, or a geoline.
    d. **Place Road Section:** the combination of place indicators is used to query the Google Directions API to retrieve geolines (note that we dilate geolines retrieved from the Google Directions API and the road database with a radius of 100m, in order to compensate for multi-lane roads). Where the first token indicates the start of a geoline and the last token the end. If one of the tokens is not of the subcategory "Location", but of the category "Road Number" or "Mile Marker Number" the road database is queried for a geopoint instead. In the case of a "Road Number" the result could be a geoline, which cannot be used in the Google Directions API, therefore the centroid of this geoline is taken instead. This way a road section can be derived in every case.
    Example: A4 Den Haag → start: (A4, Den Haag), end: Den Haag = geolines.
    e. **Place Road Direction:** the token subcategory "Location", "Road Number", or "Mile Marker Number" defining the pointed to direction is used to query the Google Places API and road database by itself. Additionally, the closest preceding place category is used as a starting point, so that the Google Directions API can be queried, in the same way as the Place Road Section approach.
    f. **Device Location:** even though geotagged tweets are rare, the ones that are geotagged can provide valuable information.

[23] https://sites.google.com/site/hectometerpalendatanederland/

3. **Intersecting Locations:** the location linking approach results in a list of possible relevant locations per place category. However, we want to find the most relevant location(s), instead of linking the tweet to all possible locations that can be found in a tweet. In order to find the most relevant location(s), we make the following assumption: all spatial indicators within a tweet are of equal importance and add to the description of one or multiple event locations. Therefore, for each place category, we intersect the found locations with each other. Note that an additional radius of 250 meters is added to the places, to increase the chance of intersection. After testing a variety of radiuses on a selection of tweet reports, this radius provided the best balance in keeping the precision of the location without missing out on possible relevant intersections. Next, we intersect the results of these intersections with each other. This results in one or multiple intersected locations, which we define as the locations the traffic events in the tweet are most likely referring to. These locations based on rules *a*, *b*, and *c*, have been visualized in Figure 3-9. The location based on rule *d* is visualized in Figure 3-8. The following rules apply to each location:

   a. $Place\ Location\ Combination = (A4\ ,\ McDonald's) \cap (\#A4) \cap (McDonald's)$
   b. $Place\ Location = \#Delft$
   c. $Place\ Road\ Section = (A4\ , Den\ Haag)$
   d. $Traffic\ Event\ Location = [(Place\ Location\ Combination\ \cap$
   $Place\ Location), (Place\ Location\ Combination\ \cap\ Place\ Road\ Section),$
   $(Place\ Location\ \cap Place\ Road\ Section)]$


ITEMS
🟥 Place Location Combination - Road Number: #A4
🟩 Place Location: #Delft
🟦 Place Road Section: A4, Den Haag
🟨 Place Location Combination: #A4, McDonald's
⬛ Place Location Combination - Location: McDonald's
Voorburg
Rijswijk
Nootdor
Delfga
Schipluiden
TRAM 1

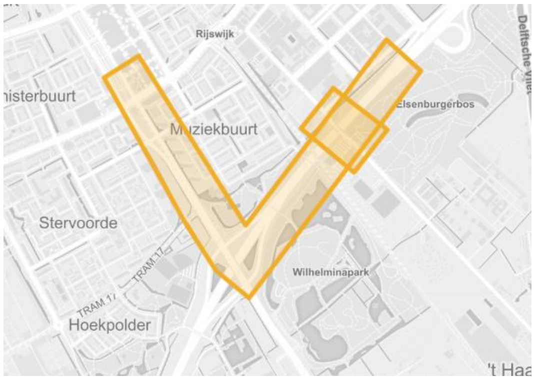


Figure 3-8: Location Linking Approach (d)

Figure 3-9: Location Linking Approach (a, b, c)

Our defined model of geocoding, enables us to transform traffic event-related tweets into one or multiple traffic event-related locations. The presented example shows how three locations can be extracted from a tweet containing two mentions of traffic jam events. It shows a solution for contradicting spatial indicators by allowing a tweet to be linked to multiple locations. Coinciding spatial indicators are also taken into account by the predefined annotation rules and intersection of places. The intersections also help with the scaling problem, as it scales a city level indicator (Delft) back to a smaller scaled polygon within Delft (Place Location Combination). Ambiguity is also tackled by the intersection, e.g., the ambiguous term McDonald's is only used in combination with less ambiguous locations as A4 and Delft.

3.6.2 Evaluation

We evaluate our geocoding approach on a randomly selected sample of 100 traffic event-related tweets. By using our geocoding approach, locations are calculated for each tweet. As place mentions in tweets are highly ambiguous, a geocoded location cannot be either correct or incorrect. Therefore, each tweet is evaluated on how well the geocoded locations suit the contents of the tweet, by ranking it into one of four categories:

Category 1: The geocoded result covers each place indicated in the tweet and includes no irrelevant locations, as shown in Figure 3-10.
Example: *#N201 hmp 28.0 rechts vrachtauto verzakt in de buitenberging . Rijstrook 2 afgesloten . @vid @ANWBverkeer https://t.co/sGAVpLHjTa*
(EN: N201 hmp 28.0 right truck subsided in the outside storage . Lane 2 closed . @vid @ANWBverkeer https://t.co/sGAVpLHjTa).

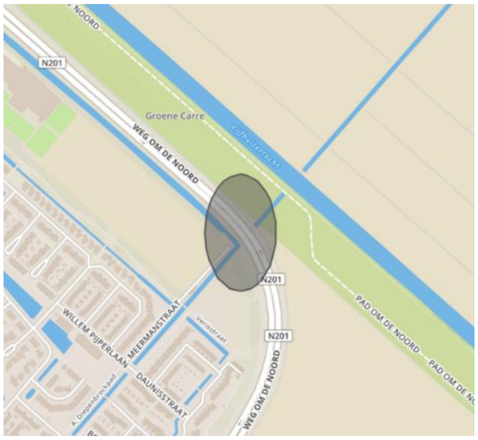


Figure 3-10: Geocoding evaluation category 1

Category 2: The geocoded result covers each place indicated in the tweet, but also includes irrelevant locations, as shown in Figure 3-11.
Example: *Afrit 7 en 8 #a28 zijn glad ! Utr ri Amersfoort (EN: Exit 7 and 8 #a28 are slippery ! Utr to Amersfoort)*

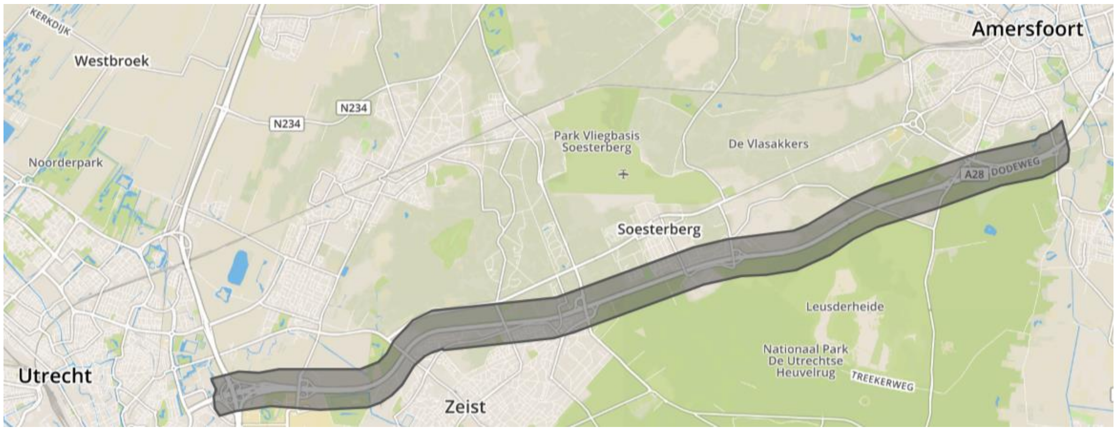


Figure 3-11: Geocoding evaluation category 2

Category 3: The geocoded result covers only part of the places indicated in the tweet, as shown in Figure 3-12.
Example: *Knijpbrug in #Hoogezand heeft er geen zin meer in , verkeer staat muurvast aan beide kanten @provgroningen @112groningennl (EN: Knijpbrug in #Hoogezand is not feeling it anymore, traffic is deadlocked on both sides @provgroningen @112groningennl)*

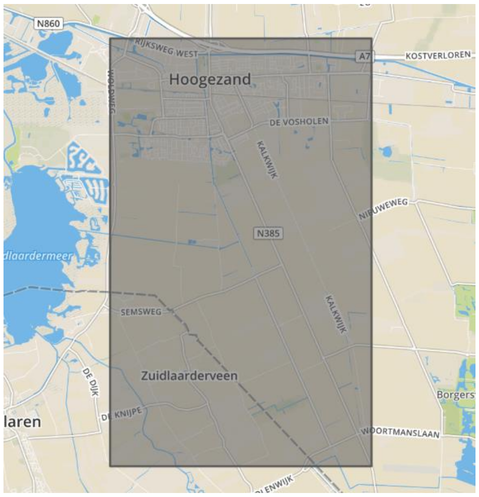


Figure 3-12: Geocoding evaluation category 3

Category 4: The geocoded result covers no places indicated in the tweet, as shown in Figure 3-13.
Example: @meldkamervid het is weer raak op #a10 thv s109 2x ✖✖ (EN: @meldkamervid it is a hit again on #a10 near s109 2x ✖✖)

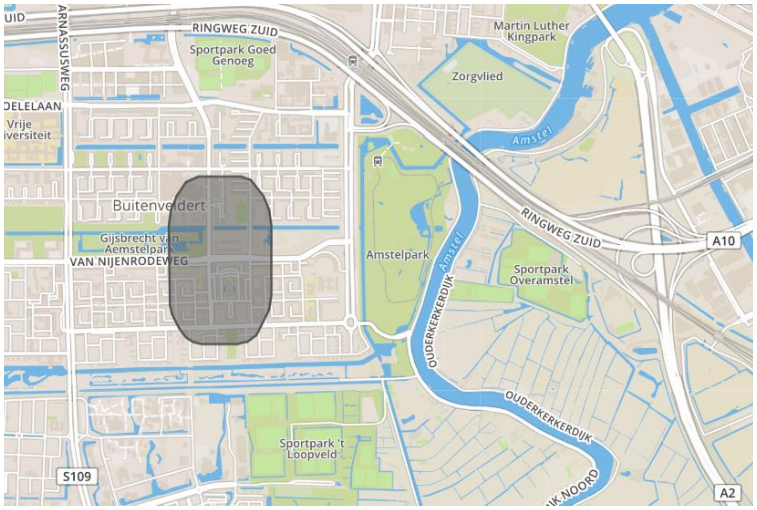


Figure 3-13: Geocoding evaluation category 4

3.7 Traffic Event Description

In this section, the traffic event description step is discussed, in which a rule-based approach is used to cluster related information from TE tweets, Waze events, TomTom events and DiTTLab traffic data. Hereafter, TE tweets, Waze events, and TomTom events will be referred to as a *traffic event reports.* This clustering step will eventually result in the detection of traffic events. The event category, location and time period are used to conduct the clustering. The rule-based clustering approach works as follows. First, a traffic event described by a newly incoming traffic event report is compared to the previously reported traffic events. This comparison can have one of two outcomes:

- **Match to existing traffic event cluster:** if the newly incoming traffic event report lies within the categorical, locational and temporal extent, then the traffic event report is added to the existing traffic event cluster.
- **No match to existing traffic event cluster:** if the newly incoming traffic event report does not lie within the categorical, locational and temporal extent, then a new traffic event cluster is created. This traffic event cluster contains the categorical, locational and temporal properties of the newly incoming traffic event.

Matching is based on a rule-based approach, in which a rule specifies the categorical, spatial and temporal extent, used to assert if the new traffic event report should be part of an existing traffic event cluster. A rule can thus be described as a triplet of the form:

(traffic event category, radius/dilation, timespan)

In which the traffic event category is one of the 13 event categories described in Section 3.3.1. Note that we used TomTom and Waze to co-create these 13 event categories for the tweet annotator. This means that these categories can be used to map TomTom events, Waze events, and TE tweets to traffic events clusters. We use a radius or dilation drawn around either geopoints or geolines/geoshapes that represent the spatial location of a traffic event. This is done because of the possible delay existing between the traffic event location and the location of the creation of the traffic event report. A timespan calculated in minutes from the creation time of the traffic event report, is used to represent the temporal extent of a traffic event. We use a timespan, because the time extracted from a traffic event description does not necessarily represent the exact time a traffic event took place.

Let us take a look at the following example rule: (Event Enforcement, 250m, 30min).
This rule asserts that for a new traffic event report to match to this existing cluster, it must have a category that can be matched to the "Event Enforcement" category. For example, a Waze event with the category "Police" can be matched to the "Event Enforcement" category as police is a type of enforcement. This rule further asserts, that a new traffic event report must be within a range of 250 meters, and within a time interval of 30 minutes.

When a traffic event report successfully matches an existing traffic event cluster, the spatial and temporal information have to be merged. In this case, traffic event reports would only have a location in the form of a geopoint, they could be merged to a weighted average location. A 1:Nratio, where $N$is the number of associated reports to a traffic event cluster, could then be applied to the weights between new traffic event reports and the existing traffic event

cluster. By applying such a ratio, it is ensured that all traffic event reports have the same impact on the average traffic event location.
However, in our case locations can also take the form of geolines, and geoshapes. Therefore, a different spatial merging approach has to be taken. We will elaborate the model, with the help of the following traffic event-related tweet, and simplified Waze, and TomTom examples listed below:

- TE Tweet Report 1:
  - *Text: "@vid ongeluk op de #A4 thv McDonald's #Delft. Vermijd A4 richting Den Haag #File" (EN: @vid accident on the #A4 near McDonald's #Delft. Avoid A4 in the direction of Den Haag #Trafficjam)*
  - *Category: Event Accident, Event Traffic Jam*
  - *Location: geoshape*
  - *Event date: 2018-03-13T10:00:00Z*
- Waze Event Report:
  - *Category: ACCIDENT*
  - *Subcategory: ACCIDENT_MINOR*
  - *Location: geopoint (dilated with a radius of 100 meters)*
  - *Creation date: 2018-03-13T10:05:00Z*
- TomTom Event Report:
  - *Category: Jam*
  - *Location: geoline (dilated with a radius of 100 meters)*
  - *Creation date: 2018-03-13T10:07:00Z*
- TE Tweet Report 2:
  - *Text: "Zojuist kop-staartbotsing gezien bij Delft, viel gelukkig mee." (EN: Just saw a rear-end collision near Delft, could have been worse.)*
  - *Category: Event Accident*
  - *Location: geoshape*
  - *Event date: 2018-03-13T10:08:00Z*

Figure 3-14 depicts the approximate visualization of the locations of these traffic event report examples.

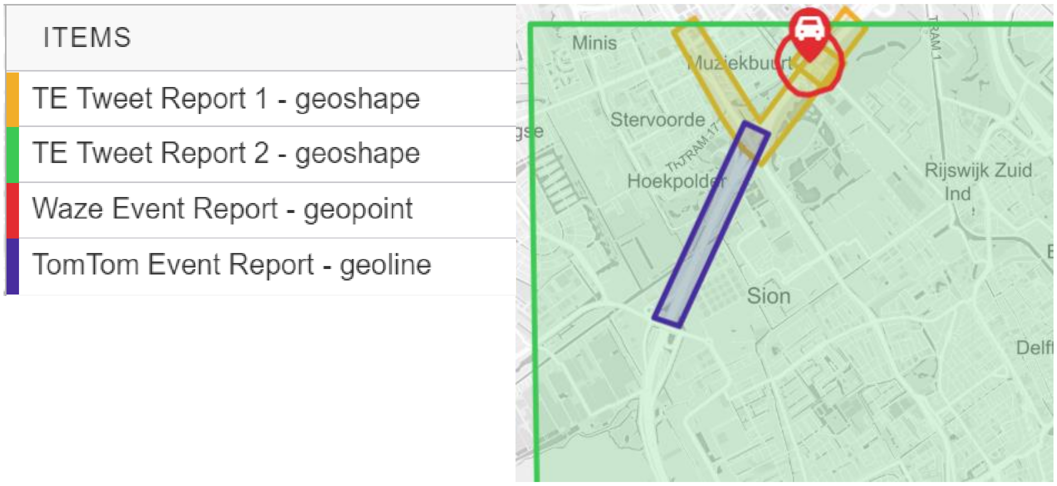

ITEMS
TE Tweet Report 1 - geoshape
TE Tweet Report 2 - geoshape
Waze Event Report - geopoint
TomTom Event Report - geoline
Minis
Muziekbuurt
Stervoorde
TRAM 17
Hoekpolder
TRAM 1
Rijswijk Zuid Ind
Sion
Delft

Figure 3-14: Traffic Event Report Location Examples

The first incoming report is a tweet that cannot be linked to an existing traffic event cluster. A new cluster, called the *MainCluster*, is therefore created with two categories and a geoshape location. When a report contains multiple categories, it also has multiple rules defining the traffic event category, radius/dilation, and timespan. The second incoming report is a Waze event, which category matches with the “Event Accident” category of the *MainCluster*. Its location also intersects with the geoshape location of the *MainCluster*. Therefore, one could say that the Waze event should be added to the cluster. Notice, however, that the Waze event does not match the “Event Traffic Jam” category. Adding the Waze event directly to the *MainCluster* would thus lead to a mismatch of category. Therefore, a subcluster is created based on the matching category. This leaves us with two clusters, one for the initial tweet report (*MainCluster*) and one for the tweet/Waze event report combination (*SubCluster1*). The third incoming report is a TomTom event, which category matches the “Event Traffic Jam” category and its geoline location intersects with geoshape location of the *MainCluster*. This results in a second subcluster (*SubCluster2*). The last incoming report is again a tweet, which has a “Event Accident” category matches with the category from *SubCluster1*. As its location also intersects with the location from *SubCluster1*, the report can be clustered to *SubCluster1*.

Some traffic event reports however, are more relevant than others. We make the following assumption: the larger a traffic event report location is, the more traffic events it can relate to. For example, a traffic event report with Delft as the only location can relate to any traffic event within Delft. Therefore, we compute the relevance of a traffic event report as follows: we divide the area of the intersection of the locations from all traffic event reports by the area of a traffic event report location. For example, in *SubCluster1* the area of the location of *TE Tweet1 Report =* $0.8km^2$, *Waze Event Report =* $0.2km^2$, and *TE Tweet2 Report =* $24km^2$. The area of $((TE\ Tweet1\ Report) \cap (Waze\ Event\ Report)) \cap (Tweet2\ Report)\ =\ 0.2km^2$. Thus, the *TE Tweet1 Report* is most likely to contribute most towards this subcluster (1.00), followed by the *Waze Event Report* (0.25), and the *TE Tweet2 Report* is most likely to be the least relevant to this cluster (0.01).

An example of what the final clustering would look like:

MainCluster:

- *Based on traffic reports: TE Tweet1 Report*
- *Category: Event Accident, Event Traffic Jam*
- *Locations: geoshape*

- *SubCluster1:*
  - *Based on traffic reports: TE Tweet1 Report, Waze Event Report, TE Tweet2 Report*
  - *Category: Event Accident*
  - *Locations:* $((TE\ Tweet1\ Report) \cap (Waze\ Event\ Report)) \cap (TE\ Tweet2\ Report)$
- *SubCluster2:*
  - *Based on traffic reports: TE Tweet Report, TomTom Event Report*
  - *Category: Event Traffic Jam*
  - *Locations:* $(TE\ Tweet1\ Report) \cap (TomTom\ Event\ Report)$

Contrary to the spatial merging approach, we do not apply a temporal merging approach. This because, the datetime of the first event report in a cluster, indicates the closest possible time to the real datetime of a traffic event. Any following matching traffic event reports to this cluster can add to the spatial information and descriptiveness, but cannot improve the datetime of the traffic event. A high-level overview of the traffic event reports clustering approach is given in Figure 3-15.

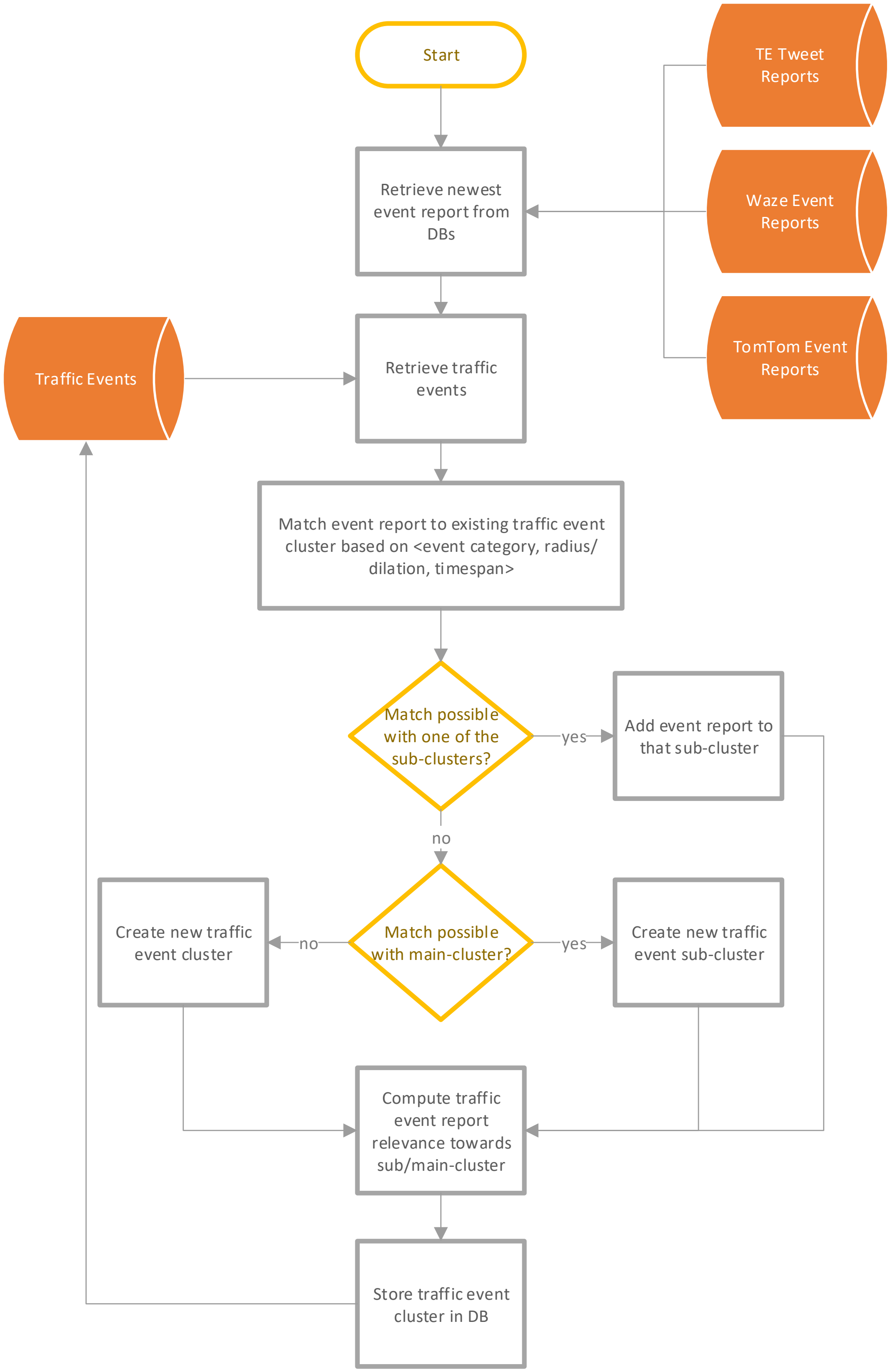

Start
TE Tweet Reports
Waze Event Reports
TomTom Event Reports
Retrieve newest event report from DBs
Traffic Events
Retrieve traffic events
Match event report to existing traffic event cluster based on <event category, radius/ dilation, timespan>
Match possible with one of the sub-clusters?
yes
Add event report to that sub-cluster
no
Create new traffic event cluster
no
Match possible with main-cluster?
yes
Create new traffic event sub-cluster
Compute traffic event report relevance towards sub/main-cluster
Store traffic event cluster in DB

Figure 3-15: High-level overview of traffic event report clustering approach

The next step is to link DiTTLab traffic data, to related traffic events. The DiTTLab traffic dataset consists of the speed/flow values per 100 meter road segment for all motorways in the Netherlands. For DiTTLab data to be related to a traffic event, its motorway geolines have to intersect the location of the traffic event. If this is the case, the traffic data for the intersected segment plus two 100 meter road segments before and after the segment, are linked to the event. We add these additional segments, as a traffic event not only influences the location of the event, but possible also road segments before and after it. For example, a traffic event of the category "Event Accident" could cause an increased congestion level before it due to a traffic jam. Additionally, we extend the time interval as defined in the traffic event cluster rule (traffic event category, radius/dilation, timespan), with an additional 15 minutes before the event start. This way we do not miss any possible increased congestion levels that could have been indicators for the traffic event to happen. For example, a traffic event of the category "Event Accident" that was caused by road debris. That same road debris could thus have caused traffic to slow down (increase congestion) in the 15 minutes leading up to the traffic event. We want to state that we have chosen to only link DiTTLab traffic data to closed traffic events clusters. Traffic event clusters are closed after the time interval from its rule has passed. We chose to take this approach as the DiTTLab traffic data is currently not available in real-time. A high-level overview of the DiTTLab traffic data to traffic events linking approach can be found in Figure 3-16.

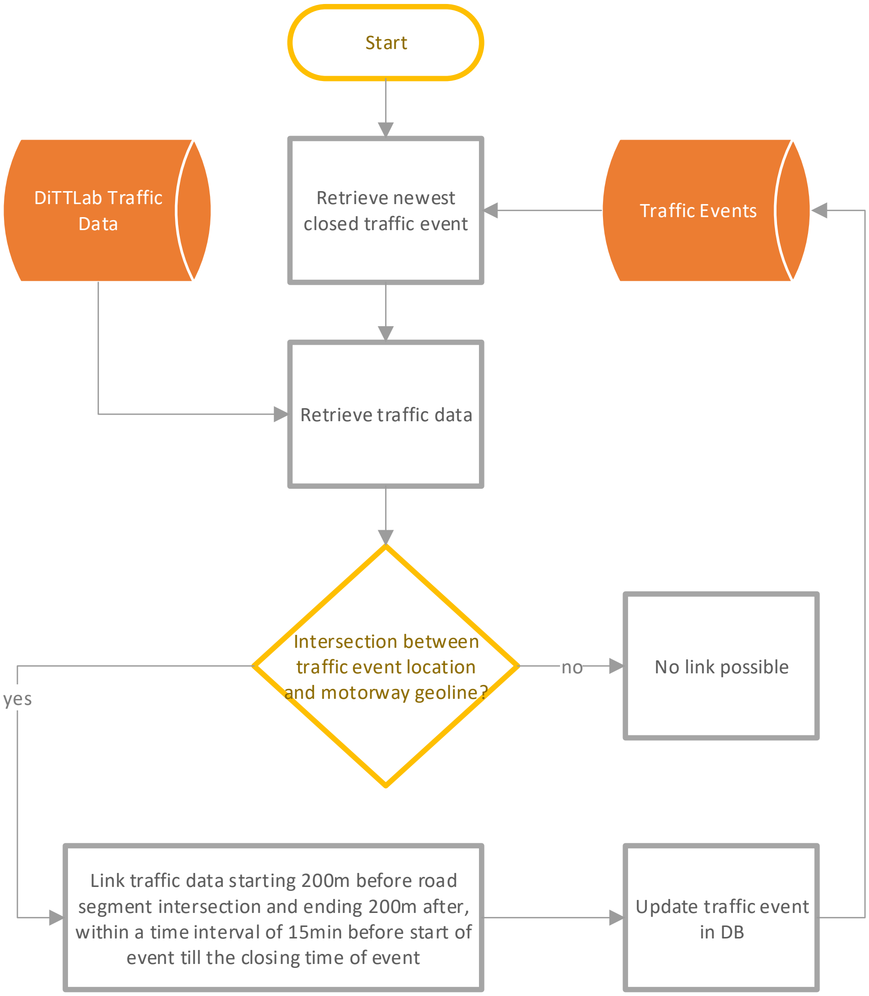

Start
DiTTLab Traffic Data
Retrieve newest closed traffic event
Traffic Events
Retrieve traffic data
Intersection between traffic event location and motorway geoline?
no
No link possible
yes
Link traffic data starting 200m before road segment intersection and ending 200m after, within a time interval of 15min before start of event till the closing time of event
Update traffic event in DB

Figure 3-16: High-level overview of DiTTLab traffic data to traffic events linking approach

3.8 System Architecture - SocialTerraffic

In this section, we discuss the architecture behind our software system, named *SocialTerraffic*[24]. We first discuss how the back-end, formed by our developed pipeline containing the detection, categorization, and description of traffic events, is to be translated to an entity-relationship model. Second, we state the requirements our system has to comply with. Last, we discuss how the data is presented in the front-end layer.

3.8.1 Entity-Relationship Model

By creating an ER model we are able to provide a high-level description of the interrelated things of interest within the traffic event domain of knowledge. This logical data model can be used to form the database behind the SocialTerraffic system. We use the Crow's Foot notation to create relationships between the entities, as illustrated in Figure 3-17. We will explain each entity from the perspective of a traffic event. A traffic event can consist of zero, one or multiple *TE_TWEET*, *TOMTOM_EVENT*, and *WAZE_EVENT* reports (note that it must have at least one of these reports to exist). It must have one or multiple event categories, where an *EVENT_CATEGORY* is a collection of multiple event categories, e.g., event hazard or event traffic jam. Each *EVENT_CATEGORY* has a *RULE*, describing the constraints for the traffic event description approach. A *TRAFFIC_EVENT* must have one or multiple Locations and can be linked to zero, one or multiple entities of *DITTLAB_DATA*. A *TE_TWEET* must be created by a *USER*, while a *USER* can create zero, one or multiple TE tweets. It always contains one or multiple text tokens, where a token is always linked to one *TOKEN_CATEGORY*. A *TOKEN_CATEGORY* is a collection of multiple categories, e.g., *PLACE_CATEGORY*, *TEMPORAL_CATEGORY*, or *EVENT_CATEGORY*. A *TE_TWEET* can have zero or one *COORDINATES*. A *WAZE_EVENT* also must be created by a *USER* in the same way as a *TE_TWEET*. It must contain one *EVENT_CATEGORY* and one *LOCATION*. A *TOMTOM_EVENT* is not created by a user, and must contain one *LOCATION*. A *LOCATION* always contains one or multiple *COORDINATES*, being able to form geopoints, -lines, and –shapes. It can contain one specific *ADDRESS*, where an *ADDRESS* does not have to be a unique location and can also just be a country. At an *ADDRESS* there can be zero, one or multiple *ROAD* entities.

[24] *SocialTerraffic* is a composition of the words *social*, *traffic*, and *terrific*.

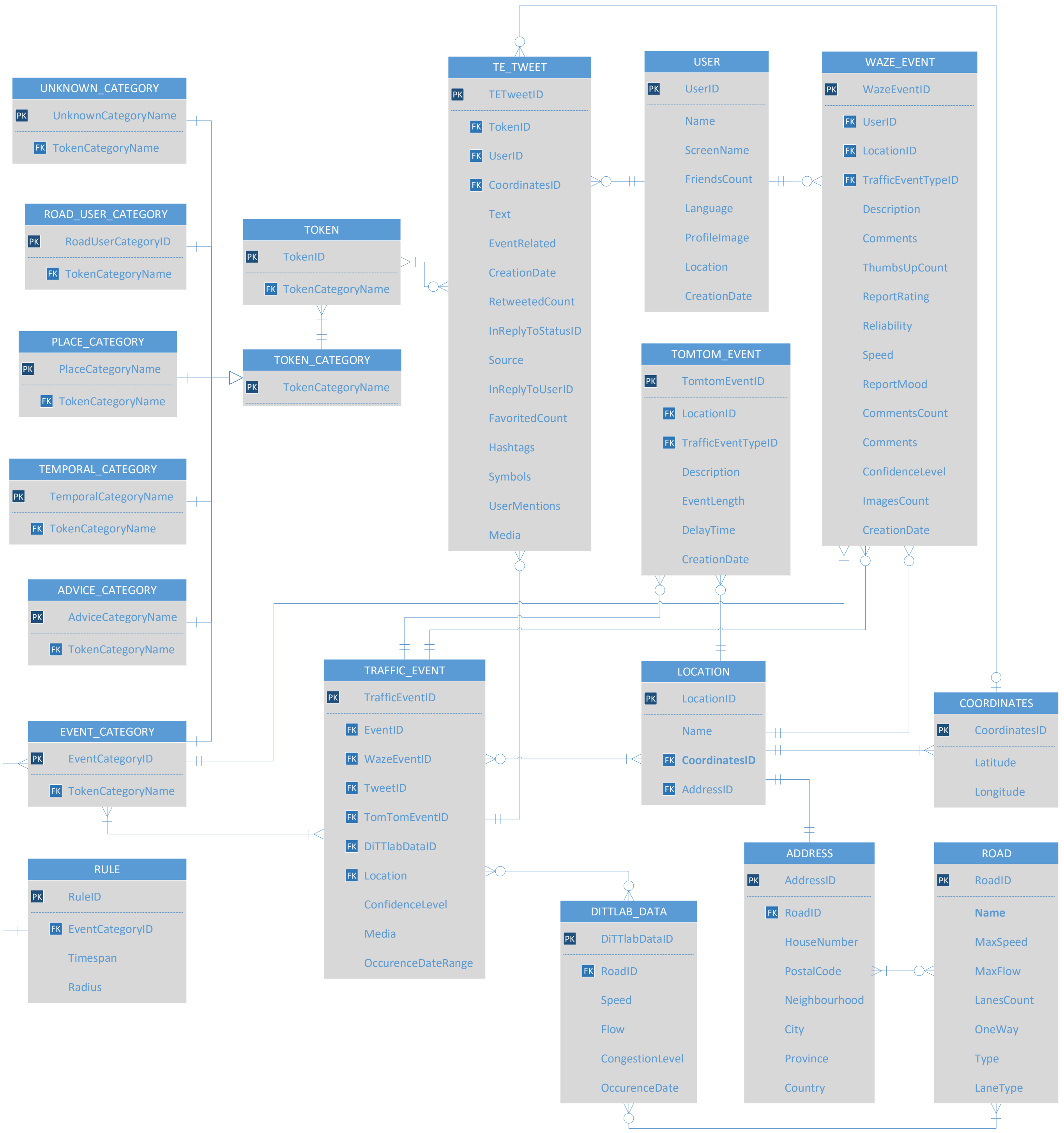


Figure 3-17: ERD Traffic Event Domain Knowledge

3.8.2 Requirements

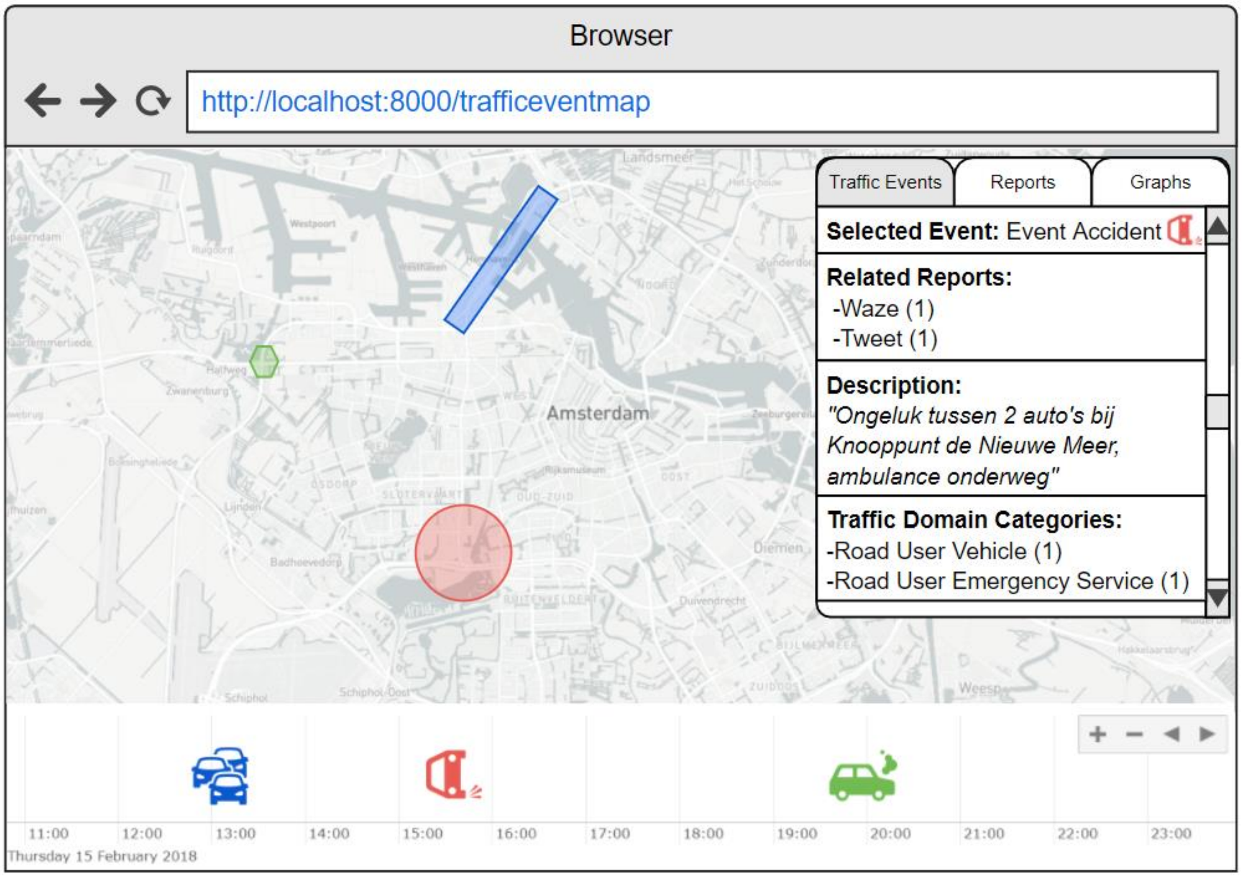
A requirement list, prioritized with the MoSCoW method, has been composed based on meetings with the stakeholders within the Web Information Systems research group and the DiTTLab, as shown in Table 3-8.

Nr.	Requirement	MoSCoW
1.	A user must be able to view the locations of traffic events on an interactive map.	Must
2.	A user must get an overview of all traffic domain categories and their count, and a description for a specific traffic event.	Must
3.	A user must be able to filter traffic events based on event category, time range and location.	Must
4.	A user must be able to view the traffic event reports that are linked to a traffic event.	Must
5.	A user must be able to view DiTTLab traffic data that is linked to a traffic event.	Must
6.	A user should be able to view auto generated graphs by selecting a traffic domain category and timespan for a specific location.	Should

Table 3-8: Requirement List

3.8.3 Data Presentation

To present the collected data to the end user a web-based interactive map application is build. Figure 3-18, depicts a wireframe of the front-end, created to serve as a visual guide representing the skeletal framework of the application. Traffic events, based on clusters from traffic event reports, are displayed on the map by drawing their locations. Different colors are used to represent each of the 13 traffic event categories and their locations on the map. In this wireframe an Event Accident is associated with the color red, an Event Traffic Jam with blue, and Event Hazard Stopped Vehicle with green. On the bottom row of the application, an interactive timeline is situated. This can be used to choose a specific time period to focus on, while the map automatically updates itself based on the new time range. Traffic events are each displayed on the time range with a custom icon/color combination. A sidebar with three different tabs enables a user to view information on events, the reports the events are based upon, and view auto-generated graphs. This wireframe provides an example of how traffic event information could be shown when a user clicks on the red Event Accident location.



Selected Event: Event Accident

Related Reports:
-Waze (1)
-Tweet (1)

Description:
"Ongeluk tussen 2 auto's bij Knooppunt de Nieuwe Meer, ambulance onderweg"

Traffic Domain Categories:
-Road User Vehicle (1)
-Road User Emergency Service (1)

Figure 3-18: Wireframe for the front-end of the SocialTerraffic system

4 Implementation

In this chapter, we discuss the technical implementation of every module discussed in the previous chapter. First, we discuss the data collection approach for tweet, Waze and TomTom event reports. Second, we discuss the evaluation of our custom rule-based traffic domain annotator. Third, we go through the machine learning based traffic event classification module and the achieved results. Fourth, the implementation of our geocoding method is discussed as well as its evaluation. Fifth, we further describe the traffic event description module and discuss the achieved results. Lastly, an overview is provided of the built SocialTerraffic system.

4.1 Data Collection

4.1.1 Collection Timespan Overview

Before discussing each data collection approach for Twitter, Waze and TomTom data, an overview is provided for the different collection timespans. Figure 4-1, provides a visual overview of the collected data sets. A short explanation for each data set is provided below:

- Twitter data was collected over the period from 28-10-2017 to 30-10-2017[25], for the purpose of creating a keyword set creation approach. This keyword set creation approach was then used to collect Twitter data over the period from 05-12-2017 to 17-02-2017. Due to some technical issues, we were not able to collect the data over the period from 04-02-2018 to 05-02-2018. Approximately half of this data set was used to train our machine learning classifier, based on the data over the period from 05-12-2017 to 06-01-2018.
- Waze data was collected over the period from 05-12-2017 to 06-02-2018, yet as the first day and last day of this period were missing parts of the data, we only focus on the period from 06-12-2017 to 05-02-2018.
- TomTom data was collected over the period from 05-12-2017 to 14-02-2018. However, due to some technical issues regarding the TomTom Traffic Incident API, there were some days that we were not able to receive TomTom event reports for the complete day or did not receive any reports at all. This caused us to omit the following days: 05-12-2017, 11-12-2017 to 08-01-2018, 30-01-2018 to 11-02-2018.

We want to state that we would have preferred a more complete Waze and TomTom data set. However, we could not repeat our experiment for all three data sources as Waze unexpectedly changed their policies in February 2018, causing the feed to no longer work.

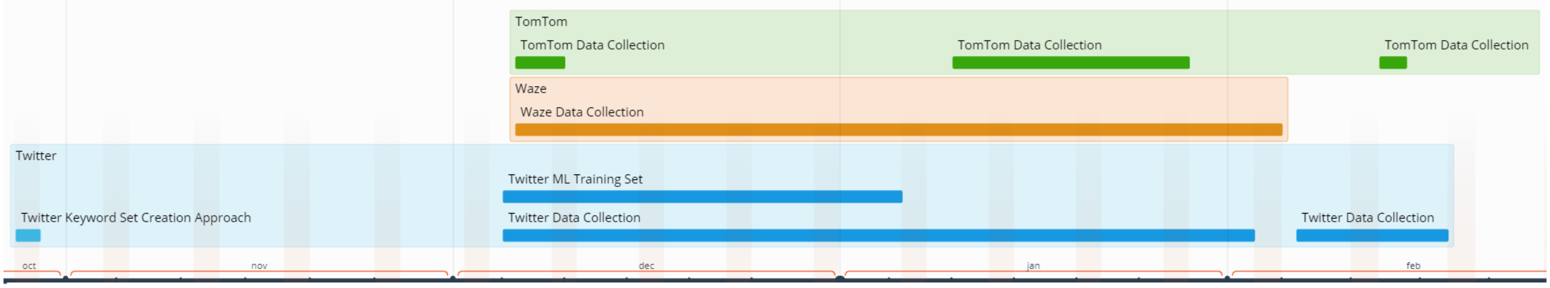


Figure 4-1: Timespan of the collected data sets.

[25] Note that when we mention a date range from date 1 to date 2, it means that date 2 is inclusive.

4.1.2 Twitter Data Collection

The Twitter data collection module was written in the Python3[26] (v. 3.6.1) language. We used the Python library Tweepy[27] (v. 3.5.0) to access the Twitter REST API. We set up an asynchronous crawling approach which allowed us to use multiple Twitter accounts to crawl Twitter without missing out on data due to Twitter limitations (180 calls per 15 minutes per account). Based on the defined keyword set creation approach as described in Section 3.1.1, we implemented all necessary steps to get an optimal Twitter data collection approach. The approach was applied on a three day period from 28-10-17 to 30-10-17. First, a Dutch language, retweet and replies filter was applied. We then used an initial keyword set based on the keywords used in the thesis by Dokter (2015), combined with the road numbers from the Dutch road network, which resulted in a dataset of 16,563 tweets over a three day period. Second, a suspicious term filter and bot filter was added, which brought the set back to 7430 tweets. Third, we applied a URL filter, bringing back the set to 2285 tweets. Last, we manually annotated each tweet in this set as TE or NTE and created by real road user accounts (RRU), as well as TE or NTE but created by non-real road user accounts (NRRU). This way we found 94 TE RRU tweets and 57 TE NRRU tweets.

Next, we identified keywords with their positive and negative correlation towards TE tweets. Table 4-1 shows the top 20 positive tokens and their bigrams based on the first iteration, while, Table 4-2 shows the top 20 negative tokens. Note that tokens from the road number list (e.g., A10, N56, s5) have been replaced with the token "ROADNAME". These results immediately show the difficulty of automating the positive keyword selection process, as the ambiguity of words and the appearance of them outside of the traffic domain plays a major factor. Take, for example, the token "file" (EN: traffic jam), which can relate to a traffic jam but also to the English word "file". This gets even worse when trying to capture negative keywords, as even though they do not appear in any TE tweet within this data collection timeframe that does not mean they will never appear in TE tweets. Filtering out tweets based on negative keywords is therefore too rigorous and not integrated. Take for example the keyword "spits" (rush hour), which is obviously traffic related but did not appear in any TE tweet within this collection time range.

[26] https://www.python.org/

[27] http://www.tweepy.org/

Token	# TE Created by RRU	# NTE Created by RRU	# TE Created by NRRU	# NTE Created by NRRU	Relation
#ROADNAME	44	25	4	65	67.69
ROADNAME	35	258	39	254	13.78
@VID	13	10	1	22	59.09
FILE	12	158	2	168	7.14
ONGEVAL	10	14	7	17	58.82
RICHTING	9	21	2	28	32.14
#FILE	8	16	0	24	33.33
RIJSTROOK	7	3	0	10	70.00
RWS_VERKEER	6	4	0	10	60.00
@ RWS_VERKEER	6	4	0	10	60.00
WEER	6	105	3	108	5.56
AANRIJDING	6	43	5	44	13.64
WEG	6	52	1	57	10.53
@RIJKSWATERSTAAT	6	11	0	17	35.29
LETSEL	5	9	6	8	62.50
@MELDKAMERVID	4	2	0	6	66.67
AFGESLOTEN	4	5	2	7	57.14
SNELWEG	3	7	0	10	30.00
THV	3	2	1	4	75.00

Table 4-1: Top 20 positive tokens in iteration 1

Token	# TE Related by RRU	# NTE Related by RRU	# TE Related by NRRU	# NTE Related by NRRU
SPITS	0	270	0	270
BRUG	0	199	0	199
'	0	79	0	79
ECHT	0	59	0	59
10 KM	0	56	0	56
😂	0	46	0	46
5 KM	0	44	1	43
KM H	0	31	9	22
JAAR	0	30	0	30
ZIEN	0	29	0	29
DAG	0	29	0	29
LT	0	29	0	29
8	0	28	3	25
TREIN	0	27	0	27
WIND	0	27	8	19
50	0	26	0	26
GING	0	25	0	25
MOOIE	0	25	0	25
VIND	0	25	0	25
BETER	0	25	0	25

Table 4-2: Top 20 negative tokens in iteration 1

Besides single keywords, we tried to capture keywords that co-occur within the same tweet to increase the collection of TE tweets. Figure 4-2, shows the top 22 tokens based on their co-occurrence within TE tweets. This shows for example, how traffic tokens often occur together with road numbers.

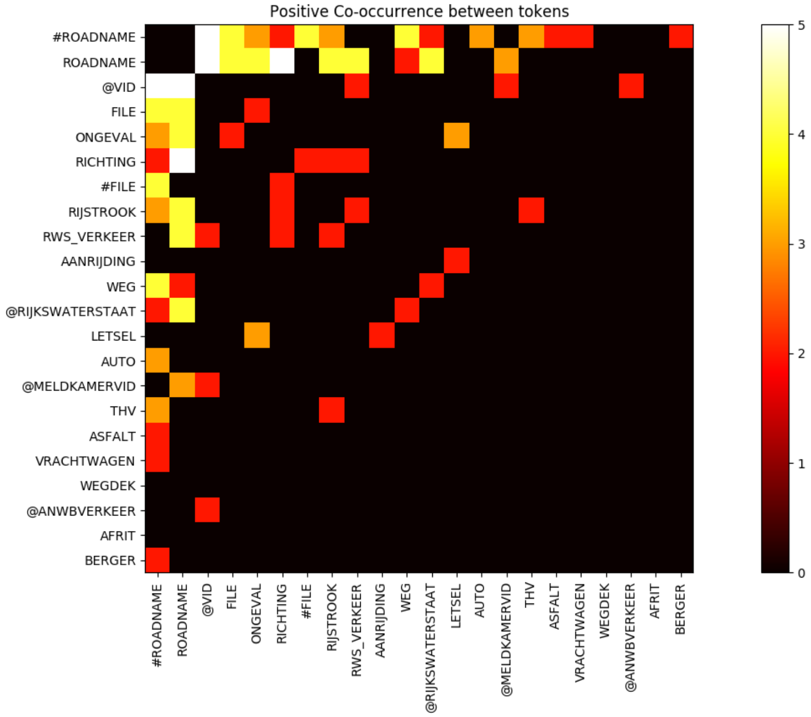


Figure 4-2: Positive co-occurrence between tokens in iteration 1

Based on the gained results we updated our positive keyword list and NRRU account list. Eventually, the third and last iteration provided us with a set of 1861 tweets, containing 138 TE RRU tweets and 24 TE NRRU tweets. A fourth iteration was performed but did not provide any new positive keywords to improve the collection approach. When evaluating this approach we found that our initial iteration contained 2285 tweets from which 4.11% TE RRU and 2.49% TE NRRU. The last iteration contained 1861 tweets from which 7.42% TE RRU and 1.29% TE NRRU. By applying this approach we significantly increased the collection of TE tweets with 80.54%, while reducing NRRU TE tweets with 48.19%. Based on this iterative approach the following lists have been formed, and were used in our final Twitter data collection approach:

Positive keyword set (in combination with a road numbers list):

['VID', 'RIJSTROOK', 'FILE', '@RIJKSWATERSTAAT', 'ONGEVAL', 'THV', '@RWS_VERKEER', '@MELDKAMERVID', 'ONGELUK', 'VRACHTWAGEN', 'AFRIT', 'VERKEER', 'ASFALT','LETSEL', 'WEGDEK', 'PECHGEVAL', 'AANRIJDING', 'VLUCHTSTROOK', 'PECH', 'BERGER', 'SPITSSTROOK','RWS', 'RIJSTROKEN', 'AFSLAG', 'BERM', '@ANWBVERKEER', 'TANKSTATION', 'SNELHEID', 'TUNNEL', 'KRUISING, AANRIJDING, AUTO, AUTO'S]

Suspicious terms set, used to filter non-real road user accounts:

['FILE', 'VERKEER', 'NEWS', 'NWS', 'NIEUWS', 'WEER', '112', 'HEADLINE', 'P2000', 'NL', 'FLITS', 'P2K', 'TV', 'RADIO', 'PROVINCIE', 'OMROEP', 'DAGBLAD', 'WEEKBLAD', 'ACTUEEL', 'GEMEENTE', 'MEDIA', 'HOLLAND', 'NOORD', 'ZUID', 'OOST', 'WEST', 'VANDAAG', 'STUDIO', 'AUTO', 'METEO', 'BRUG', 'ALARM', 'BRAND', 'AMBULANCE', 'BRANDWEER', 'VID', 'MELDKAMERVID', 'POLITIE', 'BOT', 'ANWB', 'HV', 'WAZE','TRAFFIC', 'ALERT', 'BRW', 'COP', 'SPOTTER', 'P2', 'NU', 'REDACTIE', 'DAGBLAD', 'PD', 'MEDIA', 'FM','STANDAARD', 'POLITIE', 'TRAUMA', 'HELI', 'ACTUEEL', 'INFO', 'STUDIO', 'REGIO', 'GEMEENTE', 'STAD', 'COURANT', 'PERS', 'OMROEP', 'VANDAAG', 'KRANT', 'ACTUEEL', 'ALARM', 'BRUG', 'CAR', 'AUTO']

Non-real road user accounts list, consisting of 454 account names (only 20 examples are shown).

["Verkeerscentrum", "NMBS", "VGSpijkenisse", "ANWBeuropa", "CalabotsUtrecht", "hvalmere", "ANWBeuropa", "WazeTrafficGENT", "LL3", "lingewaalalert", "BrwKrabbendijke", "KristalITdotcom", "Tom_zulu10", "brug_open", "_Veluwe", "zwolle", "RijswijksBelang", "middelburg", "cop_spotter", "_kampen", "hvzeeland"]

4.1.2.1 Twitter Data Collection Results

With this traffic data collection approach, we collected Twitter data over the period from 05-12-2017 to 17-02-2018, as visualized in Figure 4-3. Based on this visualization we get the impression that there is an even daily collection of tweets, with two extreme outliers at 11-12-2017 and 18-01-2018. Both outliers are most likely caused by of extreme weather conditions during these dates. On 11-12-2017 the Royal Netherlands Meteorological Institute (KNMI) issued a code red for heavy snowfall[28]. On 18-01-2018 the KNMI issued a code red for a heavy storm (in top 10 storms within last 50 years)[29]. Due to some technical issues, we were not able to collect the data for 04-02-2018 and 05-02-2018. The quantitative results can be found in Table 4-3.

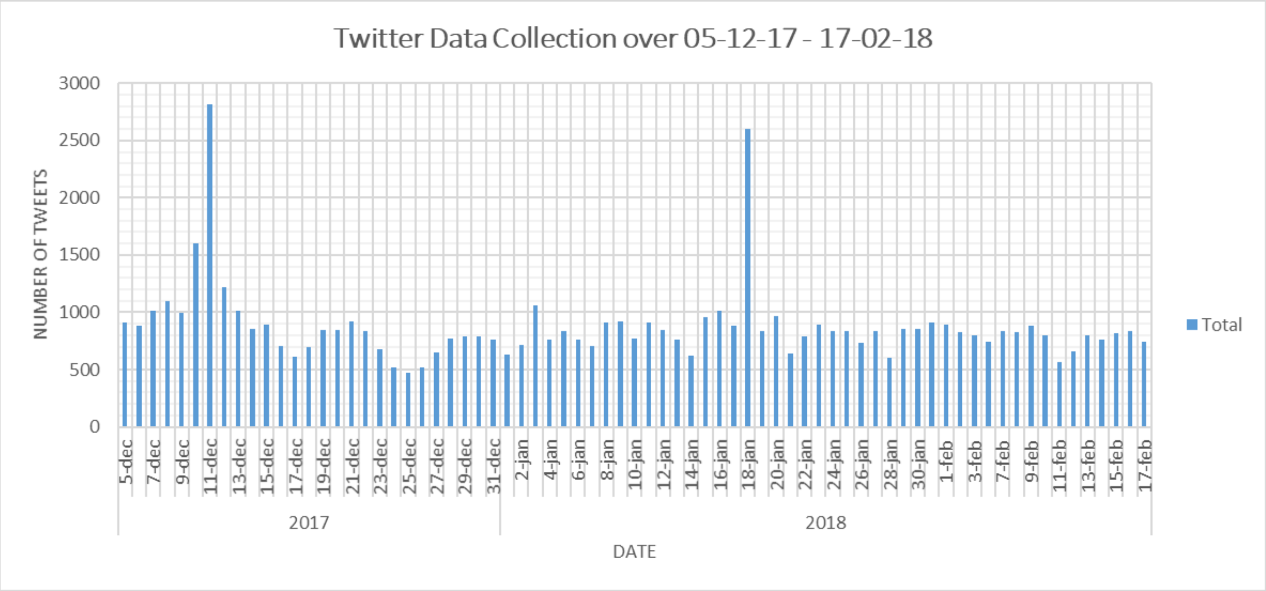


Figure 4-3: Tweets collected with the Twitter data collection

Twitter Data Collection over 05-12-2017 – 17-02-2018	
Metric	**Total # of tweets**
Mean per day	873
Median per day	837
Std. Dev. Per day	349
Min. per day	470
Max. per day	2817
Total	63,727

Table 4-3: Twitter Data Collection Metrics 05-12-2017 to 17-02-2018

[28] https://www.knmi.nl/kennis-en-datacentrum/achtergrond/code-rood-voor-zware-sneeuw-op-11-december-2017

[29] https://www.knmi.nl/kennis-en-datacentrum/achtergrond/code-rood-voor-zeer-zware-windstoten-op-18-januari-2018

We manually labeled the tweets collected between 05-12-2017 and 06-01-2018 as traffic event-related or not traffic event-related, as depicted in Figure 4-4. The quantitative results belonging to this set can be found in Table 4-4. This data clearly shows how small the percentage of TE tweets is that we have to work with, as TE tweets on average only account for 6.71% of all the collected tweets per day.

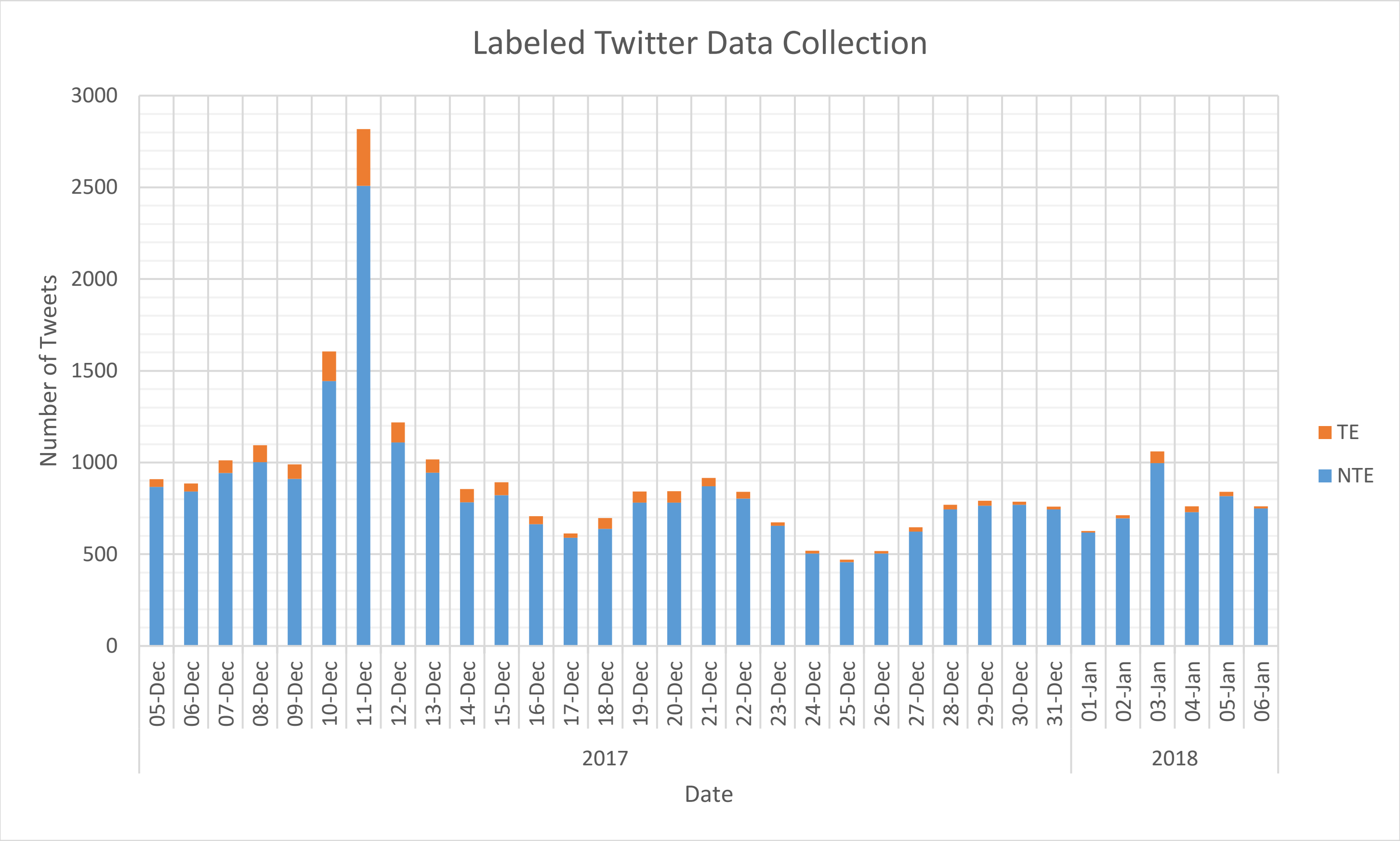


Figure 4-4: Tweets collected with the Twitter data collection approach, labeled traffic event-related (TE) or non-traffic event-related (NTE)

Twitter Data Collection over 05-12-2017 – 06-01-2018			
Metric	**# of NTE related tweets**	**# of TE related tweets**	**Total # of tweets**
Mean per day	839	54	893
Median per day	782	41	840
Std. Dev. Per day	349	56	403
Min. per day	457	8	470
Max. per day	2509	308	2817
Total	27,683	1769	29,452

Table 4-4: Twitter Data Collection Metrics 05-12-2017 to 06-01-2018

4.1.3 Waze Data Collection

The Waze data collection module was written in the Java[30] (v.1.8.0) language. We set up an asynchronous crawling approach which allowed us to send multiple calls to the Waze server (web-based live map) at the same time. A call consists of the following structure:

```
https://www.waze.com/row-rtserver/web/TGeoRSS?left=" + bbox[0] + "&right=" + bbox[2] +
"&bottom=" + bbox[1] + "&top=" + bbox[3]
```

Here the left, right, bottom, and top values represent the four corners of the provided bounding box.

By specifying a geo bounding box all Waze live map data within that region can be extracted, up to a limit of 200 “alerts” (traffic events) and 100 “jams” (extension of specific types of traffic events). We start our approach by making a call to the Waze server by providing the following bounding box representing the Netherlands:

```
left            = "3.31497114423";
right           = "7.09205325687";
top             = "53.5104033474";
bottom = "50.803721015";
```

Because of this limitation, the initial bounding box covering the Netherlands is automatically split into four smaller bounding boxes until we collect less than 200 alerts and less than 100 jams. This way we ensured that all Waze data in the Netherlands was collected. As this bounding box also intersects Germany and Belgium an additional filter is applied on the data to ensure the “country” field of a Waze report equals the Netherlands. As the Waze Live Map is updated every two minutes, our method downloads the JSON files in two-minute intervals, and stores them in a MongoDB[31] document database.

We collected Waze data over a period from 05-12-2017 to 06-02-2018, resulting in 479,703 unique Waze alerts. Figure 4-5 shows how the Waze alerts are distributed over the period from 06-12-2017 to 05-02-2018 (the data from 05-12 and 06-02 has been left out as it contained only part of the day). We only focus on Waze alerts, as jams do not provide any significant new information. Additionally, we only focus on the first appearance of unique Waze alerts, and neglect any appearances of the same alert thereafter. This because, Waze alerts (from here on out referred to as Waze event reports) can have variable lifespans, e.g., an event report with the category “accident” can remain active for 30 minutes while an event report with the category “road closed” can remain active for multiple days. Furthermore, when looking at the distribution of the Waze event reports in Figure 4-5, it is clearly visible how the number of reports significantly decreases during the weekends (e.g., 13-01/14-01 and 20-01/21-01). Also, a significant decrease in reports is noticeable in the period from 23-12-2017 to 06-02-2018, most likely due to the Christmas break[32]. The quantitative results belonging to this set can be found in Table 4-5.

[30] http://www.oracle.com/technetwork/java/javase/overview/index.html

[31] https://www.mongodb.com/

[32] https://www.rijksoverheid.nl/onderwerpen/schoolvakanties/vraag-en-antwoord/wanneer-zijn-de-schoolvakanties-in-2017-2018

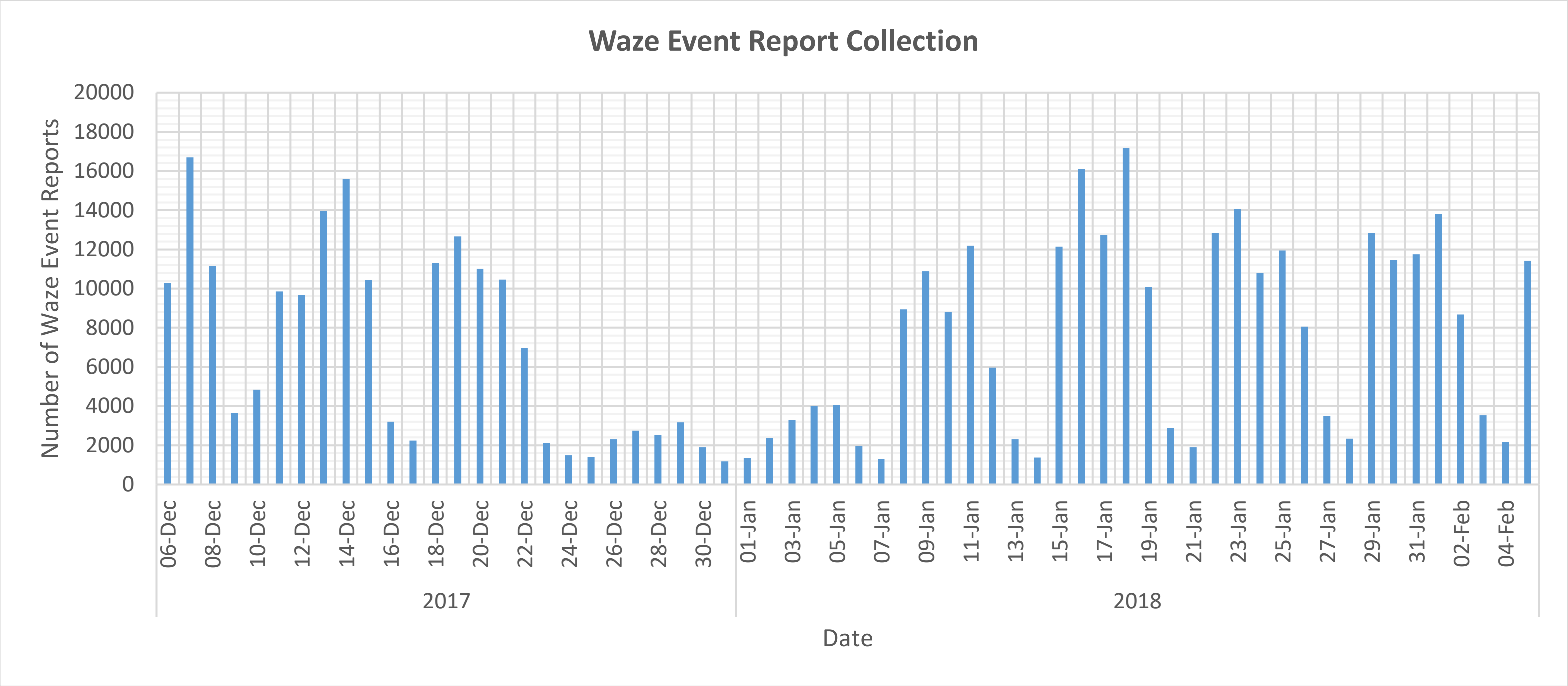


Figure 4-5: Waze Event Report Collection 06-12-17 to 05-02-18

Metric	Number of Waze Event Reports
Mean of Waze event reports per day	7482
Median of Waze event reports per day	8362
Std. Dev. of Waze event reports per day	4905
Min. of Waze event reports per day	1179
Max. of Waze event reports per day	17,189
Total Number of Waze event reports over 62 days	463,891

Table 4-5: Waze Event Report Metrics 06-12-17 to 05-02-18

Even though Waze is a community-based platform based on data from real people, there is nothing that prevents so-called non-real road-users from posting Waze event reports. As our main focus lies on geosocial data from real road-users, we analyzed the top users of our dataset, as shown in Table 4-6. Notice how 29.29% of the Waze event reports are posted by anonymous (N/A) users, followed by the user "Wegstatus.nl", with 12.54%. Even though other users also show high activity rates compared to the mean overall Waze event reports, we could not find any indication that these were non-real road-users. The user "Wegstatus.nl" however matches the website of the same name[33], which is a website that uses multiple data sources (e.g., NDW[34], NBd[35], LiveP2000.nl[36], and Buienradar[37]) to inform users on traffic situations. Therefore, we decided not to include Waze event reports from this user.

[33] https://wegstatus.nl

[34] www.ndw.nu

[35] https://www.bewegwijzeringsdienst.nl/

[36] http://livep2000.nl/

[37] https://www.buienradar.nl/

User	Number of Waze Event Reports	Percentage of Total Number of Waze Event Reports	Metric	Number of Waze Event Reports
N/A	140315	29.29%	Mean per User	13.67
Wegstatus.nl	60089	12.54%	Median per User	3
Rho65536	3363	0.70%	Std dev per User	815.59
RiCo4Cool	1056	0.22%	Min per User	1
marcogpw	795	0.17%	Max per User	140315
martiensch	582	0.12%		
DengKao	562	0.12%		
RunningJohnny	549	0.11%		
choco-nl	515	0.11%		
ArTsLeOpS	428	0.09%		

Table 4-6: Waze User Metrics 06-12-17 to 05-02-18

As Waze users are able to link their Twitter account to their Waze account, we decided to also collect all Dutch Waze related data on Twitter. Automated Waze tweets contain either information on traffic events posted by the user or a summary of the car ride of the user. Data was collected over the same period as the Waze collection. As previously stated in Section 3.1.2, we collect Waze data from Twitter based on the format of Traffic Event tweets:

Traffic Event: ***“Hielp chauffeurs in de omgeving door het melden*** *van wegwerkzaamheden op de N209 - Nieuwe Hoefweg, Bleiswijk* ***via @waze - social navigation.”***

This way we were able to collect 266 tweets from 66 unique Waze users that have their Twitter account linked to their Waze account, over the period from 21-12-2017 to 06-02-2018. We compared the creation date (rounded to seconds) of each of the collected tweets with the creation date of Waze event reports in that same period. This because Waze automatically almost instantly posts a tweet based on the Waze event report created by the user. However, as there can be multiple Waze event reports with the same date, a second comparison is performed based on the equality of the Twitter username, Twitter screen name or street name in the tweet text, with the Waze user or Waze street name. This resulted in a match for 45 tweets from 16 unique Waze users, meaning only 0.016% of the Waze event reports in that period could be linked to a tweet. Which means that we were able to link 0.055% of the Waze users in that period to their Twitter account. A selection of the results is shown in Table 4-7.

Twitter Name	Twitter Screen Name	Twitter Text	Waze Name	Waze Street	Waze Date
Edwin	edwin21	Hielp chauffeurs in de omgeving door het melden van een file op de A12 - E35 via @waze - social navigation. https://t.co/2gAdTug3ej	edwin21	A12 - E35	21-12-17 6:53
Mike van Vessem	mikevanvessem	Hielp chauffeurs in de omgeving door het melden van wegwerkzaamheden op de N209 - Nieuwe Hoefweg, Bleiswijk via @waze - social navigation...	Mike-vv	N209 - Nieuwe Hoefweg	21-12-17 11:48
Arjan Vogelaar	ArjanVogelaar	Hielp chauffeurs in de omgeving door het melden van een stilstaand voertuig op de vluchtstrook op de A4 via @waze - social navigation. ht...	ArjanVogelaar	A4	29-12-17 5:42

Table 4-7: Example of Twitter Accounts linked to Waze Accounts

Now that we have gained an insight into the Waze event report distribution per day, we examine the distribution of Waze event reports and their categories over a 24-hour period. This could give us valuable insights into the relation of Waze alerts and traffic patterns that are bounded to dayparts, as well as the difference in distribution to our other geosocial data sources. This is useful, as it shows how Waze can mitigate the weaknesses of and/or support the reports of other geosocial data sources. Figure 4-6 depicts the average distribution of the main categories of Waze event reports of the entire data set plotted on a 24-hour scale. Notice how the number of Waze event reports reduces in the nighttime hours. The *HAZARD* category is the dominant category over the entire day. Whereas the *JAM* category clearly has its peaks during the rush hour periods. Events with the category *ACCIDENT, POLICE*, and *ROAD_CLOSED*, show a more consistent pattern between 6 am and 23 pm.

Next, we look at how Waze event reports are distributed by category and subcategory per day. Table 4-8 on the next page, shows how Waze event reports are dominated by the *JAM* category (63.42%), and *HAZARD* category (29.44%). Also, note how the *HAZARD_ON_SHOULDER_CAR_STOPPED* category is highly representative in the *HAZARD* category, accounting for 68.27% of the *HAZARD* typed event reports. Another remarkable finding is that the bulk of *ROAD_CLOSED* typed event reports are of the category *ROAD_CLOSED_EVENT*, describing road closures for special events such as sport matches.

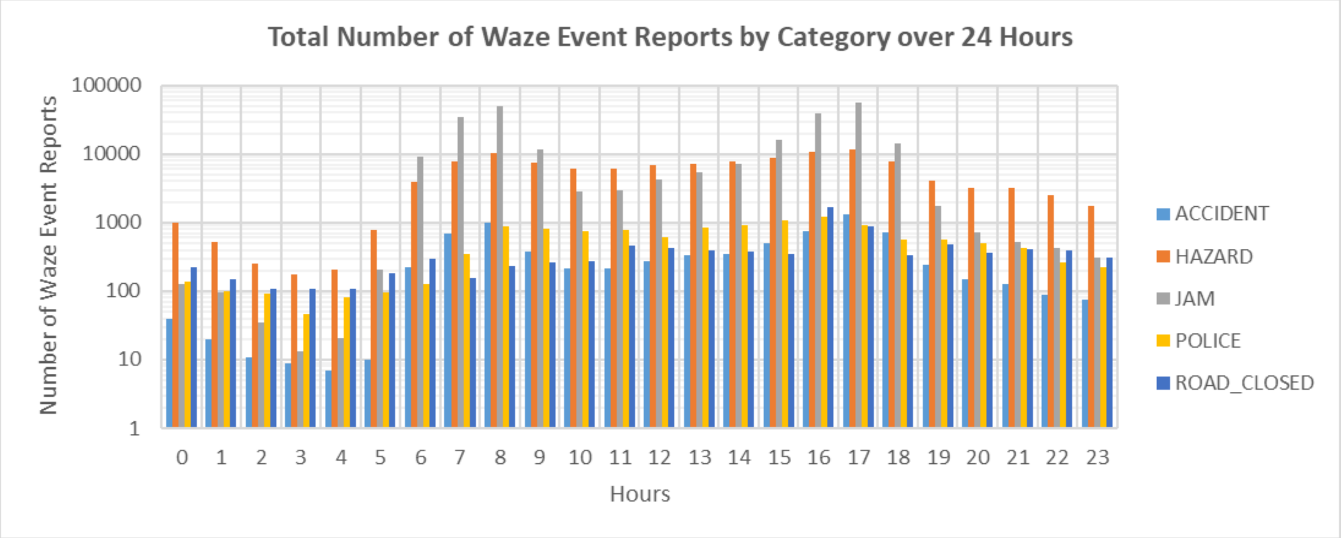


Figure 4-6: Total Number of Waze Event Reports by Category over 24 Hours

Main Category	Subcategory	Mean	Median	Std dev	Min	Max	Total	Percentage
ACCIDENT	ACCIDENT_MAJOR	21.29	19	11.12	3	65	1320	17.11%
	ACCIDENT_MINOR	70.95	75	44.52	9	162	4399	57.03%
	N/A	32.18	31	16.39	8	92	1995	25.86%
ACCIDENT Total		**124.42**	**133**	**69.41**	**28**	**303**	**7714**	**1.90%**
HAZARD	HAZARD_ON_ROAD	3.45	3	2.66	0	12	214	0.18%
	HAZARD_ON_ROAD_CAR_STOPPED	150.26	165.5	72.60	28	309	9316	7.80%
	HAZARD_ON_ROAD_CONSTRUCTION	160.87	171	94.18	30	347	9974	8.35%
	HAZARD_ON_ROAD_ICE	34.03	1.5	121.51	0	727	2110	1.77%
	HAZARD_ON_ROAD_LANE_CLOSED	0.06	0	0.25	0	1	4	0.00%
	HAZARD_ON_ROAD_OBJECT	43.03	35.5	55.68	7	453	2668	2.23%
	HAZARD_ON_ROAD_POT_HOLE	30.11	30	12.67	4	60	1867	1.56%
	HAZARD_ON_ROAD_ROAD_KILL	5.69	5.5	2.68	1	13	353	0.30%
	HAZARD_ON_ROAD_TRAFFIC_LIGHT_FAULT	4.24	3	4.87	0	31	263	0.22%
	HAZARD_ON_SHOULDER	14.32	7	16.90	0	65	888	0.74%
	HAZARD_ON_SHOULDER_ANIMALS	4.18	4	2.35	0	13	259	0.22%
	HAZARD_ON_SHOULDER_CAR_STOPPED	1314.84	1471	480.91	478	2218	81520	68.27%
	HAZARD_ON_SHOULDER_MISSING_SIGN	7.87	7	5.21	1	31	488	0.41%
	HAZARD_WEATHER	2.77	1	4.45	0	24	172	0.14%
	HAZARD_WEATHER_FLOOD	5.31	2	7.59	0	33	329	0.28%
	HAZARD_WEATHER_FOG	89.85	2	276.16	0	1419	5571	4.67%
	HAZARD_WEATHER_HAIL	23.56	1	69.63	0	468	1461	1.22%
	HAZARD_WEATHER_HEAVY_SNOW	0.19	0	0.62	0	3	12	0.01%
	HAZARD_WEATHER_MONSOON	0.02	0	0.13	0	1	1	0.00%
	N/A	31.26	27.5	19.49	8	150	1938	1.62%
HAZARD Total		**1925.94**	**2048**	**819.63**	**655**	**3628**	**119408**	**29.44%**
JAM	JAM_HEAVY_TRAFFIC	1998.69	1809	1795.72	67	6196	123919	48.17%
	JAM_MODERATE_TRAFFIC	1002.98	1021	823.91	29	2554	62185	24.17%
	JAM_STAND_STILL_TRAFFIC	866.76	785.5	777.28	32	2915	53739	20.89%
	N/A	281.06	241	223.81	25	789	17426	6.77%
JAM Total		**4149.50**	**3856.5**	**3602.06**	**185**	**12444**	**257269**	**63.42%**
POLICE	N/A	57.06	58	17.33	20	102	3538	28.61%
	POLICE_HIDING	48.32	50	17.68	16	89	2996	24.22%
	POLICE_VISIBLE	94.10	90.5	33.77	38	161	5834	47.17%
POLICE Total		**199.48**	**199**	**64.51**	**77**	**317**	**12368**	**3.05%**
ROAD_CLOSED	N/A	1.27	1	1.50	0	7	79	0.89%
	ROAD_CLOSED_CONSTRUCTION	4.40	4	3.33	0	14	273	3.07%
	ROAD_CLOSED_EVENT	137.58	110	188.14	13	1458	8530	95.77%
	ROAD_CLOSED_HAZARD	0.40	0	0.79	0	4	25	0.28%
ROAD_CLOSED Total		**143.66**	**117**	**189.28**	**13**	**1464**	**8907**	**2.20%**

Table 4-8: Waze Event Report Distribution by Category per Day

Next, we take a look at what additional information Waze could offer us to help towards the description of traffic events. We found that only 0.06% of all Waze event reports contained one or multiple images. Additionally, only 2.37% of all Waze event reports contained an additional user created description, mostly consisting of concise keywords. Examples of such descriptions are: werkzaamheden (EN: roadworks), renovatie en restauratie (EN: renovation and restauration), water en zand op de weg (EN: water and sand on the road), ongeval (EN: accident), gladheid (EN: slipperiness), zwaan op de middenberm (EN: swan on traffic separator), weg dicht vallende taken (EN: road closed falling branches).

4.1.4 TomTom Data Collection

The TomTom data collection module was written in the Java (v.1.8.0) language. We set up a crawling approach which allowed us to send calls to the TomTom Traffic Incident API every 2 minutes. The TomTom Traffic Incident API is updated every 2 minutes, with the latest information about traffic jams and traffic related incidents. A call is structured as follows:

```
minX                = "3.31497114423";
maxX                = "7.09205325687";
maxY                = "53.5104033474";
minY                = "50.803721015";
baseURL             = "https://api.tomtom.com/traffic/services/";
versionNum          = "4";
style               = "s3";
zoom                = "11";
trafficModelID      = "-1";
format              = "json";
key                 = "?key=zhW9XMcRTCCJuAjfflYGFZwPOWXVsnrs";
language            = "&language=en";
projection          = "&projection=EPSG4326";
geometries          = "&geometries=original";
expandCluster       = "&expandCluster=true";
originalPos         = "&originalPosition=true";

request             = baseURL + versionNum + "/incidentDetails/" + style + "/" + minY + "," + minX
+ "," + maxY + "," + maxX + "/" + zoom + "/" + trafficModelID + "/" + format + key + language + projection
+ geometries + expandCluster + originalPos;
```

As the used bounding box also intersects Germany and Belgium an additional filter is applied to the data. Unlike Waze event reports, the TomTom event reports do not include a "country" tag, therefore the "ID" tag (e.g., "europe_HD_NL_TTL116755785625744", or "europe_HD_BE_TTL116755785625745") is used by applying a filter on the "NL" part. The retrieved reports have been stored in a MongoDB document database.

We collected TomTom data over a period from 05-12-2017 to 14-02-2018, resulting in 90,008 unique TomTom event reports. However, due to some technical issues regarding the TomTom Traffic Incident API, there were some days that we were not able to receive TomTom event reports for the complete day or did not receive any reports at all. We therefore only look at the days where we were able to collect TomTom data for the entirety of the day. This, in order to prevent making an incorrect analysis due to the skewness in our data set. Figure 4-7 shows how the TomTom event reports are distributed over the period from 06-12-2017 to 14-02-

2018. Note, that just as with the Waze event reports, we only focus on the first appearance of unique TomTom event reports, and neglect any appearances of the same report thereafter. When looking at the distribution of the TomTom even reports, it is clearly visible how the number of reports significantly decreases during the weekends (e.g., 13-01/14-01 and 20-01/21-01). Additionally, one clear outlier is visible on the 10th of December, possibly caused due to extremely bad weather conditions on that day[38]. Table 4-9 shows the additional metrics for the TomTom event report collection.

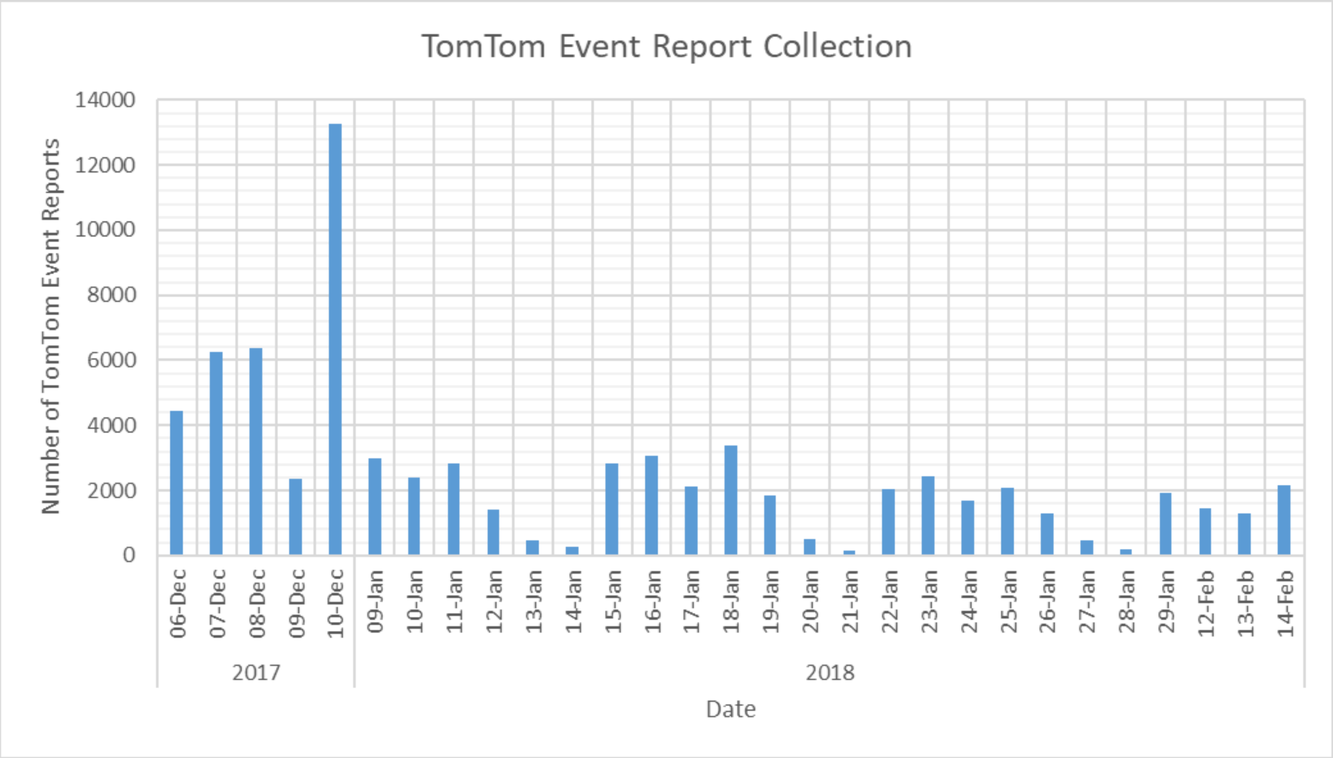


Figure 4-7: TomTom Event Report Collection 06-12-17 to 14-02-18

Metric	Number of TomTom Event Reports
Mean of TomTom Event Reports per day	2544
Median of TomTom Event Reports per day	2071
Std. Dev. of TomTom Event Reports per day	2529
Min. of TomTom Event Reports per day	166
Max. of TomTom Event Reports per day	13,249
Total Number TomTom Event Reports over 29 days	73,764

Table 4-9: TomTom Event Report Collection Metrics 06-12-17 to 14-02-18

[38] http://www.knmi.nl/kennis-en-datacentrum/achtergrond/Code-oranje-voor-zware-sneeuw-op-10-december-2017

Main Category	Subcategory (Description)	Mean	Median	Std. Dev.	Min	Max	Total	Percentage
Accident	accident	17.83	14	11.79	0	43	517	65.03%
	incident	1.97	1	2.79	0	14	57	7.17%
	accident involving heavy lorry	1.34	1	2.22	0	11	39	4.91%
	overturned heavy lorry	0.93	0	4.74	0	26	27	3.40%
	accident. stationary traffic	0.79	0	1.13	0	5	23	2.89%
	incident. stationary traffic	0.38	0	0.67	0	2	11	1.38%
	vehicle fire	0.38	0	0.93	0	4	11	1.38%
	accident. queuing traffic	0.34	0	0.66	0	2	10	1.26%
	accident involving heavy lorry. stationary traffic	0.28	0	0.74	0	3	8	1.01%
	accident. slow traffic	0.24	0	0.57	0	2	7	0.88%
Accident Total		27.41	24	19.96	4	96	795	1.08%
Dangerous Conditions	obstruction on the road	7.66	8	4.30	1	18	222	30.88%
	broken down heavy lorry	4.86	4	4.38	0	16	141	19.61%
	broken down vehicle	4.79	4	3.51	0	15	139	19.33%
	rescue and recovery work	2.59	1	2.95	0	12	75	10.43%
	emergency vehicle	1.14	0	2.16	0	7	33	4.59%
	spillage on the road	0.83	1	0.95	0	3	24	3.34%
	clearance work	0.72	0	1.20	0	4	21	2.92%
	fallen trees	0.45	0	2.19	0	12	13	1.81%
	people on roadway	0.45	0	0.72	0	3	13	1.81%
	animals on the road	0.28	0	0.45	0	1	8	1.11%
Dangerous Conditions Total		24.79	25	13.81	2	53	719	0.97%
Rain	heavy rain	83.69	0	205.11	0	994	2427	99.84%
	heavy rain. obstruction on the road	0.10	0	0.55	0	3	3	0.12%
	emergency vehicle. heavy rain	0.03	0	0.18	0	1	1	0.04%
Rain Total		83.83	0	205.73	0	998	2431	3.30%
Ice	sleet	120.52	0	444.10	0	2374	3495	36.22%
	snow on the road	113.07	0	457.97	0	2501	3279	33.98%
	heavy snowfall	97.93	0	518.20	0	2840	2840	29.43%
	snow on the road. sleet	0.38	0	1.65	0	9	11	0.11%
	snow on the road. heavy rain	0.24	0	1.28	0	7	7	0.07%
	sleet. snow on the road	0.17	0	0.75	0	4	5	0.05%
	heavy snowfall. snow on the road	0.14	0	0.73	0	4	4	0.04%
	snow on the road. heavy snowfall	0.10	0	0.55	0	3	3	0.03%
	heavy rain. snow on the road	0.07	0	0.25	0	1	2	0.02%
	sleet. heavy rain	0.07	0	0.36	0	2	2	0.02%

Ice Total		332.76	0	1414.12	0	7742	9650	13.08%
Jam	stationary traffic	858.34	773	541.15	98	2301	24892	41.37%
	slow traffic	648.21	539	714.43	17	3537	18798	31.24%
	queuing traffic	521.79	506	321.83	32	1316	15132	25.15%
	snow on the road. slow traffic	6.41	0	30.28	0	166	186	0.31%
	slow traffic. queuing traffic	4.83	4	4.68	0	17	140	0.23%
	slow traffic. snow on the road	3.76	0	17.18	0	94	109	0.18%
	slow traffic. stationary traffic	3.31	2	3.78	0	14	96	0.16%
	queuing traffic. stationary traffic	3.10	2	3.58	0	14	90	0.15%
	queuing traffic. slow traffic	2.34	2	2.37	0	8	68	0.11%
	heavy rain. slow traffic	2.14	0	6.46	0	34	62	0.10%
Jam Total		2074.79	1910	1418.82	156	5171	60169	81.57%

Table 4-10: TomTom Event Report Distribution by Category per Day

Table 4-10, shows how TomTom event reports are distributed by category and subcategory per day. The official TomTom documentation does not use the terms category and subcategory. However, the reports contain an "icon category id", which we use as a main category indicator. Additionally, reports contain a "description" and "cause" tag, which are part of a set of 443 incident categories (note that these can be used interchangeably as "description" and "cause"), as explained in Section 3.1.3. We found that 100% of the TomTom event reports contain a "description", whereas only 7.58% of the TomTom event reports contain a "cause". We, therefore, decided to use the description as a subcategory, as is shown in Table 4-10. This table contains the main categories, with the corresponding top 10 subcategories (note that the *Rain* category only contained 3 subcategories). Notice how the *Jam* category is predominant over the other categories accounting for 81.57% of the TomTom event reports. When looking at each category separately it stands out that the most occurring subcategories have very general descriptions, e.g., *accident* (65.03%), *obstruction on the road* (30.88%), *sleet* (36.22%), and *stationary traffic* (41.37%).

Now that we have gained an insight into the TomTom event report distribution per day, we examine the distribution of TomTom event reports and their main categories over a 24-hour period. This could give us valuable insights into the relation of TomTom event reports that are bounded to dayparts, as well as the difference in distribution to our other geosocial data sources. Figure 4-8 depicts the average distribution of the main categories of TomTom event reports of the entire data set plotted on a 24-hour scale. Notice how the number of all TomTom event reports, except for those with the category *Ice* and *Jam* reduce in the nighttime hours. The jam category is the dominant category over the entire day, with peaks during the rush hour periods. Events with the *Accident* and *Dangerous Conditions* category, show a more consistent pattern between 6 am and 20 pm.

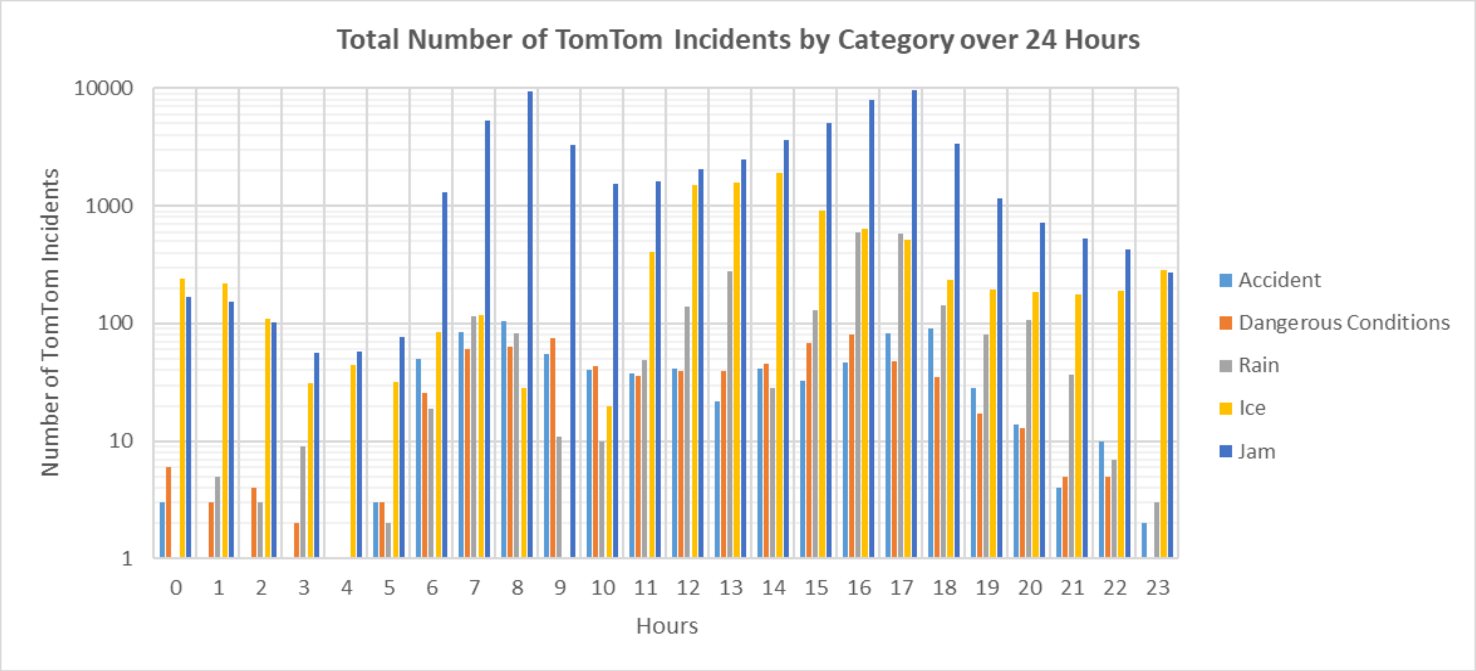


Figure 4-8: Total Number of TomTom Event Reports by Category over 24 Hours

4.2 Rule-based Traffic Domain Annotator

In order to extract relevant traffic domain information from the collected tweet text data, we created a rule-based traffic domain annotator written in the Python3 (v. 3.6.1) language. We used the Python library Pyparsing[39] (v. 2.1.4) as an alternative approach to creating and executing simple grammars. With the help of this annotator, we annotated our entire tweet collection including TE and NTE tweets. Before annotating the entire tweet set, we ran multiple tests on a random sample of 100 tweets of our manually annotated TE tweet set. This way we were able to debug and improve our methods for annotating tweet tokens into 27 unique traffic related categories. Next, we evaluated the annotator based on a randomly selected sample of 200 annotated traffic event-related tweets, not including the tweets used for testing. During the evaluation, we noticed that two tweets were not located in the Netherlands (one in Belgium, and one in South-Africa) and were therefore removed from this analysis. The annotator was able to annotate a total of 1641 token sets, from which 91.77% proved to be annotated with the correct category, whereas 6.09% was categorized incorrectly and 2.13% proved to be too ambiguous to the evaluator to make a clear judgement on. Table 4-11, gives a complete overview of the statistics on the correctness of the evaluation.

Metric	Number of Incorrectly Categorized Token Sets	Number of Unsurely Categorized Token Sets	Number of Correctly Categorized Token Sets	Total Number of Categorized Token Sets
Mean per Tweet	1.32	1.17	7.64	8.33
Median per Tweet	1	1	7	8
Std. Dev. Per Tweet	0.65	0.37	3.41	3.53
Min. Per Tweet	1	1	1	2
Max. per Tweet	4	2	21	21
Total over all Tweets	100	35	1506	1641
Percentage over all Tweets	6.09%	2.13%	91.77%	100.00%

Table 4-11: Rule-based Traffic Domain Annotator Evaluation Metrics

However, as the data is imbalanced, only taking into account these numbers could be misleading. Therefore, we additionally want to get a clear view of the way the annotator categorizes tweets by category. Table 4-12 is a confusion matrix displaying the results of the annotator evaluation. The true positives on the diagonal are highlighted in green. The average precision over all categories is 0.970, the average recall is 0.828, the average f1-score is 0.874, and the average accuracy is 0.964. The confusion matrix further shows how the categories *event_hazard_object* (0.444), *event_hazard_roadwork* (0.375), *event_hazard_violation* (0.462) and *event_traffic_jam* (0.803) negatively deviate from the average f1-score.

[39] http://pyparsing.wikispaces.com/

		PREDICTED CATEGORY (as determined by the annotator)																														
		advice	event_accident	event_closure	event_enforcement	event_hazard_animal	event_hazard_object	event_hazard_road_condition	event_hazard_roadworks	event_hazard_stopped_vehicle	event_hazard_trafficlight	event_hazard_trafficsign	event_hazard_violation	event_hazard_weather	event_trafficjam	media_attachment	n/a	place_location	place_location_combination	place_mile_marker	place_road_direction	place_road_infrastructure	place_road_lane	place_road_section	roaduser_casualty	roaduser_emergency_service	roaduser_person	roaduser_traffic	roaduser_transport	roaduser_vehicle	timex	Total Actual Category
ACTUAL CATEGORY (as determined by the evaluator)	advice	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
	event_accident	0	38	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44
	event_closure	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28
	event_enforcement	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
	event_hazard_animal	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	event_hazard_object	0	0	0	0	0	2	0	0	0	0	0	0	1	0	0	3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	7
	event_hazard_road_condition	0	0	0	0	0	0	22	0	0	0	0	0	3	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31
	event_hazard_roadworks	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	8	0	0	0	1	0	0	1	0	0	0	0	0	0	0	13
	event_hazard_stopped_vehicle	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
	event_hazard_trafficlight	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9
	event_hazard_trafficsign	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
	event_hazard_violation	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
	event_hazard_weather	0	0	0	0	0	0	0	0	0	0	0	0	52	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55
	event_trafficjam	0	0	0	0	0	0	0	0	0	0	0	0	0	55	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	1	76
	media_attachment	0	0	0	0	0	0	0	0	0	0	0	0	0	0	71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	71
	n/a	0	2	1	0	0	0	1	0	0	0	0	0	1	6	0	683	7	0	0	0	0	0	1	1	0	0	0	0	0	1	704
	place_location	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	174	1	0	0	0	0	0	0	0	0	0	0	0	0	188
	place_location_combination	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	62	0	0	0	0	0	0	0	0	0	0	0	0	69
	place_mile_marker	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	8	0	0	0	0	0	0	0	0	0	0	0	9
	place_road_direction	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	19	0	0	0	0	0	0	0	0	0	0	21
	place_road_infrastructure	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2	1	0	0	19	0	0	0	0	0	0	0	0	0	25
	place_road_lane	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	27	0	0	0	0	0	0	0	0	29
	place_road_section	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	26	0	0	0	0	0	0	0	27
	roaduser_casualty	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	10
	roaduser_emergency_service	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	21	0	0	0	0	0	23
	roaduser_person	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	16	0	0	0	0	18
	roaduser_traffic	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	4
	roaduser_transport	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	11
	roaduser_vehicle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	52	0	55
	timex	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	1	0	0	0	0	0	0	73	78
	Total Predicted Category	6	40	26	6	1	2	23	3	5	7	3	3	57	61	71	785	183	65	8	20	19	27	29	12	21	16	4	11	52	75	**1641**
	Precision	1.000	0.950	0.962	1.000	1.000	1.000	0.957	1.000	1.000	1.000	1.000	1.000	0.912	0.902	1.000	0.870	0.951	0.954	1.000	0.950	1.000	1.000	0.897	0.833	1.000	1.000	1.000	1.000	1.000	0.973	**0.970**
	Recall	0.857	0.864	0.893	0.857	1.000	0.286	0.710	0.231	0.714	0.778	0.750	0.300	0.945	0.724	1.000	0.970	0.926	0.899	0.889	0.905	0.760	0.931	0.963	1.000	0.913	0.889	1.000	1.000	0.945	0.936	**0.828**

Table 4-12: Rule-based Traffic Domain Annotator Confusion Matrix

After evaluating the annotator, we first used it to annotate the tweet set from 05-12-2017 to 06-01-2018, as this set was going to be used to train our traffic event classifier. Table 4-13, shows how the annotator annotated the TE tweets over this period per day with unique categories. Note how the event categories *event_accident*, *event_closure*, *event_weather* and *event_trafficjam* have a significantly higher rate of appearance than the other event categories. Besides, keep in mind that a tweet can have multiple event categories of different types, that is why the *n/a* category has such a high percentage of 99.35% as a tweet almost always contains a token set that cannot be annotated.

Category	Mean	Median	Std dev	Min	Max	Total	Percentage
advice	2.12	2	2.21	0	9	70	3.79%
event_accident	9.18	7	5.66	2	21	303	16.40%
event_closure	8.82	9	6.57	0	33	291	15.76%
event_enforcement	0.64	0	0.73	0	2	21	1.14%
event_hazard_animal	0.79	0	1.01	0	3	26	1.41%
event_hazard_object	0.91	1	1.08	0	4	30	1.62%
event_hazard_road_condition	5.27	2	8.07	0	35	174	9.42%
event_hazard_roadwork	3.91	3	3.28	0	13	129	6.98%
event_hazard_stopped_vehicle	2.24	1	1.94	0	6	74	4.01%
event_hazard_trafficlight	1.58	1	1.46	0	5	52	2.82%
event_hazard_trafficsign	0.91	1	1.22	0	4	30	1.62%
event_hazard_violation	0.70	0	1.06	0	5	23	1.25%
event_hazard_weather	8.67	2	18.70	0	93	286	15.48%
event_trafficjam	12.30	10	14.85	1	84	406	21.98%
media_attachment	23.55	19	24.81	3	126	777	42.07%
n/a	55.61	41	58.06	10	320	1835	99.35%
place_infrastructure_type	4.00	3	4.64	0	25	132	7.15%
place_location	38.88	27	44.27	6	245	1283	69.46%
place_location_combination	18.94	11	17.59	3	93	625	33.84%
place_mile_marker	7.00	6	4.79	0	20	231	12.51%
place_road_direction	6.27	4	8.07	0	43	207	11.21%
place_road_lane	10.36	7	9.12	1	49	342	18.52%
place_road_section	6.64	5	8.02	0	41	219	11.86%
roaduser_casualty	1.79	2	1.07	0	4	59	3.19%
roaduser_emergency_service	5.30	5	2.80	1	14	175	9.47%
roaduser_general	3.70	2	5.11	0	26	122	6.61%
roaduser_traffic	2.58	2	2.45	0	11	85	4.60%
roaduser_transport	3.67	3	4.73	0	26	121	6.55%
roaduser_vehicle	13.21	8	14.60	1	76	436	23.61%
timex	16.33	12	19.29	3	107	539	29.18%

Table 4-13: Annotated Twitter Collection Metrics by Category from 05-12-17 to 06-12-18

Additionally, just as with Waze and TomTom event reports, we examined the distribution of tweet event reports and their main traffic event categories over a 24-hour period. Figure 4-9 depicts the average distribution of the main categories of tweet event reports of the entire data set plotted on a 24-hour scale. A clear decrease in all traffic events in noticeable during the nighttime hours. Additionally, an increase in events is visible during the rush hour periods, especially for *event_trafficjam* typed events.

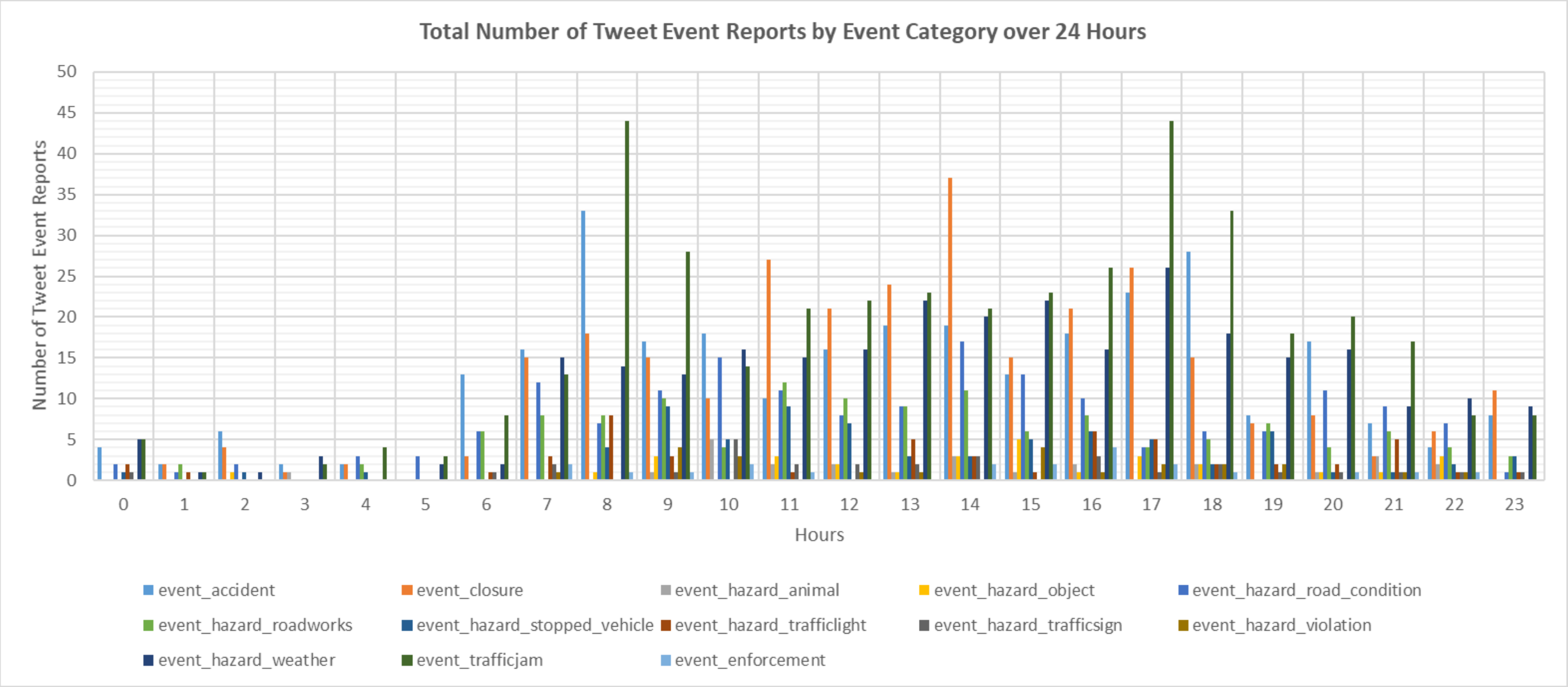


Figure 4-9: Total Number of Tweet Event Reports by Event Category over 24 Hours

4.3 Traffic Event Classification

Supervised binary classification was applied in order to predict if a tweet is either traffic event-related or non-traffic event-related. This classifier was written in the Python3 (v. 3.6.1) language. We used the Python library Scikit-learn[40] (v. 0.19.1), which provided us with the tools for the data analysis. Additionally, we used the Python library Imbalanced-learn[41] (v. 0.3.2), which provided us with under- and over-sampling methods. We used our manually labeled tweet collection set as described in Section 4.1.2, for training and validation purposes. The tweet collection thus consists of 29,452 tweets, from which 27,683 labeled as NTE and 1769 labeled as TE. In order to perform machine learning on tweet text documents, the text content had to be turned into numerical feature vectors. Therefore, we tokenized this tweet set with a special Dutch-based tokenizer designed for Twitter text named Ucto[42]. Additionally, we filtered out stopwords based on a Dutch stopword list[43]. We did not apply the pre-processing technique of lemmatization as stated in our experiment design, as the Frog NLP lemmatizer[44] proved to be too time- as well as computationally expensive for this study. Subsequently, we engineered a number of features based on the tweet text documents, namely: n-grams, tf-idf, and syntactic features. We did not use PoS tagging as stated in our experiment design, as one again the Frog NLP PoS tagger proved too time- as well as computationally expensive for this study. Subsequently, the entire tokenized tweet set was annotated by our rule-based traffic domain annotator (note that we did not filter on stopwords during this process). Based on these tokenized tweet text documents we engineered n-gram and tf-idf features.

After defining all our features we were able to train a classifier to predict the category of a tweet. We started out with a Multinomial Naïve Bayes typed classifier, which is a variant often used for text classification purposes. We then applied a stratified 10-fold cross validation method on the data in order to evaluate the different combinations of features and to estimate how accurately the model performs in practice. Table 4-14, shows the result with the best performance based on a combination of average f1-score (0.94), accuracy (0.935) and AUC ROC score (0.873), after evaluating with different feature combinations and parameters. Note however the extreme differences in precision and recall between non-traffic event-related (0) and traffic event-related (1) tweets. One possible explanation for this discrepancy is the imbalance of the two datasets. Therefore, we repeated the experiment and tried to compensate for this imbalance by resampling the dataset with over- and under-sampling techniques. In Table 4-15, it can be seen that with a random over sampling technique the precision for detecting traffic event-related tweets is reduced with 10 percentage points while the recall is increased with 25 percentage points. Table 4-16, shows the result when applying a random under sampling technique. Although, here the recall is increased to 0.86 for TE tweets, the precision is even further decreased to 0.27.

[40] http://scikit-learn.org/stable/index.html
[41] https://github.com/Toblerity/Shapely
[42] https://github.com/proycon/python-ucto
[43] https://github.com/stopwords-iso/stopwords-nl
[44] https://languagemachines.github.io/frog/

Grid Search		
Classifier	MultinomialNB	
Features	Tweet Text Character N-gram	n-gram = (1, 4)
	Tweet Text Annotated Word N-gram	n-gram = (1, 4)
Cross Validation	10-fold	

Classification Report					Confusion Matrix		
	Precision	Recall	F1-score	Support		Predicted: 0	Predicted: 1
0	0.97	0.96	0.96	27600	Actual: 0	26616	1067
1	0.46	0.52	0.49	1769	Actual: 1	846	923
Avg/total	0.94	0.94	0.94	29452			

Metrics	
Accuracy	0.935
AUC_ROC	0.873

Table 4-14: MultinomialNB based Classification Metrics

Grid Search		
Resampled	RandomOverSampler	
Classifier	MultinomialNB	
Features	Tweet Text Character N-gram	n-gram = (1, 5)
	Tweet Text Annotated tf-idf	n-gram = (1, 3)
Cross Validation	10-fold	

Classification Report					Confusion Matrix		
	Precision	Recall	F1-score	Support		Predicted: 0	Predicted: 1
0	0.98	0.91	0.95	27683	Actual: 0	25246	2437
1	0.36	0.77	0.49	1769	Actual: 1	400	1369
Avg/total	0.95	0.90	0.92	29452			

Metrics	
Accuracy	0.904
AUC_ROC	0.921

Table 4-15: MultinomialNB Oversampled based Classification Metrics

Grid Search		
Resampled	RandomUnderSampler	
Classifier	MultinomialNB	
Features	Tweet Text Character N-gram	n-gram = (1, 5)
	Tweet Text Word N-gram	n-gram = (1, 3)
	Tweet Text Annotated tf-idf	n-gram = (1, 3)
Cross Validation	10-fold	

Classification Report					Confusion Matrix		
	Precision	Recall	F1-score	Support		Predicted: 0	Predicted: 1
0	0.99	0.85	0.91	27683	Actual: 0	23529	4154
1	0.27	0.86	0.41	1769	Actual: 1	255	1514
Avg/total	0.95	0.85	0.88	29452			

Metrics	
Accuracy	0.850
AUC_ROC	0.911

Table 4-16: MultinomialNB Undersampled based Classification Metrics

Besides a Naïve Bayes typed classifier, a Support Vector Machine typed classifier was used. Table 4-17, shows the results with the best performance with the original unbalanced dataset. Note how by changing the classifier type, a different feature set with different parameters becomes more effective compared to the best set with a Naïve Bayes typed classifier. An improvement in average f1-score (0.95), accuracy (0.956) and AUC ROC score (0.940) can be observed, compared to all results where a Naïve Bayes typed classifier was used. Additionally, we applied over- and under-sampling techniques as the results in Table 4-18 and Table 4-19 show. Table 4-18, shows a slight improvement in recall score for TE tweets (0.61) but at the cost of a slight decrease in precision (0.62). However, the overall AUC ROC did improve when using random oversampling. When applying a random under-sampling method, as shown in Table 4-19, the precision for TE tweets decreases drastically to 0.31, while the recall improves to a score of 0.88. Also, the average f1-score (0.90), and accuracy (0.874) are impaired.

Grid Search		
Classifier	LinearSVM	
Features	Tweet Text Character N-gram	n-gram = (1, 6)
	Tweet Text Annotated Word N-gram	n-gram = (1, 3)
	Tweet Text Annotated tf-idf	n-gram = (1, 3)
	Tweet Text Word N-gram	n-gram = (1, 2)
Cross Validation	10-fold	

Classification Report					Confusion Matrix		
	Precision	Recall	F1-score	Support		Predicted: 0	Predicted: 1
0	0.97	0.98	0.98	27683	Actual: 0	27124	559
1	0.65	0.58	0.61	1769	Actual: 1	746	1023
Avg/total	0.95	0.96	0.95	29452			

Metrics	
Accuracy	0.956
AUC_ROC	0.940

Table 4-17: LinearSVM based Classification Metrics

Grid Search		
Resampled	RandomOverSampler	
Classifier	LinearSVM	
Features	Tweet Text Annotated tf-idf	n-gram = (1, 3)
	Tweet Text tf-idf	n-gram = (1, 3)
Cross Validation	10-fold	

Classification Report					Confusion Matrix		
	Precision	Recall	F1-score	Support		Predicted: 0	Predicted: 1
0	0.97	0.98	0.98	27683	Actual: 0	27034	667
1	0.62	0.61	0.61	1769	Actual: 1	696	1073
Avg/total	0.95	0.95	0.95	29452			

Metrics	
Accuracy	0.954
AUC_ROC	0.955

Table 4-18: LinearSVM Oversampled based Classification Metrics

Grid Search		
Resampled	RandomUnderSampler	
Classifier	LinearSVM	
Features	Tweet Text Annotated Word N-gram	n-gram = (1, 1)
	Tweet Text tf-idf	n-gram = (1, 1)
Cross Validation	10-fold	

Classification Report				
	Precision	Recall	F1-score	Support
0	0.99	0.87	0.93	27600
1	0.31	0.88	0.45	1769
Avg/total	0.95	0.87	0.90	29452

Confusion Matrix		
	Predicted: 0	Predicted: 1
Actual: 0	24179	3504
Actual: 1	217	1552

Metrics	
Accuracy	0.874
AUC_ROC	0.942

Table 4-19: LinearSVM Undersampled based Classification Metrics

Before making a decision on the choice of which traffic event classifier to use from here on out, it is important to reconsider what the most important aspect of the classifier should be in our situation. On the one hand, one could say that it is important to have both an as high as possible precision and recall value for detecting traffic event-related tweets. This because our main goal is to detect as many true positive instances of traffic event-related tweets as possible, but to reduce the number of false negatives as they contaminate our set. On the other hand, one could say that we want an as high as possible recall value for detecting traffic event-related tweets, even if this comes at the cost of a lower precision value. This because traffic event-related tweets get clustered with Waze and TomTom event reports later on in the pipeline anyway. One could count on false negative tweets to get exposed in that stage as they most likely will not meet the requirements for getting clustered.

We decided to go with the most reliable option by choosing the classifier with the best combination of precision and recall values for detecting traffic event-related tweets. This classifier is based on Linear SVM with a random over sampler and performs best based on the combination of average f1-score of 0.95, accuracy of 0.954 and AUC ROC of 0.955, as can be found in Table 4-18. We persisted this model with pickle, a Python module that enables objects to be serialized to files on disk and deserialized back into the program at runtime. This model was then applied to the second half of the Twitter data set ranging from 07-01-2018 to 17-02-2018. A visual overview of the distribution of tweets classified as TE or NTE can be found in Figure 4-10. Note how the daily distribution of TE/NTE tweets is similar to the graph displayed in Figure 4-4. This is also reflected in the quantitative results as shown in Table 4-20, e.g., the manually labeled mean of TE related tweets per day is 56/892 (6.3%) compared to 48/857 (5.6%) as calculated by our classifier.

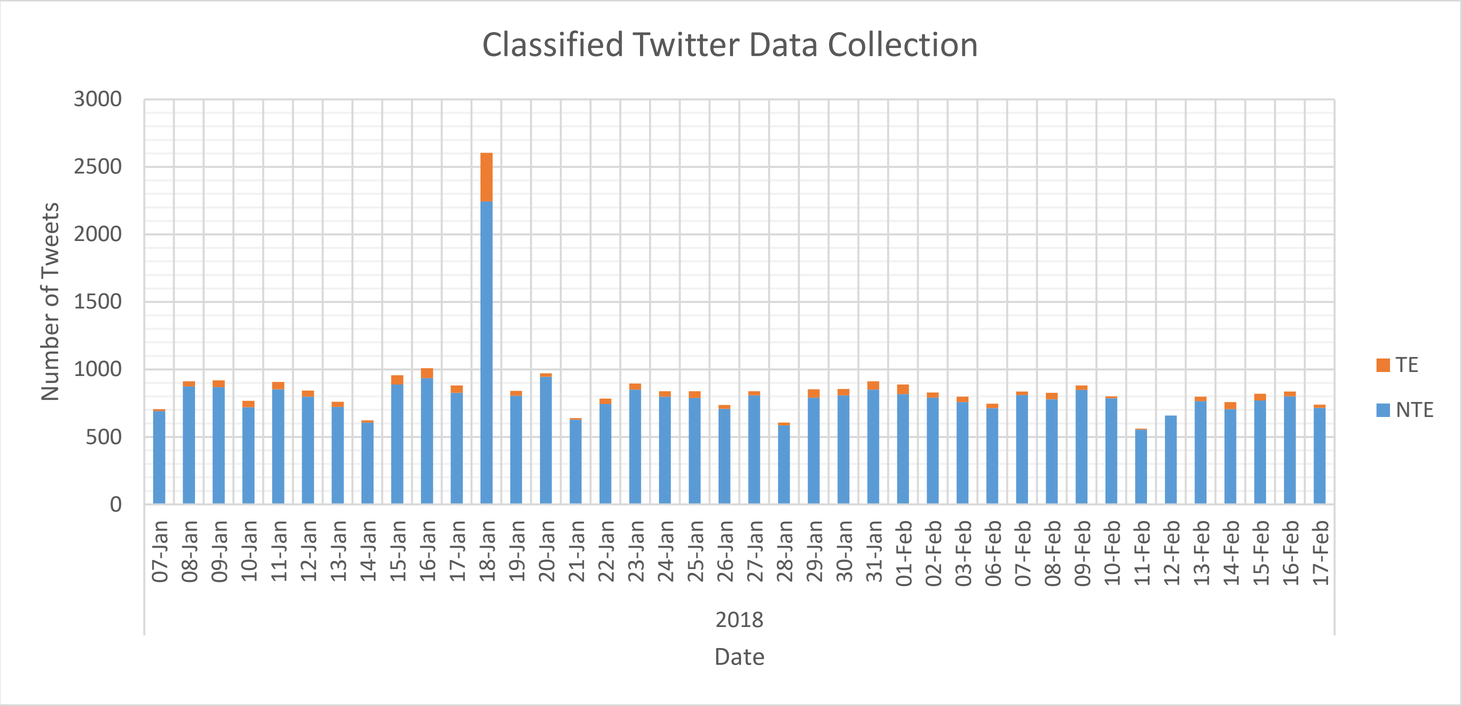


Figure 4-10: Tweets collected with the Twitter data collection approach, classified as TE or NTE by our trained Linear SVM based classifier

Classified Twitter Data Collection over 07-01-2018 – 17-02-2018			
Metric	**# of NTE related tweets**	**# of TE related tweets**	**Total # of tweets**
Mean per day	810	48	857
Median per day	791	41	837
Std. Dev. Per day	245	53	297
Min. per day	553	9	562
Max. per day	2243	361	2604
Total	32,399	1875	2604

Table 4-20: Classified Twitter Data Collection Metrics 07-01-2018 to 17-02-2018

4.4 Geocoding

A geocoding method was created that uses spatial indicators in tweets, as annotated by our rule-based traffic domain annotator. The method links these spatial indicators to a geographic location and uses an intersection technique to find a list of the most relevant locations in a tweet. This geocoding module was written in the Python3 (v. 3.6.1) language. We used a Python library Googlemaps[45] (v. 2.5.1), which allowed us to use the Google Directions API and Google Places API in Python. Additionally, we used the Python library Shapely[46] (v. 1.6) to manipulate and analyze geometric objects. We already gave a comprehensive overview of our geocoding approach in Section 3.6.1, however we will shortly discuss some additional interesting details/restrictions that came to light during the implementation. In all cases a suitable solution was applied, unless otherwise stated.

1. Google Places API, used to link a token with a *place_location* category to a geographical location:
    a. Tokens appended with a "#" deliver different results than tokens without.
    b. The API returns two geometry-related results for a queried place. The first one, called "location" provides the latitude and longitude of the place, while the second one "viewport" provides the preferred viewport on the map when viewing this place. We use this viewport to create a bounding box, as this better represents the location than a single coordinate.
    c. In some random cases, the API includes a shape of the entire Netherlands in its results, while querying for a single small place within the Netherlands.
    d. The API can only return up to 60 results for a single query, e.g., the query "McDonald's" only returns 60 locations for a McDonald's in the Netherlands, even though there are 245 establishments in the Netherlands. For this limitation no suitable solution was found.
    e. The API has a default limit of 1,000 free requests per 24 hour period, calculated as the sum of client-side and server-side requests. In order to overcome this limitation we increased this limit free of charge up to 150,000 requests per 24 hour period, by enabling billing by verifying our identity with a credit card.
2. Google Places API, used to link a token with a *place_location_combination* category to a geographical location:
    a. The API enables to query for places that are in the vicinity of other places by using the following format: "placeA near placeB". This way a more precise location can be gathered, however the API returns different results when switching the tokens, in other words "placeB near placeA" returns a different result. Additionally, the API provides a Dutch alternative to the keyword "near", namely "in de buurt van", however again this provides different results than when using the English version "near". We made the decision to keep the order of the tokens in the way they appear in the tweet, and use the Dutch keyword "in de buurt van". This because we want to stay as close as possible to the place intended by the writer of the tweet, and as the tweets are Dutch it seemed logical to use the Dutch version of the API.

[45] https://github.com/googlemaps/google-maps-services-python

[46] https://github.com/Toblerity/Shapely

3. Road database, used to link a token with a place *road_mile_marker* category to a geographical location:
    a. As our road database is based on data from Rijkswaterstaat Ministry of Infrastructure and Water Management, dating back to November 2015, there could be some instances where roads have been updated. Also, only A- and N-roads are included, causing us to miss out on so called S-, E- and r-roads. In these cases the tokens are used to query the Google Places API.
    b. Tweets do not always contain an existing mile marker number, however in combination with a road number it could still be useful. Therefore, we round mile marker numbers to one decimal and query the database for the closest related number.
4. Google Directions API, used to link a token with a *place_road_section* or *place_road_direction* to a geographical location:
    a. This API contains a parameter "mode" that specifies the mode of transport to use when calculating directions. As we are first and foremost interested in traffic events on roads that allow motorized vehicles we set this parameter to "driving".
    b. This API contains a parameter "alternatives" which specifies that the service may provide more than one route alternative in the response. We set this parameter to "true" as we want to retrieve as many as possible routes between two locations as possible, so that we do not miss out on road locations.
    c. This API needs an "origin" token as start location and a "destination" as end location. However, the response could be different depending on the order of tokens. For example, Delft → Rotterdam gives different results than Rotterdam → Delft. We made the decision to keep the order of the tokens in the way they appear in the tweet, as we want to stay as close as possible to the place intended by the writer of the tweet.

With the help of our annotator we annotated the TE tweets, which provided us with the needed place related categorized tokens for each tweet. We then ran multiple tests on a on a random sample of 100 tweets, in order to debug and improve our geocoding methods. Next, we evaluated the geocoding module based on a randomly selected sample of 100 geocoded traffic event-related tweets from the period 05-12-2017 to 06-01-2018, which were not included in the test set. As place mentions in tweets are highly ambiguous, a geocoded location cannot be either correct or incorrect. Therefore, each tweet got evaluated on how well the geocoded locations suit the contents of the tweet. For this, a custom category ranking system was used as previously described in Section 3.6.2.

Table 4-21 on the next page, shows how the 100 tweets have been evaluated into the four different categories. It shows that the majority (49%) of the tweets can be geocoded to a location that covers all place indicators in the tweet and includes no irrelevant locations. Additionally, 37% of the geocoded tweets include all relevant place indicators, however also a number of irrelevant place indicators. The remaining 14% of the tweets either is geocoded to a part of relevant indicators or to no relevant indicators at all. We also computed the distribution of the geocoding methods for each category, and the existence of a cross intersection between the derived sub locations in a tweet (Cross Intersection/ No Cross Intersection). Besides, we looked at the influence of the number of place-related token sets in a tweet. Note that this number is significantly higher in categories B, C and D compared to category A.

Category	Distribution of Geocoding Methods	>= 1 Token	>= 2 Tokens	>= 3 Tokens	>= 4 Tokens	5 Tokens
A						
Cross Intersection	9.00%	9	9	0	0	0
Place Direction/Section	1.00%	1	0	0	0	0
Place Location	9.00%	9	0	0	0	0
Place Location Combination	11.00%	11	0	0	0	0
No Cross Intersection	3.00%	3	3	1	0	0
Place Road Mile Marker	16.00%	16	0	0	0	0
Total A	**49.00%**	**49**	**12**	**1**	**0**	**0**
B						
Cross Intersection	6.00%	6	6	5	2	0
Place Location	5.00%	5	0	0	0	0
Place Location Combination	1.00%	1	0	0	0	0
No Cross Intersection	20.00%	20	20	10	3	2
Place Road Mile Marker	5.00%	5	0	0	0	0
Total B	**37.00%**	**37**	**26**	**15**	**5**	**2**
C						
Cross Intersection	1.00%	1	1	1	0	0
Place Direction/Section	1.00%	1	0	0	0	0
Place Location	2.00%	2	0	0	0	0
No Cross Intersection	3.00%	3	3	2	0	0
Total C	**7.00%**	**7**	**4**	**3**	**0**	**0**
D						
Cross Intersection	3.00%	3	3	1	0	0
Place Location	1.00%	1	0	0	0	0
No Cross Intersection	3.00%	3	3	1	0	0
Total D	**7.00%**	**7**	**6**	**2**	**0**	**0**
End Total	**100.00%**	**100**	**48**	**21**	**5**	**2**

Table 4-21: Geocoding Evaluation Metrics

4.5 Traffic Event Description

A traffic event description module was developed to cluster related information from traffic event reports (TE tweets, Waze and TomTom events) and DiTTLab traffic data. This traffic event description module was written in the Python3 (v. 3.6.1) language. We already gave a comprehensive overview of our traffic event description approach in Section 3.7, however we will shortly discuss some additional interesting details regarding the used sources and clustering techniques.

The clustering of traffic event reports is based on a rule-based approach, in which a rule specifies the categorical, spatial and temporal extent, used to assert if the new traffic event report should be part of an existing traffic event cluster. Traffic event reports are matched on category, based on the 13 event categories described in Section 3.3.1. In the previous sections we showed how a tweet gets collected, categorized as TE or NTE, its tokens annotated to a variety of categories by our annotator, and finally geocoded. Therefore a tweet is already matched to zero or more of the 13 event categories. However, Waze and TomTom event reports use their own category definitions as explained in Sections 3.1.2, and 3.1.3. Hence, we created a traffic event rule collection containing each of the 13 event categories, matched to its corresponding TomTom and Waze categories, as the two examples show in Table 4-22 and Table 4-23.

Category	Event_accident
TomTom Category Icon ID	1
TomTom Categories	Secondary accident
	Chemical spillage accident
	Fuel spillage accident
	Accident clearance
	Multi-vehicle accident
	Serious accident
	(…17 more…)
Waze Categories	ACCIDENT
	ACCIDENT_MINOR
	ACCIDENT_MAJOR

Table 4-22: Traffic Event Rule Collection - Event_accident

Category	Event_hazard_trafficlight
TomTom Category Icon ID	3
TomTom Categories	Traffic lights not working
	Traffic lights working incorrectly
	Temporary traffic lights working incorrectly
Waze Categories	HAZARD_ON_ROAD_BROKEN_TRAFFIC_LIGHT

Table 4-23: Traffic Event Rule Collection - Event_hazard_trafficlight

Next, we take a closer look at how traffic event reports are clustered together based on locational features. Tweets were geocoded to a geoshape, however, Waze and TomTom only contain a single geopoint as location. We therefore, drew a radius of 100 meters around Waze and TomTom locations, in the same way as with tweets. Next, we intersected traffic event reports based on their geoshape and an additional radius. This was done because of the possible delay existing between the traffic event location and the location of the creation of the traffic event report. We have chosen to have this radius customizable for each traffic event category. For example, a traffic event with the category *Event_hazard_violation* would only be relevant within a radius of 100 meters if it was about a reckless driver. While a traffic event with the category *Event_traffic_jam* would be relevant within a radius of 1 kilometer if it was about a heavy traffic jam. For the evaluation of our traffic event description method, we used a radius of 150 meters for each of the different categories.

Besides categorical and spatial features, traffic event reports were clustered together based on a temporal feature. For this, the creation dates of each traffic event report were used. A timespan calculated in minutes from the creation time of the traffic event report, was used to represent the temporal extent of a traffic event. We used a timespan, because the time extracted from a traffic event description does not necessarily represent the exact time a traffic event took place. We have chosen to have this timespan customizable for each traffic event category. For example, a traffic event with the category *Event_hazard_violation* would only be relevant for a couple of minutes for a specific location if it was about a reckless driver. While, a traffic event with the category *Event_trafficjam* could be relevant for an hour for the same location if it was about a heavy traffic jam. For the evaluation of our traffic event description method, we used a time span of 15 minutes for each of the different categories.

Note, that when a traffic event report does not match a previous traffic event report on either its category, location, or timespan it is not discarded. This event report is just seen as the starting point of a new traffic event cluster, and therefore regarded as unrelated to the previous event cluster. An example of an traffic event cluster can be found in Table 4-24.

Key	Value	Type
_id	5afc2974e8b2900e404e9ce6	ObjectId
mainReportID	europe_HD_NL_TTL116026360217778	String
mainReportType	{ 13 fields }	Object
mainReportCategory	[1 elements]	Array
0	Event_trafficjam	String
mainReportLocation	{ 2 fields }	Object
mainReportLocationArea	0.13419460479408543	Double
mainReportDate	2017-12-06T16:13:18.246Z	Date
subClusters	[1 elements]	Array
0	{ 3 fields }	Object
subReportsCategory	Event_trafficjam	String
subReportsIntersectedLocation	{ 2 fields }	Object
subReportsIntersectedLocationArea	0.0	Double
subReportsMainReportRelevance	0.0	Double
subReports	[5 elements]	Array
0	{ 10 fields }	Object
subReportID	alert-907664744/9d48c9bc-3c49-3a94-901a-7c29e92bcf82	String
subReportType	{ 16 fields }	Object
subReportLocation	{ 2 fields }	Object
subReportLocationArea	0.02377992076632436	Double
subReportIntersectedLocation	{ 2 fields }	Object
subReportIntersectedLocationArea	0.05402956385563263	Double
subReportDate	2017-12-06T16:14:45.686Z	Date
subReportRelevanceSub	0.36332033860283597	Double
mainReportRelevanceSub	0.4026209841933523	Double
subReportRelevanceTotal	0.01635243657126755	Double
1	{ 10 fields }	Object
2	{ 10 fields }	Object

Table 4-24: Traffic Event Cluster (some keys have been collapsed, for readability purposes)

4.5.1 Traffic Event Description Evaluation

We applied our traffic event description method on the collected Twitter, Waze, TomTom and DiTTLab data over the period from 05-12-2017 to 06-01-2018. This provided us with a total of 210,614 unique clusters, consisting of 33,528 TomTom, 1411 tweet, and 175,675 Waze main event reports. Of these 1411 tweet reports, 540 did not contain a traffic event category (*N/A*). This does not mean that the tweet is not traffic event-related, it just could not get assigned to a traffic event-related category by our annotator, and to a location by our geocoder. Therefore, due to our strict linking based on category, location and time, the tweet report could not be linked to another event report. The same applies for 2425 Waze event reports that could not be linked based on category as they contained an abstract *HAZARD* category without any subcategories, or contained the category *CHIT CHAT*, which also could not be linked to any of our traffic event categories specified in our traffic event rule collection. Table 4-25, shows how subreports were linked to their related main reports, and by what type of event category. Keep in mind, that a main report can have multiple categories, therefore the end totals deviate from the number of unique main cluster reports. When taking a closer look at each main report type, it becomes visible that TomTom event reports are most likely to be linked to another TomTom event report. Whereas tweet reports are more likely to link to a Waze report, and Waze reports to other Waze reports. However, even more noticeable are the number of cases a category from a main event report cannot be linked to a related traffic event report: TomTom event reports (41.57%), tweet event reports (85.30%), and Waze event reports (81.89%). As stated in Section 4.1.1, due to some technical issues there were some days that we were not able to collect tweet, TomTom, and Waze event reports for the complete day or did not receive any reports at all. We compensate for this issue by filtering out those days no data was available, the results are shown in Table 4-26. Note how the linkage results for tweet event reports with TomTom reports improve (1.84% → 6.42%), as well as the results for Waze event reports with TomTom reports (5.74% → 20.34%).

Additionally, we applied our traffic event description method on the collected Twitter, Waze, TomTom and DiTTLab data over the period from 07-12-2017 to 17-02-2018. We specifically split the sets over two periods, as the first time period contains our manually annotated tweet reports and the second period the tweet reports classified by our machine learning based classifier. This way we are able to clearly show the difference between the two. Over the second period we obtained 260,656 unique clusters, consisting of 58,673 TomTom, 1517 tweet, and 200,466 Waze main event reports. Of the 1517 tweet reports, 646 did not contain a traffic event category (N/A). The same applies for 1803 Waze and 12 TomTom event reports. Table 4-25, shows how subreports were linked to their related main reports, and by what type of event category. Again, we also compensate the data for dates no data could be collected, which is shown in Table 4-26. When comparing the compensated data from period 1 to period 2, some categories show significant differences. For example, the percentage of linked TomTom event reports to Waze reports increases from 7.52% to 23.48% while the linkage towards TomTom event reports reduces from 52.04% to 20.88%.

	Number of Linked Subreports									
Main Reports	waze_report		tweet_report		tomtom_report		N/A		End Total	
TomTom Event Report	P1	P2	P1	P2	P1	P2	P1	P2	P1	P2
event_accident	7.03%	6.38%	2.34%	3.38%	37.50%	46.15%	53.13%	44.09%	128	533
event_closure	3.23%	9.36%	0.00%	2.25%	61.29%	39.70%	35.48%	48.69%	31	267
event_hazard_animal	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	100.00%	4	5
event_hazard_object	4.00%	1.69%	0.00%	0.00%	12.00%	14.35%	84.00%	83.97%	50	237
event_hazard_road_condition	0.00%	0.00%	6.67%	0.00%	40.00%	42.86%	53.33%	57.14%	15	7
event_hazard_roadwork	11.81%	15.10%	0.37%	1.01%	42.07%	36.91%	45.76%	46.98%	271	298
event_hazard_traffic_light	0.00%	6.29%	0.00%	0.63%	0.00%	47.17%	0.00%	45.91%	0	159
event_hazard_stopped_vehicle	15.94%	0.00%	0.00%	0.00%	47.83%	0.00%	36.23%	100.00%	69	2
event_hazard_violation	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	100.00%	2	12
event_hazard_weather	1.13%	1.58%	1.48%	0.20%	84.02%	72.80%	13.37%	25.42%	11613	1011
event_trafficjam	12.09%	20.64%	0.65%	0.50%	30.57%	20.15%	56.69%	58.72%	21345	56130
Total TomTom Event Report	**8.25%**	**19.98%**	**0.94%**	**0.53%**	**49.24%**	**21.51%**	**41.57%**	**57.98%**	**33528**	**58661**
Tweet Event Report										
event_accident	6.79%	3.77%	6.17%	5.02%	1.23%	5.86%	85.80%	85.36%	162	239
event_closure	3.61%	3.85%	4.64%	7.14%	1.55%	1.10%	90.21%	87.91%	194	182
event_enforcement	7.69%	0.00%	7.69%	0.00%	0.00%	0.00%	84.62%	100.00%	13	22
event_hazard_animal	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	100.00%	17	11
event_hazard_object	5.56%	11.54%	0.00%	0.00%	0.00%	0.00%	94.44%	88.46%	18	26
event_hazard_road_condition	4.90%	5.56%	3.92%	3.70%	0.98%	0.00%	90.20%	90.74%	102	54
event_hazard_stopped_vehicle	2.94%	2.22%	2.94%	0.00%	2.94%	4.44%	91.18%	93.33%	34	45
event_hazard_violation	0.00%	0.00%	0.00%	5.88%	0.00%	0.00%	100.00%	94.12%	10	17
event_hazard_weather	12.43%	2.25%	7.34%	22.47%	3.39%	0.00%	76.84%	75.28%	177	89
event_trafficjam	13.89%	14.52%	4.17%	2.15%	2.08%	11.29%	79.86%	72.04%	144	186
Total Tweet Event Report	**7.81%**	**5.97%**	**5.05%**	**5.97%**	**1.84%**	**4.48%**	**85.30%**	**83.58%**	**871**	**871**
Waze Event Report										
event_accident	5.14%	6.40%	0.71%	0.73%	1.23%	3.68%	92.92%	89.20%	3501	4000
event_closure	0.00%	0.00%	0.34%	0.59%	0.00%	0.08%	99.66%	99.33%	2365	2388
event_enforcement	4.10%	5.57%	0.02%	0.00%	0.00%	0.00%	95.88%	94.43%	5295	6908
event_hazard_animal	1.34%	0.00%	0.00%	0.00%	0.00%	0.00%	98.66%	100.00%	299	326
event_hazard_object	0.69%	1.51%	0.09%	0.26%	0.43%	1.84%	98.80%	96.39%	1166	1525
event_hazard_road_condition	6.25%	3.43%	1.27%	0.00%	0.00%	0.00%	92.48%	96.57%	1184	991
event_hazard_roadwork	0.00%	0.00%	0.02%	0.00%	0.04%	0.19%	99.94%	99.81%	4802	5371
event_hazard_stopped_vehicle	3.47%	5.01%	0.04%	0.04%	0.04%	0.13%	96.45%	94.82%	44246	45373
event_hazard_traffic_sign	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	100.00%	220	279
event_hazard_weather	3.78%	3.59%	1.76%	0.26%	5.36%	0.45%	89.10%	95.70%	6479	2674
event_trafficjam	17.28%	19.02%	1.02%	1.02%	9.18%	18.55%	72.52%	61.41%	103836	128828
Total Waze Event Report	**11.65%**	**13.88%**	**0.72%**	**0.70%**	**5.74%**	**12.16%**	**81.89%**	**73.26%**	**173250**	**198663**

Table 4-25: Subreports linked to Main Reports by Category, where P1 equals the period from 05-12-2017 to 06-01-2018 and P2 equals the period from 07-01-2018 to 17-02-2018

	Number of Linked Subreports									
Main Reports	waze_report		tweet_report		tomtom_report		N/A		End Total	
TomTom Event Report	P1	P2	P1	P2	P1	P2	P1	P2	P1	P2
event_accident	6.93%	7.32%	2.97%	3.10%	34.65%	47.01%	55.45%	42.57%	101	451
event_closure	3.23%	10.30%	0.00%	2.58%	61.29%	39.06%	35.48%	48.07%	31	233
event_hazard_animal	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	100.00%	2	4
event_hazard_object	2.44%	2.15%	0.00%	0.00%	7.32%	13.98%	90.24%	83.87%	41	186
event_hazard_road_condition	0.00%	0.00%	6.67%	0.00%	40.00%	0.00%	53.33%	100.00%	15	3
event_hazard_roadwork	10.04%	16.30%	0.40%	1.11%	43.78%	32.96%	45.78%	49.63%	249	270
event_hazard_traffic_light	0.00%	7.69%	0.00%	0.77%	0.00%	46.92%	0.00%	44.62%	0	130
event_hazard_stopped_vehicle	10.71%	0.00%	0.00%	0.00%	53.57%	0.00%	35.71%	100.00%	56	2
event_hazard_violation	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	100.00%	2	9
event_hazard_weather	1.13%	1.88%	1.49%	0.23%	84.33%	72.22%	13.05%	25.67%	11176	853
event_trafficjam	11.57%	24.31%	0.68%	0.55%	31.80%	19.44%	55.95%	55.70%	17548	44836
Total TomTom Event Report	**7.52%**	**23.48%**	**1.00%**	**0.58%**	**52.04%**	**20.88%**	**39.44%**	**55.06%**	**29221**	**46977**
Tweet Event Report										
event_accident	0.00%	6.38%	6.06%	5.67%	3.03%	7.80%	90.91%	80.14%	33	141
event_closure	0.00%	5.30%	2.63%	9.85%	7.89%	0.76%	89.47%	84.09%	38	132
event_enforcement	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	100.00%	2	14
event_hazard_animal	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	100.00%	3	6
event_hazard_object	0.00%	7.69%	0.00%	0.00%	0.00%	0.00%	100.00%	92.31%	3	13
event_hazard_road_condition	9.09%	7.41%	13.64%	7.41%	4.55%	0.00%	72.73%	85.19%	22	27
event_hazard_stopped_vehicle	0.00%	0.00%	0.00%	0.00%	16.67%	11.76%	83.33%	88.24%	6	17
event_hazard_violation	0.00%	0.00%	0.00%	10.00%	0.00%	0.00%	100.00%	90.00%	2	10
event_hazard_weather	12.70%	2.53%	9.52%	25.32%	4.76%	0.00%	73.02%	72.15%	63	79
event_trafficjam	13.33%	15.96%	6.67%	2.13%	20.00%	17.02%	60.00%	64.89%	15	94
Total Tweet Event Report	**6.42%**	**6.75%**	**6.95%**	**8.63%**	**6.42%**	**5.63%**	**80.21%**	**78.99%**	**187**	**533**
Waze Event Report										
event_accident	4.85%	6.51%	1.02%	0.82%	3.95%	5.01%	90.18%	87.66%	886	2917
event_closure	0.00%	0.00%	0.16%	0.74%	0.00%	0.11%	99.84%	99.15%	632	1759
event_enforcement	5.23%	5.75%	0.00%	0.00%	0.00%	0.00%	94.77%	94.25%	918	4766
event_hazard_animal	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	100.00%	45	219
event_hazard_object	0.44%	1.83%	0.00%	0.33%	2.20%	2.16%	97.36%	95.67%	227	1202
event_hazard_road_condition	16.67%	3.68%	2.00%	0.00%	0.00%	0.00%	81.33%	96.32%	150	679
event_hazard_roadwork	0.00%	0.00%	0.00%	0.00%	0.10%	0.27%	99.90%	99.73%	1002	3737
event_hazard_stopped_vehicle	3.52%	5.28%	0.05%	0.05%	0.21%	0.19%	96.22%	94.48%	8105	31619
event_hazard_traffic_sign	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	100.00%	30	195
event_hazard_weather	3.80%	3.62%	2.74%	0.28%	18.83%	0.49%	74.64%	95.61%	1790	2458
event_trafficjam	16.44%	19.11%	1.50%	1.23%	29.54%	24.44%	52.52%	55.22%	26170	92117
Total Waze Event Report	**11.94%**	**14.03%**	**1.15%**	**0.84%**	**20.34%**	**16.07%**	**66.57%**	**69.06%**	**48846**	**141668**

Table 4-26: Subreports linked to Main Reports by Category where P1 equals the period from 05-12-2017 to 06-01-2018 and P2 equals the period from 07-01-2018 to 17-02-2018 (only including the dates where data could be collected for all sources)

In Section 3.7, a method for computing the relevance of a traffic event reports was explained. In short, this included that the relevance of a traffic event report is computed as follows: we divide the area of the intersection of the locations from all traffic event reports by the area of a traffic event report location. In other words, in order to compute the relevance of a subreport to its "subReportIntersectedLocation", which is the intersection between a subreport and the main report, we have to divide the intersection of the subreport with its main report by the subreport resulting in the "subReportRelevanceSub". The same way we can calculate the relevance of the main report towards that intersection by dividing the intersection of the subreport and the main report by the main report resulting in the "mainReportRelevanceSub". However, if there is more than one subreport of the same event category linked to a main report, then we can also calculate the relevance of the subreport to all subreports with the same category by intersecting all subreports and diving it by the subreport, which we call "subReportRelevanceTotal". In the same way we can compute the relevance of a cluster of subreports related to one category towards the main report, resulting in the "subReportsMainReportRelevance". For further information regarding the terminology, please refer back to Table 4-24. Now that we have further explained our definition of relevance, we can take a look at the distribution of the "subReportRelevanceSub" and "mainReportRelevanceSub" as illustrated in Figure 4-11. Note how a subreport is often only for a small percentage (0-10%) relevant towards a subcluster intersection, while a main report is often relevant for a large percentage (90-100%) towards a subcluster. When looking at the relevance of subreports towards a main reports, as depicted in Figure 4-12, we see that the majority of the subreports have less than 20% relevance towards the main report.

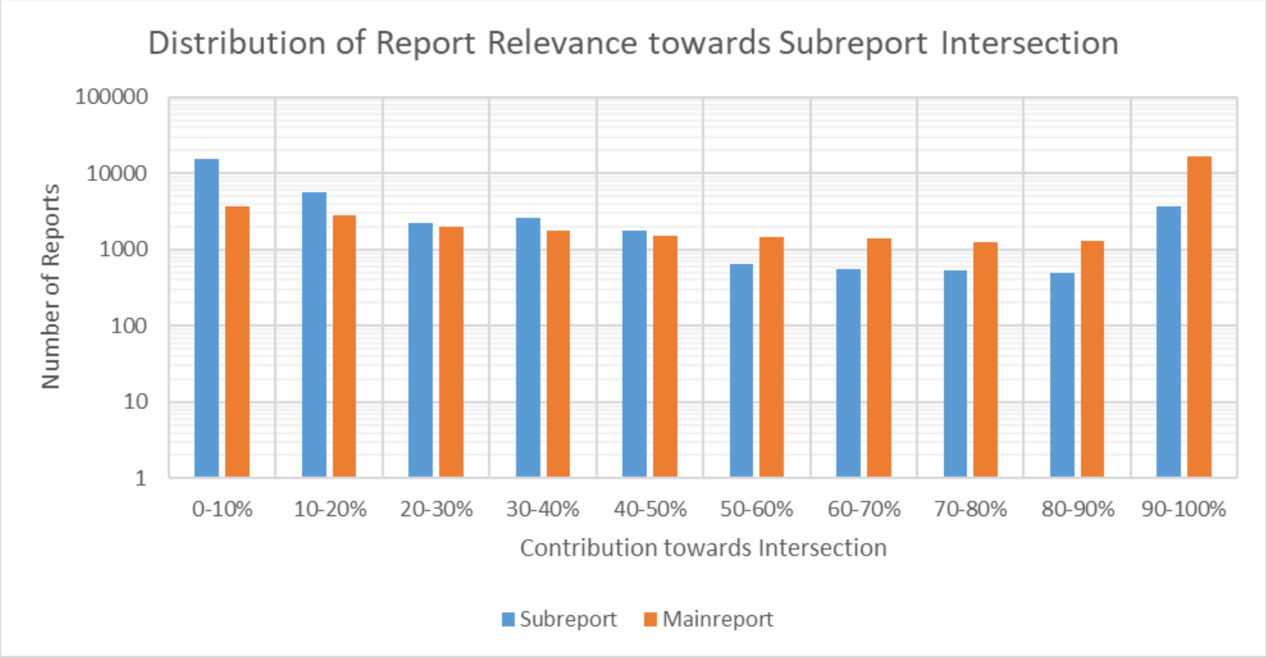


Figure 4-11: Distribution of Report Relevance towards Subreport Intersection

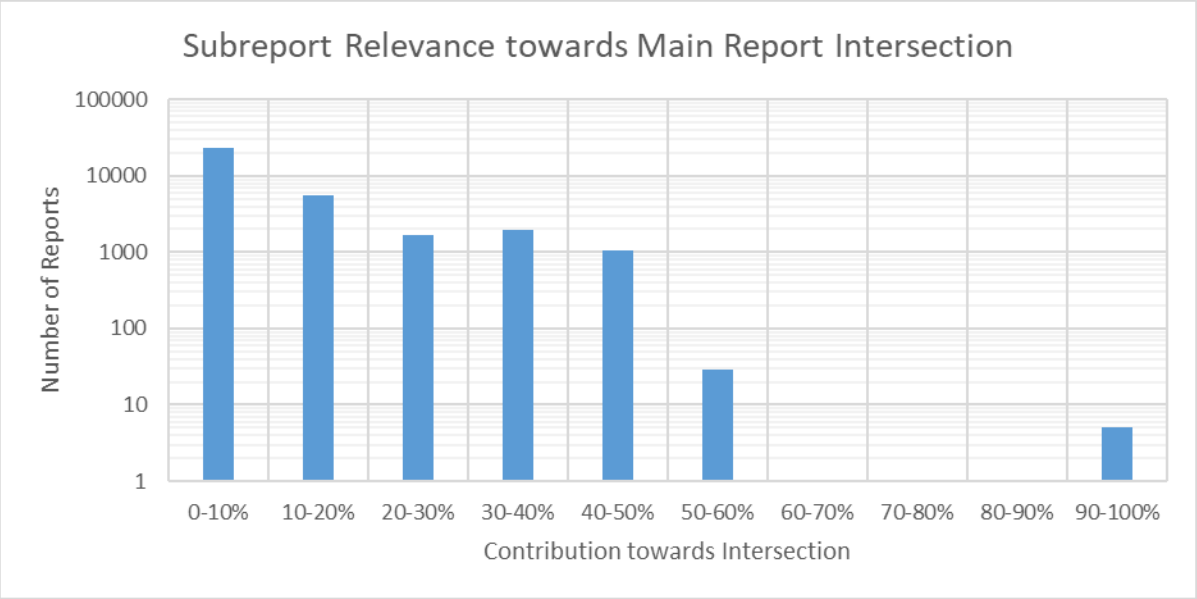


Figure 4-12: Subreport Relevance towards Main Report Intersection

Next, we linked all traffic event report clusters that consisted of more than one traffic event report where possible to the DiTTLab data. As previously explained in Section 3.7, for DiTTLab data to be related to a traffic event, its motorway geolines have to intersect the location of the traffic event. If this is the case, the traffic data for the intersected segment plus two 100m road segments before and after the segment, are linked to the event. Additionally, we extend the time interval as defined in the traffic event cluster rule (traffic event category, radius/dilation, timespan), with an additional 15 minutes before the event start. Our initial linking is done based on a direct intersection without the two additional 100m road segments. Another requirement that we added is that the "subReportsIntersectedLocationArea" must be greater than 0 km^2 and smaller than 20 km^2. This way, over the period from 05-12-2017 to 06-01-2018, we were able to link 24,072 out of the 51,111 subcluster locations (47.10%). When we compensate the data for dates no data could be collected we were able to link 15,128 out of the 31,089 subcluster locations (48,66%). Table 4-27, provides an overview of the categories by report type that have been linked to DiTTLab data.

	Number of Linked DiTTLab Data by Report							
Subcluster Categories by Report	**waze_report**		**tweet_report**		**tomtom_report**		**End Total**	
TomTom Event Report	**Unfiltered**	**Filtered**	**Unfiltered**	**Filtered**	**Unfiltered**	**Filtered**	**Unfiltered**	**Filtered**
event_accident	13.64%	18.75%	6.82%	9.38%	79.55%	71.88%	44	32
event_closure	5.26%	5.26%	0.00%	0.00%	94.74%	94.74%	19	19
event_hazard_object	25.00%	25.00%	0.00%	0.00%	75.00%	75.00%	8	4
event_hazard_road_condition	0.00%	0.00%	33.33%	33.33%	66.67%	66.67%	3	3
event_hazard_roadwork	9.52%	6.78%	1.59%	1.69%	88.89%	91.53%	63	59
event_hazard_stopped_vehicle	12.12%	6.67%	0.00%	0.00%	87.88%	93.33%	33	30
event_hazard_weather	1.13%	1.15%	2.28%	2.25%	96.59%	96.60%	3902	3823
event_trafficjam	21.52%	19.62%	2.07%	2.10%	76.42%	78.28%	3388	2946
Total TomTom Event Report	**10.62%**	**9.20%**	**2.20%**	**2.21%**	**87.18%**	**88.59%**	**7460**	**6916**
Tweet Event Report								
event_accident	47.06%	0.00%	41.18%	66.67%	11.76%	33.33%	17	5
event_closure	30.00%	0.00%	50.00%	0.00%	20.00%	100.00%	10	2
event_enforcement	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	0
event_hazard_object	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	0
event_hazard_road_condition	57.14%	40.00%	28.57%	40.00%	14.29%	20.00%	7	5
event_hazard_stopped_vehicle	33.33%	0.00%	33.33%	0.00%	33.33%	100.00%	3	1
event_hazard_weather	41.67%	50.00%	33.33%	50.00%	25.00%	0.00%	12	4
event_trafficjam	62.50%	25.00%	12.50%	25.00%	25.00%	50.00%	8	4
Total Tweet Event Report	**47.46%**	**26.32%**	**33.90%**	**36.84%**	**18.64%**	**36.84%**	**59**	**19**
Waze Event Report								
event_accident	66.29%	40.30%	9.14%	7.46%	24.57%	52.24%	175	67
event_closure	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%	3	0
event_enforcement	98.46%	100.00%	1.54%	100.00%	0.00%	0.00%	65	8
event_hazard_animal	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2	0
event_hazard_object	66.67%	20.00%	0.00%	0.00%	33.33%	80.00%	12	5
event_hazard_road_condition	62.50%	83.33%	37.50%	16.67%	0.00%	0.00%	32	12
event_hazard_stopped_vehicle	97.51%	91.80%	0.86%	1.23%	1.63%	6.97%	1163	244
event_hazard_weather	21.62%	12.35%	20.74%	12.65%	57.64%	75.00%	458	340
event_trafficjam	49.77%	24.16%	5.07%	3.86%	45.15%	71.98%	14643	7517
Total Waze Event Report	**52.75%**	**25.97%**	**5.32%**	**4.19%**	**41.94%**	**69.84%**	**16553**	**8193**

Table 4-27: Traffic Event Clusters linked to DiTTLab data

4.5.2 Traffic Event Description Insights

In Table 4-26, in the previous section we could see that for period 1, 44.82% of the traffic event clusters consists of more than one event report. For period 2, this percentage comes down to 34.39%. But what insights can be gained from these clusters that contained more than one event report? First, we look at how tweet event reports contribute to Waze and TomTom event reports. Table 4-26, shows that in period 1 only 1.00% of the tweets get linked to a TomTom main report, and 0.58% in period 2. In the case of Waze event reports, these numbers come down to 1.15% and 0.84%. Additionally, this table shows that tweet reports contribute mostly in accident, weather, and traffic jam categories. Second, we look at how Waze event reports contribute towards tweet and TomTom event reports. Table 4-26 shows that in period 1 7.52% of the clusters with a TomTom event as main report contain a Waze report, and 23.48% in period 2. Waze reports seem to contribute most towards TomTom events in the categories accident, roadwork, and traffic jams. For clusters with a tweet as main report the percentage of linked Waze reports sits on 6.42% for period 1 and 6.75% for period 2. In this case, Waze reports clearly contribute most in the traffic jam category. These numbers are lower than one would expect based on the fact that Waze is a social platform specifically specialized in traffic event reporting. Third, we look into the clustering of TomTom event reports towards Waze and tweet event reports. Table 4-26 shows that 20.34% of the clusters with a Waze event as main report, contain a TomTom report in period 1, compared with 16.07% in period 2. Moreover, this table shows that 6.42% of the clusters with a tweet event as main report, contain a TomTom report in period 1, compared with 5.63% in period 2. We see that a relatively high percentage of TomTom events cluster with Waze event reports, however, this is mostly based on the weather and traffic jam categories. Besides linking to traffic events from other data sources, traffic events can also form a cluster with events from the same data source. These percentages are significantly higher than the previously found percentages. 52.04% of TomTom subreports cluster with a TomTom main report in period 1, and 20.88% in period 2. 6.95% in period 1 for tweet subreports with tweet main reports, and 8.63% in period 2. And 11.94% for Waze subreports with Waze main reports in period 1, compared to 14.03% in period 2.

We showed what the effect of clustering event reports is on each event report type and their event categories. In Figure 4-6, Figure 4-8 and Figure 4-9, we also showed the average category distribution over the period of a day. This gives a clear view of the weaknesses and strengths of the different geosocial sources given their abilities to describe event categories around the day. We have also discussed the strengths and weaknesses of each of the geosocial sources given the type of information they provide. Tweet reports are able to provide a lot of context as they contain more elaborate descriptions and often contain some sort of media. However, this comes at the cost of precision regarding their categorical, locational and temporal descriptiveness. This is where Waze and TomTom events contribute most, as they contain exact categorical, locational and temporal features. However, their ability to provide context is mostly limited to predefined categorical descriptions.

Beside coverage of time and type of information, there is one last type of coverage that we have not yet discussed, which regards the coverage of space. In order to show the distribution of the different geosocial sources and their event categories we used our SocialTerraffic application to map the event reports. We selected the date 15-01-2018, as this date contains an average representation of Waze, TomTom, and tweet event reports. We decided to take a closer look at one very common category namely event traffic jam and one less common category namely event accident. This in order to clearly show the different coverage of space between geosocial sources and categories. Figure 4-15, depicts the mapping of 218 Waze event reports of the accident category. Note how the reports appear more frequently around the larger cities (e.g., Amsterdam, Rotterdam) and motorways (e.g., A2, A20). Figure 4-14, depicts the mapping of 34 TomTom event reports of the accident category. This number is significantly less than the number of Waze reports, however, a single TomTom report covers a larger location compared to a single Waze report. Figure 4-13, depicts the mapping of 12 tweet event reports of the accident category. Analyzing tweet reports based on location is less reliable, as their geolocation is derived from the tweet text. However, it again seems that the reports appear more often around the larger cities and motorways. Next, we look at the clustered events based on the event reports. Figure 4-18, shows 208 events based on a main report, i.e., event reports that could not be clustered to other event reports by our algorithm. Figure 4-16, depicts the locations of 29 events based on a main report and a single subreport. This can occur when only one other event report is clustered towards the main report. However, this also occurs when multiple subreports are clustered to the main report, but not all subreports intersect with each other. Figure 4-17, contains the locations of 27 events based on a main report and multiple subreports, i.e., the main report was clustered with multiple subreports and these subreports all intersect with each other. It becomes clear that many events are dropped due to the clustering process, while the event locations tend to move even closer to the larger cities and motorways.

Next, we look at the events of the traffic jam category. As the amount of reports on traffic jams is significantly higher than reports on accidents, the timeframe was reduced to the morning rush hour period, ranging from 06:30 to 09:30 on 15-01-2018. Figure 4-21 depicts the locations of 4045 Waze event reports and confirms our previous findings concerning the concentration of report locations. The same is true for the 1107 TomTom event reports depicted in Figure 4-20. The reports clearly concentrate around the motorways and larger cities, whereas the provinces such as Zeeland, Friesland, Groningen, and Drenthe are only sparsely covered. Additionally, again the number of tweets is significantly lower with 9 event reports, yet concentrated around the same locations as the reports from TomTom and Waze, as shown in Figure 4-19. When clustering the event reports as shown in Figure 4-23, Figure 4-22, and Figure 4-24, the clustering process causes many events to get dropped and event locations to concentrate towards the motorways and large cities as seen before.

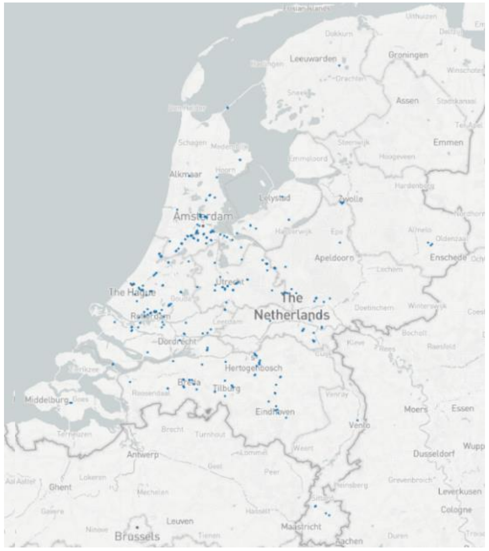


Figure 4-15: Waze Event Accident (218 event reports)

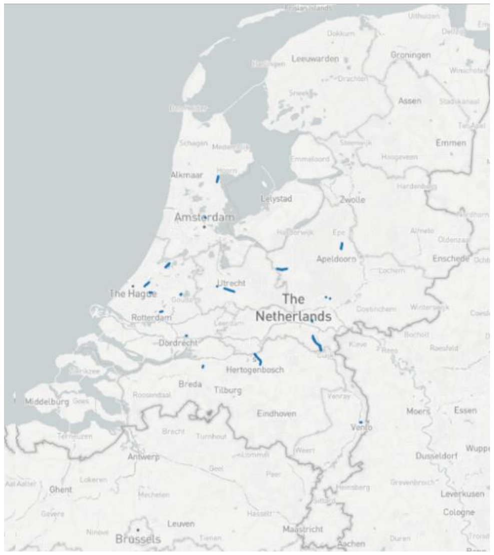


Figure 4-14: TomTom Event Accident (34 event reports)

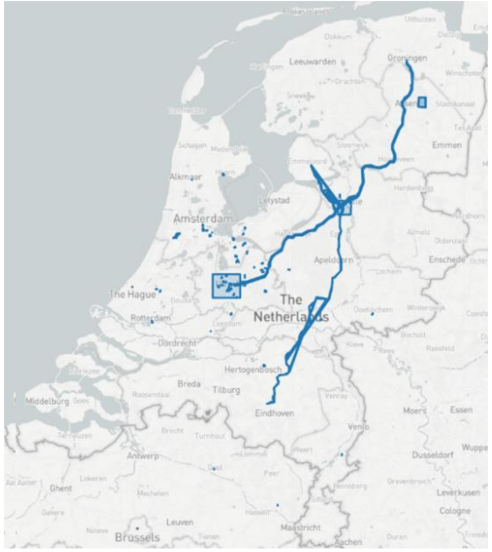


Figure 4-13: Tweet Event Accident (12 event reports)

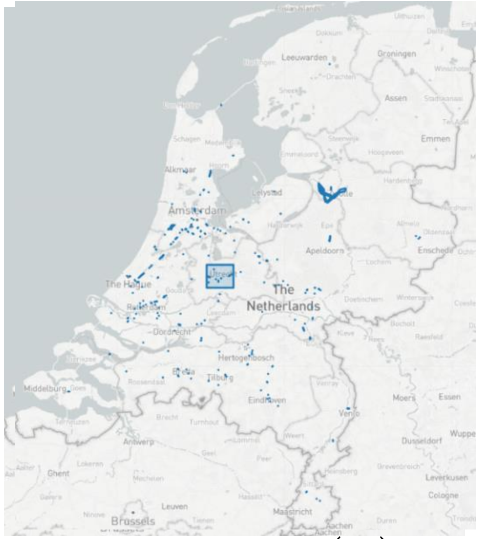


Figure 4-18: Accident Events (208) based on a single main report

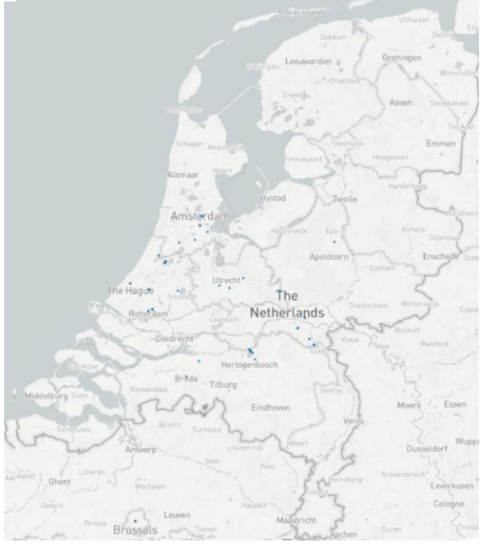


Figure 4-16: Accident Events (29) based on a single main- and subreport

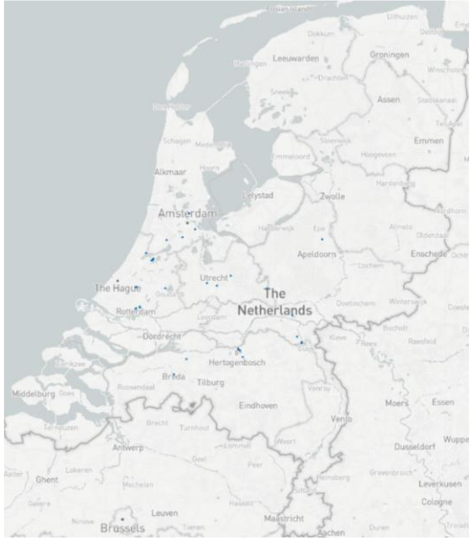


Figure 4-17: Accident Events (27) based on a single main- and multiple subreports

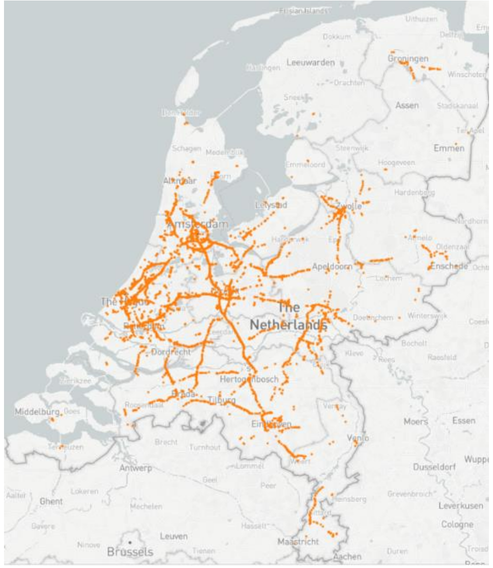


Figure 4-21: Waze Event Traffic Jam (4045 event reports)

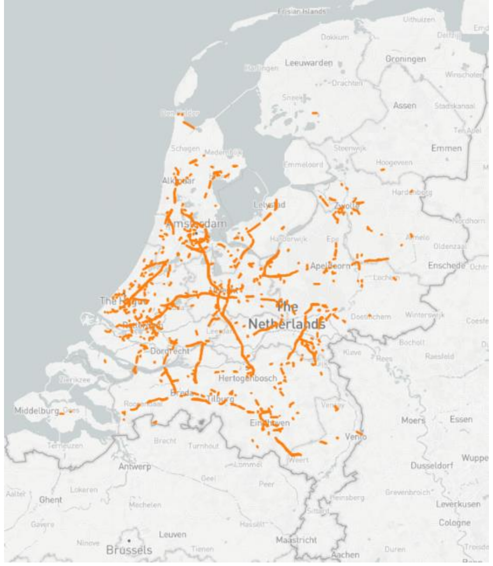


Figure 4-20: TomTom Event Traffic Jam (1107 event reports)

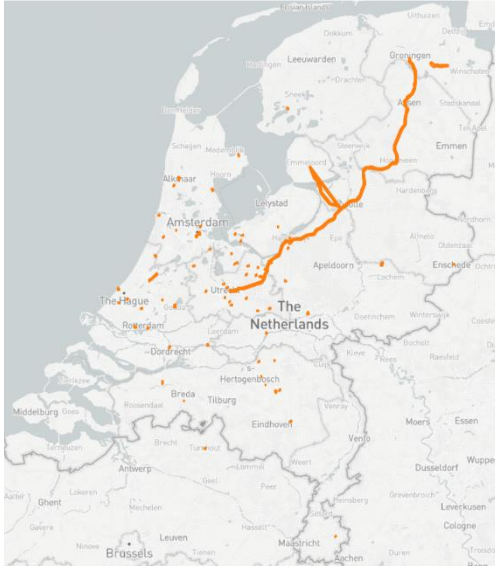


Figure 4-19: Tweet Event Traffic Jam (9 event reports)

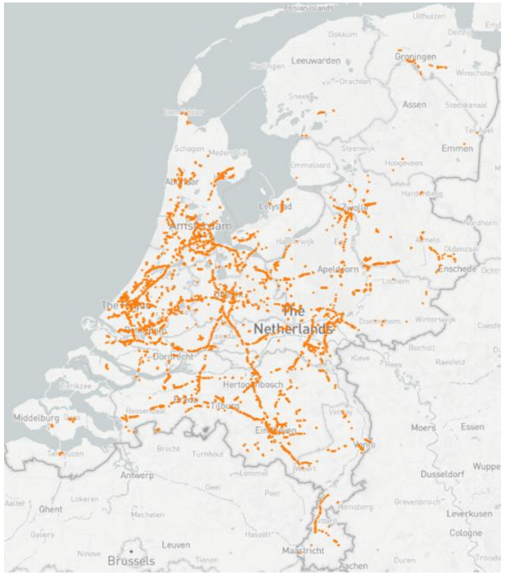


Figure 4-23: Traffic Jam Events (2257) based on a single main report

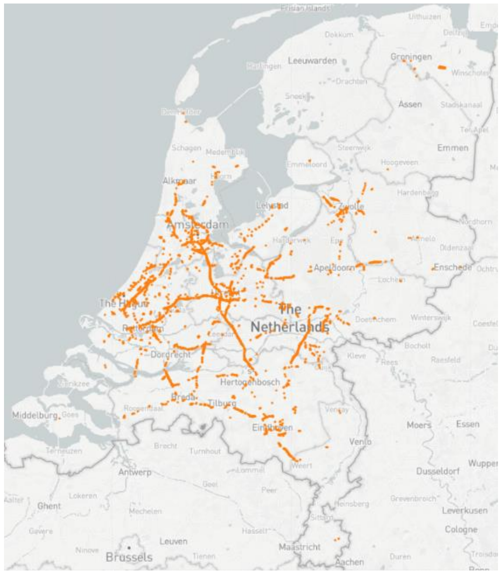


Figure 4-22: Traffic Jam Events (2081) based on a single main- and subreport

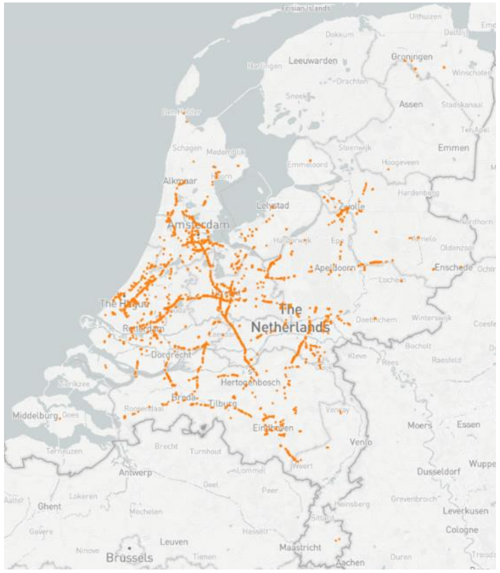


Figure 4-24: Traffic Jam Events (1685) based on a single main- and multiple subreports

4.6 SocialTerraffic System

In order to create the web-based interactive map application SocialTerraffic, a number of tools were used. First, Node.js[47] (v. 7.1.0) an asynchronous event-driven JavaScript runtime environment was used to write our application. Second, we used Express[48] (v. 4.13.1), a Node.js web application framework which includes a set of features to develop the web application. Third, we used the MongoDB Node.js Driver (v. 3.1) to let our application communicate with our database. Last, a number of open-source JavaScript libraries of which we listed the most important ones:

- Leaflet[49] (v. 1.0.3), for creating interactive maps.
- Jquery[50] (v. 3.3.1), for HTML document traversal and Ajax.
- Vis[51] (v. 3.12.0), for creating interactive timelines.
- Moment[52] (v. 2.22.2), for parsing, validating, manipulating and displaying date objects.
- Plotly[53] (v. 1.0.6), for creating charts.

Figure 4-25, depicts the start screen of the application containing the following four main sections:

1. The map of the Netherlands which is prominently centered in the middle.
2. The control panel on the left side, which allows for switching between map styles, traffic event categories, and showing additional event reports.
3. The menu sidebar on the right side, used for exploring traffic events, requesting traffic event information, and displaying traffic data based graphs.
4. The timeline on the bottom, displaying the number of active events by time and category.

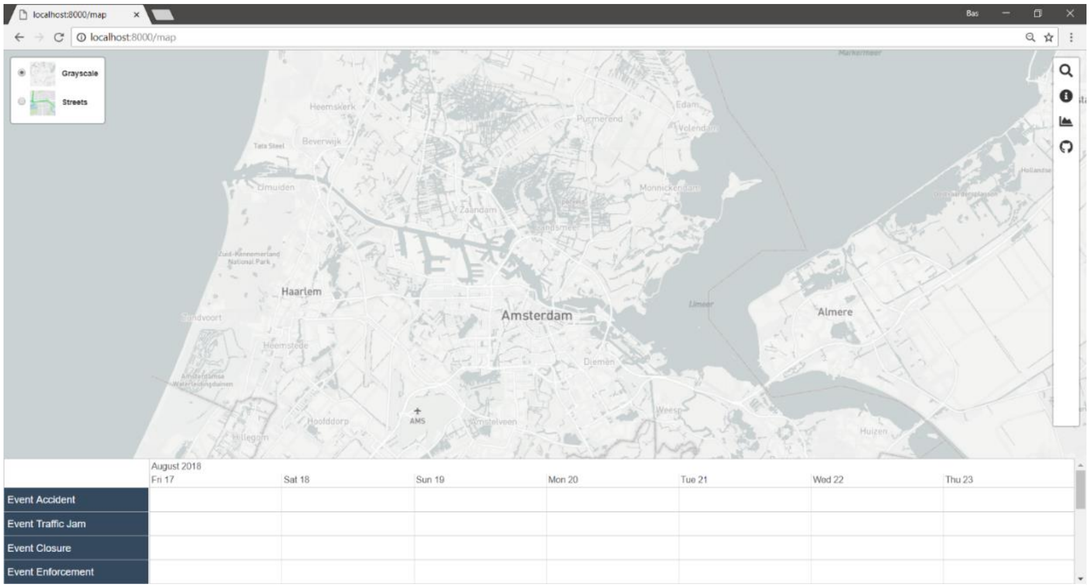


Figure 4-25: SocialTerrafic Application Start Screen

[47] https://nodejs.org/en/

[48] https://expressjs.com/

[49] https://leafletjs.com/

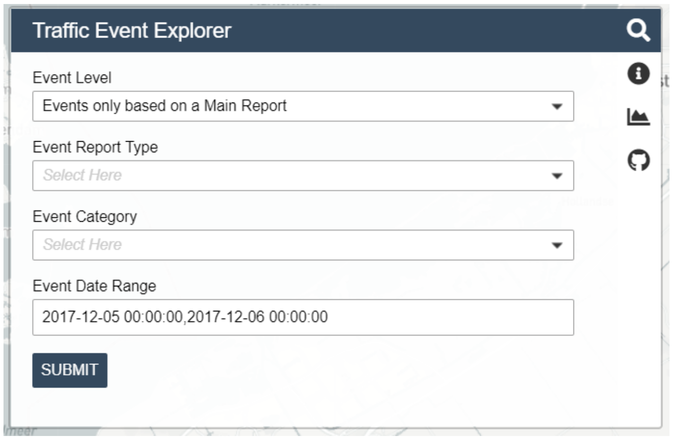
[50] https://jquery.com/

[51] http://visjs.org/

[52] https://momentjs.com/

[53] https://plot.ly/javascript/

By clicking on the search icon in the menu sidebar, the sidebar will unfold the Event Explorer submenu, as illustrated in Figure 4-26. In this menu, a user is able to select one of three event levels. The first event level contains only events based on a main report. The second event level contains events based on a main report and a single subreport. The third event level contains events based on a main report and multiple subreports. The second selection a user has to make is based on which report types he wants have included in the traffic events. A user is able to select and combine tweet, Waze, and TomTom reports. The third selection is based on the event category of the traffic event. A user can choose one or multiple categories from a predefined list of 13 different event categories. Lastly, the user has to select a date range for the traffic events. By pressing the submit button, a query is generated based on the selection and the data is retrieved from the database. As an example, we searched for events based on a main report and multiple subreports, all report types, all event categories, for the period of 08-12-2017 06:30 to 08-12-2017 09:30. Figure 4-27, shows the result of this query. Notice how the event location shapes are drawn on the map based on their geographic data structure (GeoJSON), each with its own distinctive category color. The control panel on the left, as well as the timeline on the bottom, get automatically updated with the related event types painted in the same color.



Traffic Event Explorer

Event Level: Events only based on a Main Report

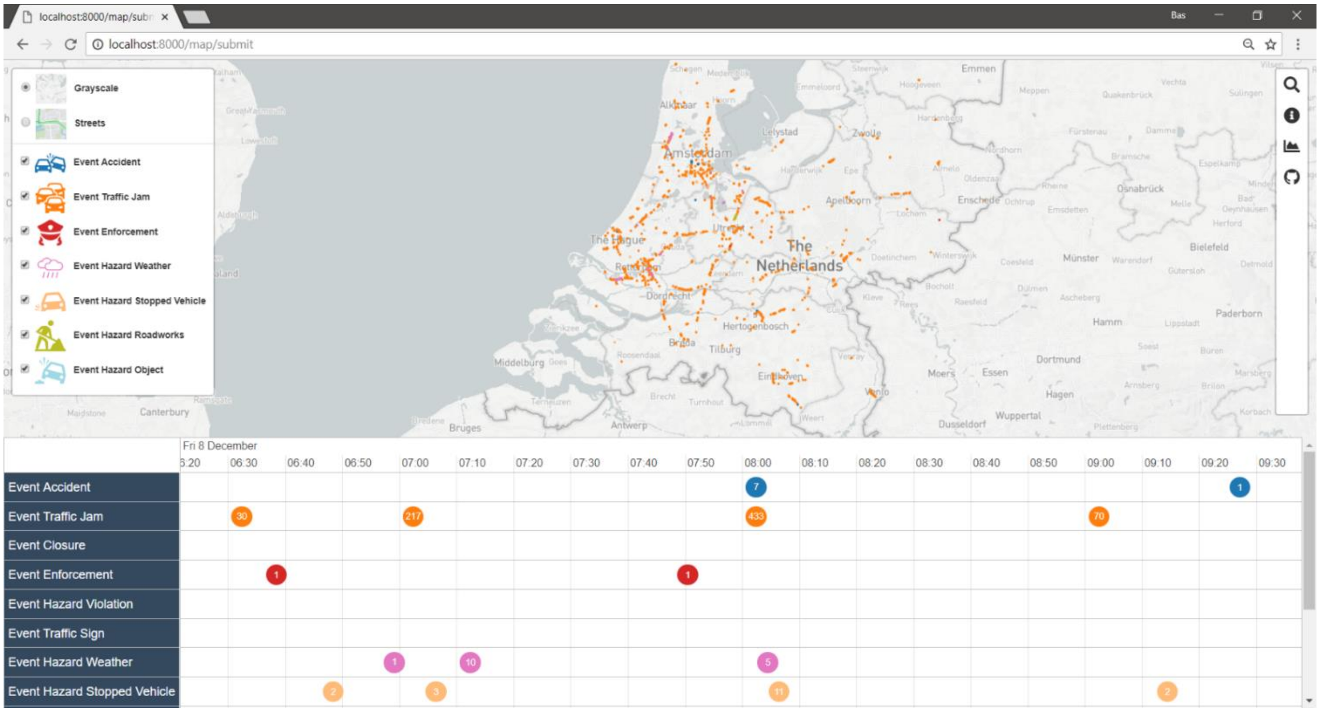
Event Report Type: Select Here

Event Category: Select Here

Event Date Range: 2017-12-05 00:00:00,2017-12-06 00:00:00

SUBMIT

Figure 4-26: SocialTerraffic Menu Sidebar - Traffic Event Explorer



localhost:8000/map/submit

Figure 4-27: SocialTerraffic screen after submitting query

After the event data is loaded into the application a user might want to filter out traffic events based on their category. The control panel on the left side enables a user to do this by simply unchecking an event box. This hides the events on the map and as well on the timeline. Figure 4-28 shows what it looks like when we uncheck the Event Traffic Jam box.

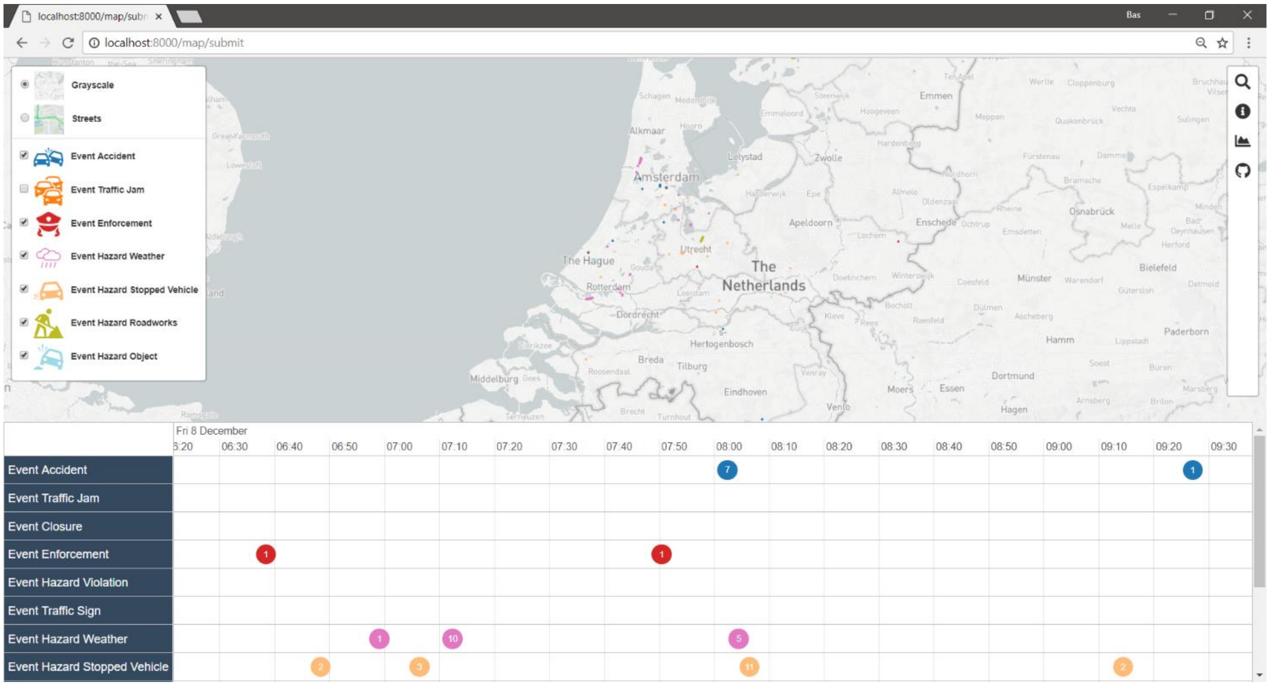
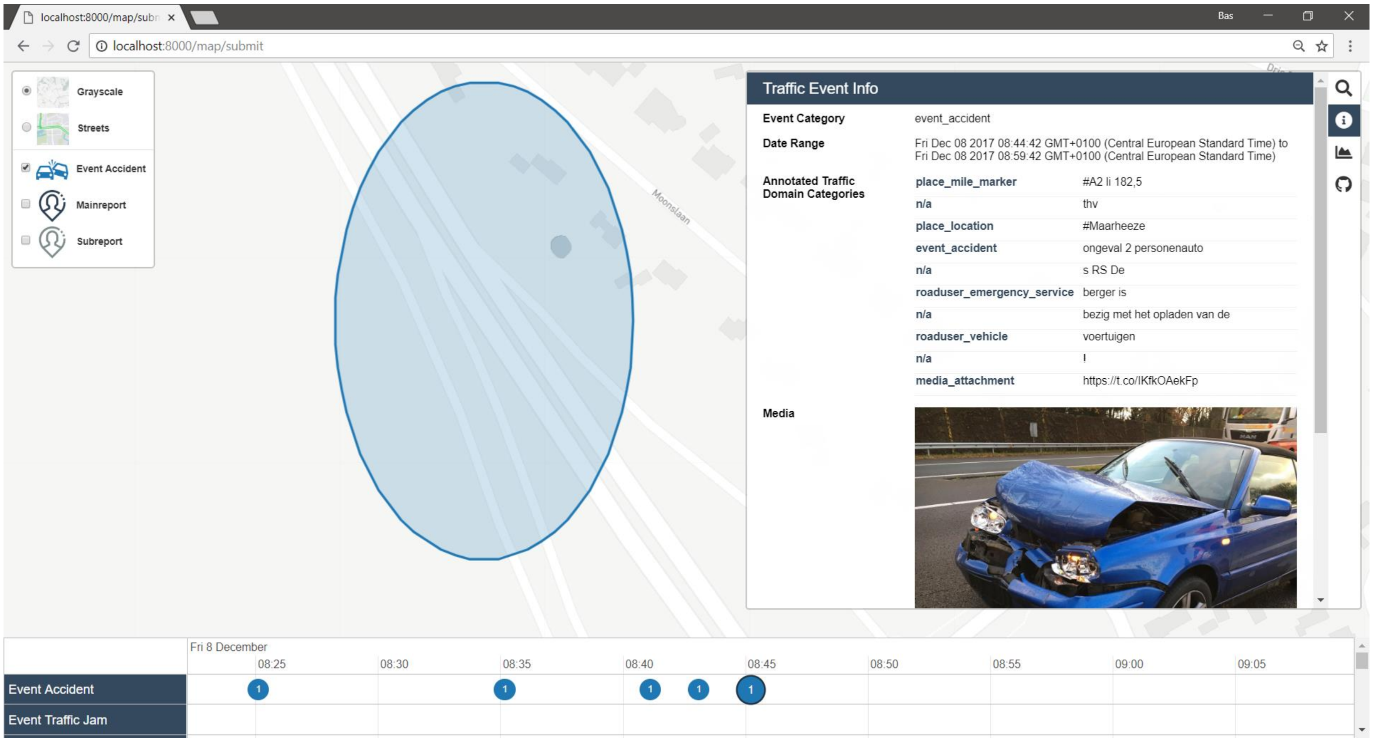


Figure 4-28: SocialTerraffic screen after unchecking the Event Traffic Jam box

Another special thing to mention is the clustering of events on the timeline. As many events can happen within the same small time frame, drawing each single event would overfill and reduce the responsiveness of the timeline. Therefore we implemented a clustering approach which clusters the events by time unit (day, hour, minute etc.) depending on the zoom level of the timeline. When the user zooms in on the timeline the elements get split. For example, Figure 4-28 shows a timeline where one item contains 7 accident events; Figure 4-29 shows that when zoomed in, the item splits into 7 separate items.

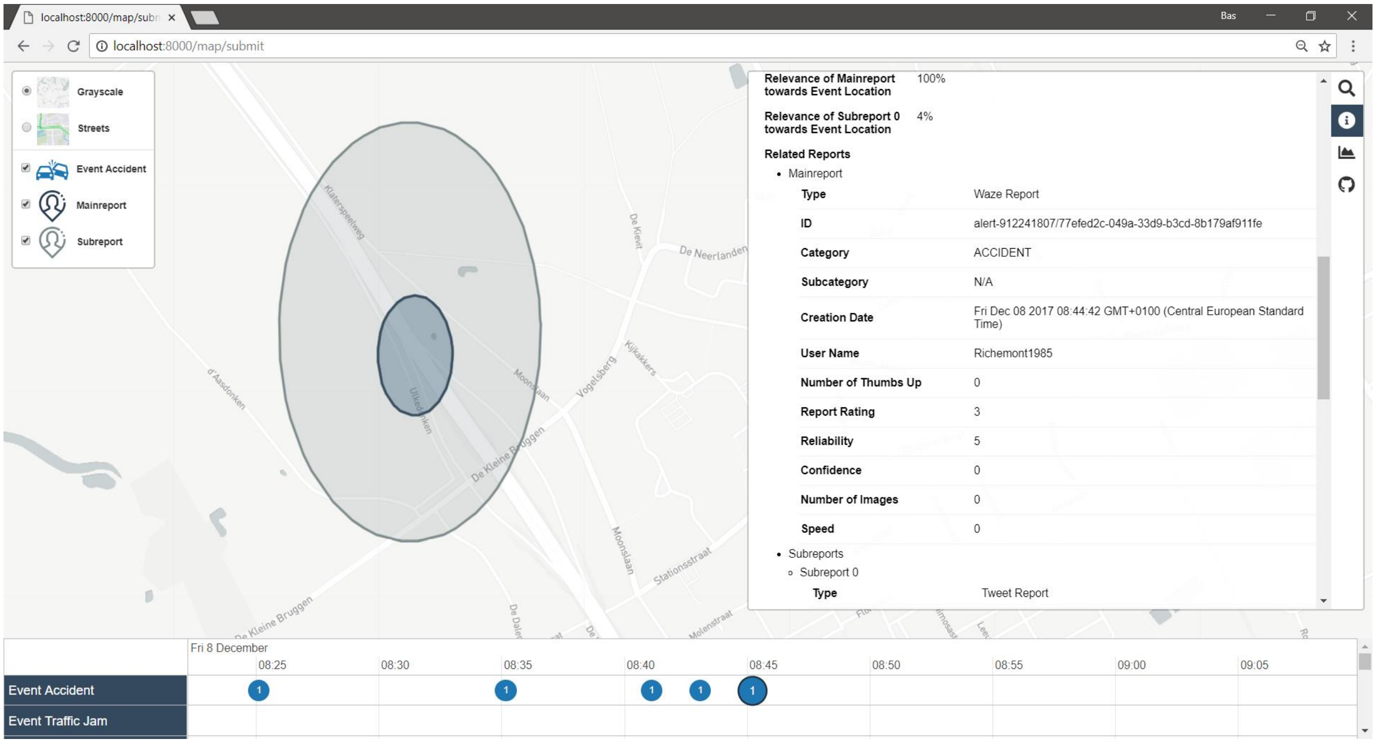
Besides providing a high-level overview of the traffic events, a user can get detailed information for each event. When a user either clicks on the event location on the map or on an event on the timeline, the menu sidebar unfolds the Traffic Event Info submenu. This submenu contains the event category, date range, annotated traffic domain categories (only applies if the cluster contains tweet reports), media, relevance percentage of the main report towards the event location, relevance of the subreports towards the event location, and the properties of each report that is part of the event cluster. Figure 4-29 depicts this screen after a user selects a traffic event. The Traffic Event Info submenu is visible at the right side (only the top half is shown, the bottom half is shown in Figure 4-30). Note that the related event item on the timeline automatically gets highlighted, and the control panel gets two additional filter boxes named Mainreport and Subreport. By checking these boxes the locations of the event reports the intersected location of the selected traffic event is based upon are drawn on the map, as shown in Figure 4-30. Keep in mind that the shown subreport relevance of 4% towards the event location is based on the locations that were dilated with 150 meters, as explained in Section 4.5. The event report locations on the map are the original not yet dilated locations, and therefore do not visually match the relevance percentages.



Traffic Event Info

Event Category	event_accident	
Date Range	Fri Dec 08 2017 08:44:42 GMT+0100 (Central European Standard Time) to Fri Dec 08 2017 08:59:42 GMT+0100 (Central European Standard Time)	
Annotated Traffic Domain Categories	**place_mile_marker**	#A2 li 182,5
	n/a	thv
	place_location	#Maarheeze
	event_accident	ongeval 2 personenauto
	n/a	s RS De
	roaduser_emergency_service	berger is
	n/a	bezig met het opladen van de
	roaduser_vehicle	voertuigen
	n/a	!
	media_attachment	https://t.co/IKfkOAekFp
Media		

Figure 4-29: SocialTerraffic screen after selecting a traffic event on the map or timeline



Relevance of Mainreport towards Event Location 100%

Relevance of Subreport 0 towards Event Location 4%

Related Reports

- Mainreport

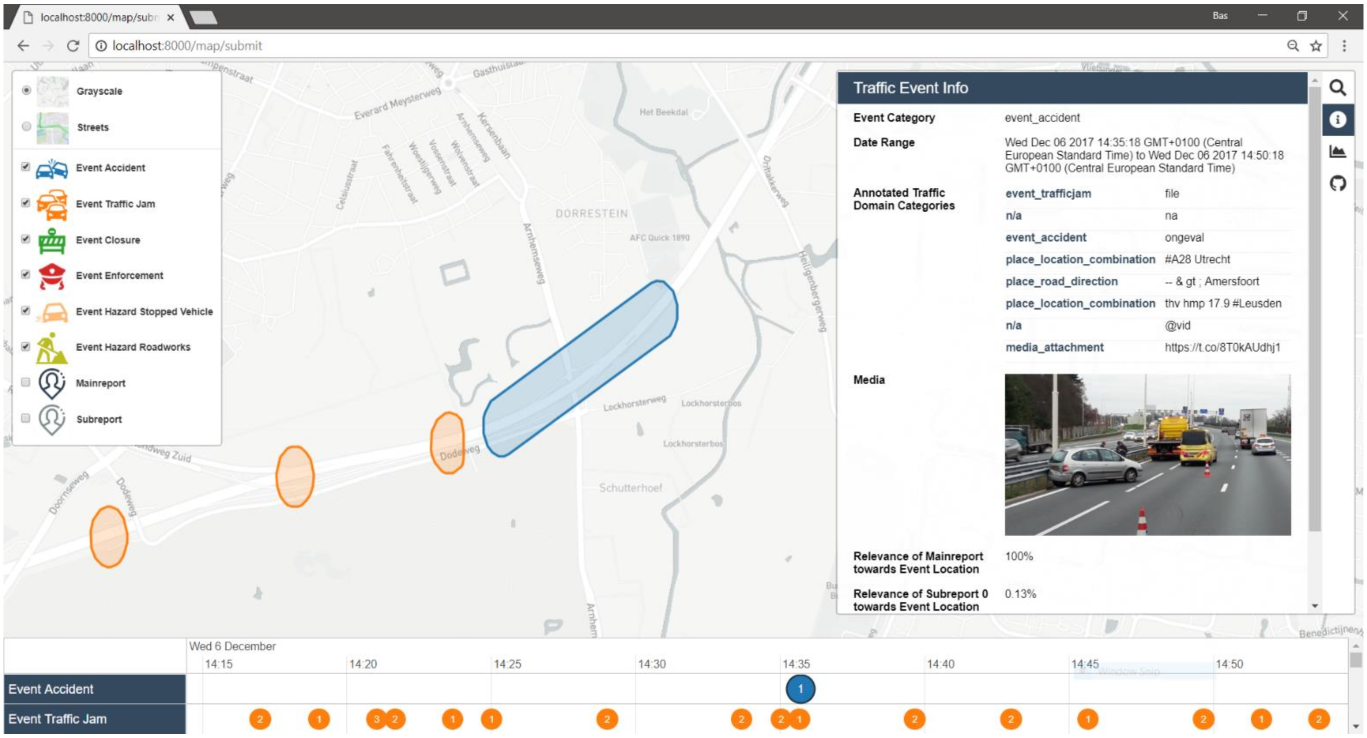
Type	Waze Report
ID	alert-912241807/77efed2c-049a-33d9-b3cd-8b179af911fe
Category	ACCIDENT
Subcategory	N/A
Creation Date	Fri Dec 08 2017 08:44:42 GMT+0100 (Central European Standard Time)
User Name	Richemont1985
Number of Thumbs Up	0
Report Rating	3
Reliability	5
Confidence	0
Number of Images	0
Speed	0

- Subreports
  - Subreport 0

Type	Tweet Report

Figure 4-30: SocialTerraffic screen after checking the Mainreport and Subreport boxes

Our application would not be complete without providing the option to see how the collected geosocial data enriches the traffic data. Therefore, when a user clicks on the chart icon on the menu sidebar it unfolds the DiTTLab Traffic Data submenu. In this menu, the user can select a time range, based on this time range and the location of the traffic event a query is generated that retrieves the DiTTLab speed and flow data for that location and time period. This traffic data gets charted in a similar way as is done within the DiTTLab NDW app[54]. This means that we create a heat map with the time on the x-axis, the distance on the y-axis, and speed/flow on the z-axis. Let us take a look at a new example to illustrate this feature. Figure 4-31, shows a detected traffic event with an accident category attached to it. It consists of a main report (TomTom) and one subreport (tweet). Also, notice how the system detects three traffic events of the traffic jam category around the same location and time period as the accident based traffic event.



Traffic Event Info

Event Category	event_accident	
Date Range	Wed Dec 06 2017 14:35:18 GMT+0100 (Central European Standard Time) to Wed Dec 06 2017 14:50:18 GMT+0100 (Central European Standard Time)	
Annotated Traffic Domain Categories	event_trafficjam	file
	n/a	na
	event_accident	ongeval
	place_location_combination	#A28 Utrecht
	place_road_direction	-- & gt ; Amersfoort
	place_location_combination	thv hmp 17.9 #Leusden
	n/a	@vid
	media_attachment	https://t.co/8T0kAUdhj1
Media		
Relevance of Mainreport towards Event Location	100%	
Relevance of Subreport 0 towards Event Location	0.13%	

Figure 4-31: SocialTerraffic screen with the focus on a traffic event of the accident category

Next, we open the DiTTLab Traffic Data submenu and select a time range starting an hour before the start of the event and ending an hour after the event. By selecting a larger time range a better overview can be gained of the speed/flow changes before and after the event. Additionally, we have to take into account that the accident described in the event could have happened in the period before the event was registered by geosocial data. The chart in Figure 4-32 shows how the traffic speed on the left side of the road remains constant and has normal values between 100 and 120 km/h. The same can be said for the traffic flow, as shown in Figure 4-33. However, when looking at the graph in Figure 4-34, the traffic speed is reduced by half between 14:10 and 14:50. This time period corresponds to the time period of the event the system registered. Additionally, the chart in Figure 4-35 shows that the traffic flow is also reduced by half in the same period. This shows how the event found by clustering geosocial data can help to enrich and explain anomalies found in traffic data.

[54] http://dittlab-apps.tudelft.nl:8080/app-ndw/

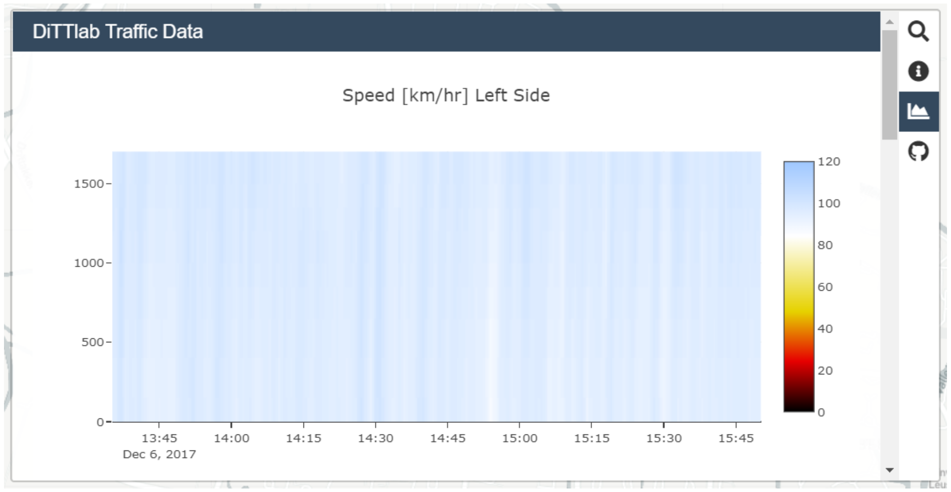


Figure 4-32: DiTTlab Traffic Data submenu, speed heat map for the left side of the road

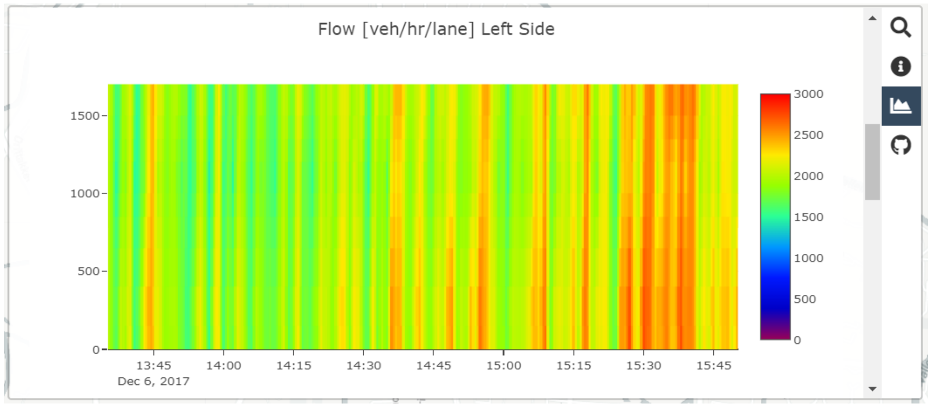


Figure 4-33: DiTTlab Traffic Data submenu, flow heat map for the left side of the road

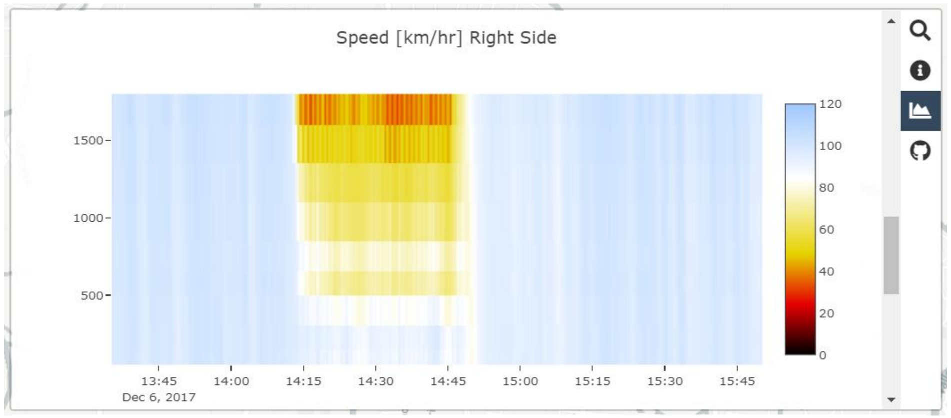


Figure 4-34: DiTTlab Traffic Data submenu, speed heat map for the right side of the road

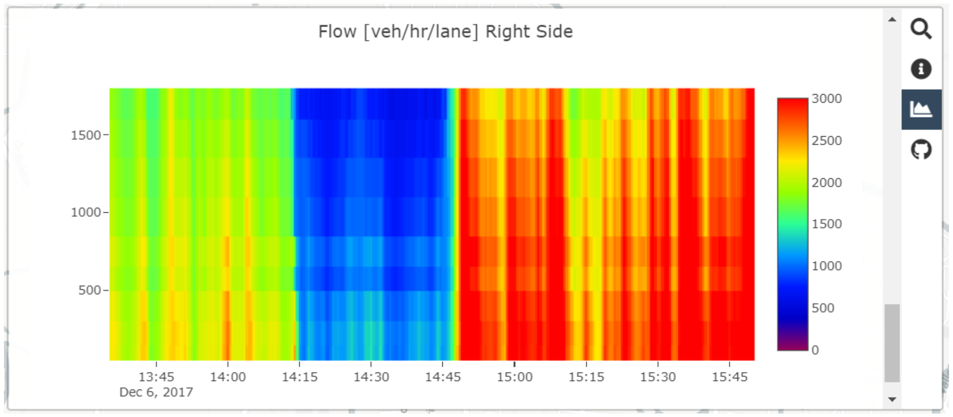


Figure 4-35: DiTTlab Traffic Data submenu, flow heat map for the right side of the road

5 Discussion

In this chapter, we discuss the results obtained in 4 regarding the design and implementation that lead towards the answering of our research questions.

5.1 Data Collection

As a starting point of our pipeline, we implemented a data collection approach to collect relevant tweets, Waze and TomTom reports. The main goal that we aimed to achieve when collecting traffic event-related tweets was to optimize the recall and precision as much as possible. This in order to mitigate the possible bias formed due to a limited keyword selection approach. By applying a custom keyword selection and filter process we were able to increase the collection of real road-user traffic event-related tweets by 80.54%, while reducing the number of non-real road-user traffic event-related tweets with 48.19%, relative to our initial set. Our proposed data collection approach provides an alternative to the limited keyword approach as seen in related work, e.g., from Wanichayapong et al. (2011), D'Andrea et al. (2015), and Nguyen et al. (2016). Besides, we have taken a different approach compared to all other related work by focusing only on the collection of tweets by so called real road-users. Whereas, the datasets in related work include a mixture of traffic event-related geosocial posts from real road-users, news agencies, bots, and emergency agencies. This affects the results, as traffic event-related geosocial posts from news agencies, bots and emergency agencies contain a different syntax than geosocial posts from real road-users. With the help of this keyword selection approach Twitter data has been collected over a period ranging from 05-12-2017 to 17-02-2018. The amount of the on average 873 collected tweets per day proved very sparse. Especially considering that on average only 6.71% of these tweets are deemed traffic event-related. We consider two possible causes for this limited amount of collected traffic event-related tweets. One the one hand, we have to consider that the number of traffic event-related tweets posted by real road-users is limited by default; as it is forbidden by law to use your phone while driving in the Netherlands. A user is only legally allowed to use its phone while inside a vehicle when the vehicle is stationary, e.g., in a traffic jam, in front of a traffic light or on a parking, or when the user is a passenger. Additionally, the rise of other social platforms could have caused a reduction in the amount of traffic event-related information that users post on Twitter. On the other hand, we have to take into account some possible limitations and weaknesses regarding our tweet collection approach. First, we applied the keyword selection approach on a limited time period of three days. This could have caused a bias towards our keyword set due to the found types of traffic events in that timespan. In other words, there is a possibility that we missed out on some relevant keywords or included keywords that only prove relevant for that limited time period. Second, we used a keyword filter based on the Dutch language, therefore capturing also Belgium or in exceptional cases even South African tweets. Third, bot accounts are filtered both manually and on “terms”, therefore newly created bot accounts that do not contain these “terms” are still included.

Next, we have extended upon related work by collecting data from two additional geosocial data sources: Waze and TomTom. We collected data from Waze over a period ranging from 06-12-2017 to 05-02-2018, resulting in an average amount of 7482 Waze event reports per day. Compared to tweets, Waze event reports are specialized towards the categorization of traffic events based on a specified event category. This means that they are more limited in their descriptiveness of traffic events compared to tweets. We found that only 0.06% of all Waze event reports contained one or multiple images, and 2.37% of all Waze event reports contained a user created description. Additionally, we want to state that our Waze collection approach was based on a live map feed and not an official API. Waze changed their policies in February 2018, causing the feed to no longer work. No alternatives have been presented by Waze to this time. Waze does, however, have a collaboration program called the Connected Citizens Program[55], which could present an alternative way for Waze data collection. Note that application attempts for this program, during this thesis, were turned down by Waze, causing us to create our own Waze collection approach.

Furthermore, TomTom data was collected over the period from 05-12-2017 to 14-02-2018. Leaving out the days we were not able to collect TomTom data, an average of 2543 TomTom event reports was collected per day. Compared to tweets and Waze event reports, TomTom reports contain only descriptions and causes selected from a fixed set of categories. However, as compensation, this set is more extensive than the category set that Waze provides. One of the weaknesses in our TomTom event collection approach, is that due to some technical issues regarding the TomTom Traffic Incident API there were some days that we were not able to receive TomTom event reports for the complete day or did not receive any reports at all. In this research we have tried to account for this problem were possible. Additionally, we cannot say with absolute certainty that all data from TomTom is real road-user based, which therefore could have caused some bias in some of the experiments.

Overall we can say that all three data sources combined contain enough descriptive possibilities to provide a complete picture of a traffic event. However, we find that in the case of Twitter data the collected amounts are too sparse to be of any contribution towards the descriptiveness of traffic events on its own. In most cases, related work showed that Twitter data on its own is suitable to describe traffic events, however as stated before these works do not distinguish real road-users from other users. Therefore, we cannot make a one to one comparison, based on the collected data, to these works.

5.2 Rule-based Traffic Domain Annotator

In order to extract relevant traffic domain information from the collected tweet text data, we created a rule-based traffic domain annotator. An evaluation on this annotator provided us with an average accuracy of 0.964 over 30 distinct categories, an average precision of 0.970, an average recall of 0.828, resulting in an average f1-score of 0.874. We want to note however, that due to the unbalanced frequency of the categories this score is not entirely representative.

No related work has been found that contains such an extensive annotator that is able to categorize tweets by traffic events, so no comparison could be made. There are, however, some downsides to an annotator that has to be able to differentiate that many categories. First, this annotator is only able to map token sets based on predefined grammatical rule structures. This causes limitations regarding complex sentences that have related words located far from each other in a sentence. Additionally, the annotator cannot account for all

[55] https://www.waze.com/ccp

possible spelling variations of words within tweets. Moreover, the Dutch language is known for its many word compounds, especially compared to English, which also causes problems for the annotator. Second, ambiguity of words is a hard problem to deal with for an annotator. However, our annotator is able to catch a great number of these cases by defining grammatical rule structures. Third, there is a special case of ambiguity regarding place indicators. As we use a library composed of place terms from GeoNames and OpenStreetMap, which contain a great number of ambiguous terms that could either be a location or a common word used in the Dutch language. Having to deal with that many downsides towards the use of a rule-based annotator, begs the question why we did not apply a machine learning approach to classify the tweets to traffic event categories. The first counterargument for such an approach would be that we are not only interested to which traffic event categories a tweet belongs, but also in the tweet text itself that indicates that category. Second, as shown before, the number of collected tweets is too sparse to be able to train a classifier that is able to classify tweets into such a wide range of categories as our annotator currently does. With the help this annotator we have been able to annotate our tweet set and provide traffic event-related categories to them.

5.3 Traffic Event Classification

In order to automatically detect if a tweet is traffic event-related or not, a supervised binary classification approach was applied. Based on an evaluation of multiple classifier and feature combinations a classifier has been chosen based on the best combination of precision (0.62) and recall (0.61) values for detecting traffic event-related tweets. This classifier is based on Linear SVM with a random over-sampler and also performs best based on the combination of an average f1-score of 0.95, accuracy of 0.954 and AUC ROC of 0.955. Due to time constraints, we limited our experiment to only two different classification algorithms based on Naïve Bayes and Support Vector Machine algorithms which according to previous works and theory should perform best for text-based documents such as tweets. Also, for this reason, we did not include pre-processing techniques such as lemmatization and features such as part of speech tagging. Therefore, other classification algorithms with different feature sets could provide better results than presented in this study. As our precision, recall, and f1-scores greatly differ based on the detection of traffic event or non-traffic event-related tweets, we compare the weakest values (traffic event-related) to related work. Machine learning classifier results presented in related work seem to outperform the results in our work significantly, e.g., in the work by D'Andrea et al. (2015), a precision of 0.953, recall of 0.965 and f1-score of 0.958 are achieved. However, we again want to strongly emphasize that we specifically manually annotated traffic event-related tweets from real road-users as traffic event-related and annotated traffic event-related tweets from other users as non-traffic event-related. This means that classification of traffic event-related tweets in our case is much more difficult, as traffic event-related tweets from news agencies, bots, and emergency agencies contain a much more structured syntax than tweets from real road-users. Hence, the better results in previous related work. All in all, the results of the classifier evaluation can be seen as somewhat disappointing, especially when considering that the amount of traffic event-related tweets per day by itself is already very sparse.

5.4 Geocoding

In order to assign geographic locations to tweets, a geocoding method was developed that uses spatial indicators in tweets, annotated by our rule-based traffic domain annotator, to derive locations. Based on an evaluation of this method, we were able to determine that the majority (49%) of the tweets can be geocoded to a location that covers all place indicators in the tweet and include no irrelevant locations. Additionally, 37% of the geocoded tweets include all relevant place indicators, however also a number of irrelevant place indicators. The remaining 14% of the tweets either is geocoded to a part of relevant indicators or to no relevant indicators at all. What also came forward from these results, is that the more token sets categorized as place indicators a tweet contains, the more difficult it becomes for our geocoding method to correctly assign all locations to a tweet. This is most likely caused due to the fact that a tweet can refer to multiple unrelated locations. In our approach, as thoroughly explained in Section 3.6.1, we opted to first and foremost tackle the challenges spatial indicators bring with them such as issues regarding: contradictions, confirmations, scaling and ambiguity. A model was therefore designed that computes the intersections of a multitude of spatial indicators in a tweet.

Compared to geocoding methods in related work we see that most work relies either on the device location automatically added when posting a tweet, or on a simplified geolocation method that links a tweet to a single geopoint. The geocoder presented in the work by Gu et al. (2016) comes closest to our geocoding approach, as it extracts road names, intersection names, highway exit numbers and highway mile markers to compute a single geolocation. With this method they were able to geocode 64.0% of tweets by influential users, and 4.9% of tweets created by individual users. Compared to these results our geocoding method performs significantly better.

5.5 Traffic Event Description

A traffic event description module was developed to cluster related information from traffic event reports (TE tweets, Waze and TomTom events) and DiTTLab traffic data. This clustering is performed based on a rule-based matching approach in which a rule specifies the categorical, spatial and temporal extent used to assert if the new traffic event report should be part of an existing traffic event cluster. In the evaluation of this approach, we used 13 unique event categories, a radius of 150 meters, and a timespan of 15 minutes. We applied this description method over the collected Twitter, Waze, TomTom and DiTTLab data over the period from 05-12-2017 to 06-01-2018, and the period from 07-01-2018 to 17-02-2018. Afterwards, we removed the clusters over periods where data was missing due to technical issues. We found that for the first period 44.82% of the traffic event clusters consists out of more than one event report, and for the second period 34.39%. One explanation for this could be that our rule-based matching approach is too strict, regarding the spatial radius and temporal timespan. Another explanation only related to tweets relies on the matching on dates that is performed based on the creation date of traffic events. Where the creation date of a Waze and TomTom report event is highly likely to be very closely related to the datetime of the actual event occurred, this does not necessarily have to be the case for tweet reports. Tweets can also refer to past and future events, as well as refer to longer timespans. Additionally, tweets can be created at any location and refer to any location, whereas Waze and TomTom reports can only refer to the location at which they were created. Besides, our traffic event description evaluation was performed based on a rule with a radius of 150 meters and a timespan of 15 minutes for all 13 event categories. Results could improve when rules with customized values are specified for each event category by a traffic domain expert. Furthermore, we evaluated how many clusters could be linked to DiTTLab traffic data. We were able to link 15,128 out of the 31,089 subcluster locations (48,66%). This amount is very reasonable taking into account that we only have DiTTLab traffic data available for motorways in the Netherlands. Lastly, we looked at the spatial coverage of traffic events, by analyzing the spatial properties of traffic jams with the categories accident/ traffic jam. This analysis showed that event locations appear more often around larger cities and motorways. An explanation for this could be that the larger amount of road traffic in these regions reflects on the amount of geosocial reports.

5.6 SocialTerraffic System

A web-based interactive map application named SocialTerraffic was developed, to provide a way to present the collected and processed data, by our pipeline, to the end user. In Section 3.8.2, we specified a list of five must have and one should have requirements for the application. The first requirement stated that a user must be able to view the locations of traffic events on an interactive map. We met this requirement, as in our application a user can create and send a query based on event level, report type, category, and date range, to the database. The database returns the clustered traffic events and maps their location shapes on an interactive map. The second requirement stated that a user must get an overview of all traffic domain categories and their count, and a description for a specific traffic event. This requirement has also been met, as we implemented a timeline that shows the categories, for the queried events, with their count over time. Additionally, when a user clicks on an even location on the map or timeline a traffic event info submenu appears that shows event related information. The third requirement described that a user must be able to filter traffic events based on event category, time range and location. We met this requirement by implementing a control panel that allows for quick switching between and filtering out traffic event categories. The developed timeline can be zoomed in/out on to focus on events within a specific time frame. Similarly, the map can be zoomed in/out on to focus on events within a specific location. The fourth requirement stated that a user must be able to view the traffic event reports that are linked to a traffic event. This requirement was also met, as we included the information of all reports that a traffic event cluster consists out of. This information is put in a related reports section, represented as an accordion (collapsible content), in the traffic event info submenu. The fifth requirement denoted that a user must be able to view DiTTLab traffic data that is linked to a traffic event. The sixth requirement stated that a user should be able to view automatically generated graphs by selecting a traffic domain category and timespan for a specific location. We met both requirements by implementing a DiTTlab traffic data submenu. This menu becomes active after a user has selected a traffic event, and enables the user to generate speed/flow charts based on DiTTLab traffic data and a specified time range.

We were able to partly compare our application to related work from Daly et al. (2013), Dokter (2015), Nguyen et al. (2016). From all reviewed related work, these works were the only ones with some form of custom application that was comparable to ours. Daly et al. (2013), developed a web client named Dub-STAR, where traffic congestions are drawn on a map and related detected events are linked to these congestions. Users are able to filter the events based on a list of specific event types. Dokter (2015), developed a CouchApp based web-interface that links tweets to traffic events. These events were mapped, and the tweets were used to provide descriptive information. Nguyen et al. (2016), created an interface in which geo-located traffic event-related tweets are mapped on a 3D Bing map in a real-time fashion. A timeline is included to filter events based on a date range. Based on applications developed in related work we can say that our application contains all features presented in these works. Moreover, our application contains a lot of features unseen in previous work and in addition contains an intuitive graphical user interface.

6 Conclusions

In this chapter, an answer to the main research question: "*To what extent can geosocial data enrich traffic data to improve the detection, categorization, and description of non-recurrent traffic events?*", is provided. Additionally, an outlook for future work is provided.

6.1 Conclusion

In this work, we described to what extent geosocial data is able to enrich traffic data, by creating a pipeline that is able to collect, detect, categorize, and cluster social and traffic data in order to provide a description of non-recurrent traffic events. In order to answer our main research question we divided it into five separate research sub-questions.

RQ1: *What is the current state of the art regarding non-recurrent traffic event detection, categorization, and description by using traffic data and geosocial data, individually or combined?*

We presented, discussed and compared related work on traffic event detection, categorization, and description divided by traffic-, social-, and the combination of traffic and geosocial data sources. We discovered that in studies related to only traffic data, traffic event detection proved to be the only focus point. In these studies, traffic event detection is based on algorithms that depend on data from roadway-based sensors. Weaknesses of this approach are related to the quality of measurements which depend on the density of the road sensor network, and the fact that algorithms are road-type dependent. Studies related to only geosocial data contained traffic event detection, categorization, and description approaches. We found that weaknesses of these approaches are based on the use of only one type of data source, biased data collection approaches, non-existing categorization approaches and limited geocoding techniques. Lastly, we found that studies related to the combination of traffic data and geosocial data are limited in amount and distinguish themselves mostly in their traffic event description approach. However, a weakness of this approach in these works is, that a mandatory combination of traffic data anomalies and geosocial data is enforced which could lead to a loss of important semantic data.

RQ2: *How can non-recurrent traffic event-related geosocial posts be detected?*

We developed a Twitter, Waze and TomTom data collection approach. Waze and TomTom event reports are by default traffic event-related and are thus automatically detected at this stage. In order to detect traffic event-related tweets out of our collected tweet set, multiple traffic event classifiers have been trained based on a supervised binary classification approach. Based on a comparison of the evaluated classifiers a traffic event classifier has been created that yields a good performance for detecting non-traffic event-related tweets and a sufficient performance for detecting traffic event-related tweets.

RQ3: *How can detected non-recurrent traffic event-related geosocial posts be categorized by event type?*

In order to categorize tweets on multiple traffic event-related event types, a rule-based traffic domain annotator has been created. This annotator is able to categorize the token sets of a tweet into 27 unique traffic related categories, from which 13 traffic event-related. Additionally, we were able to categorize Waze and TomTom event reports into one of these same 13 traffic event categories.

RQ4: *How can categorized geosocial posts be used to describe non-recurrent traffic events?*

We created a traffic event description method based on a rule-based approach, in which a rule specifies the categorical, spatial and temporal extent, used to assert if the new traffic event report should be part of an existing traffic event cluster. Traffic event reports are matched on a category based on 13 unique traffic event categories. Each traffic event category forms its own rule defining a radius and time range suited for that category. As tweets do not contain a geolocation from themselves, a custom geolocation approach has been developed to assign a location to a tweet. This method uses spatial indicators in tweets, as annotated by our rule-based traffic domain annotator. The method links these spatial indicators to a geographic location and uses an intersection technique to find a list of most relevant locations in a tweet. The evaluation of this method provided good results and allowed us to cluster tweets to the other data. The evaluation of the cluster approach showed that many event reports get lost as they cannot be clustered together with other event reports. Out of the clusters that could be created almost half could also be linked to DiTTLab traffic data.

RQ5: *How to develop a software system that is able to perform the detection, categorization, and description of non-recurrent traffic events?*

With the parts developed in the answering of RQs 2 to 4, a pipeline was developed that is able to perform the detection, categorization and description of traffic events, and store this data in a database. To present the collected and processed data to the user a web-based interactive map application was developed named SocialTerraffic. This application enables the user to view the traffic events and their descriptions on an interactive map. Besides, with this application a user is able to filter traffic events based on event category, date range and location. Additionally, the application is able to generate speed/flow charts based on traffic data related to a traffic event.

All in all, this work shows that geosocial data is able to enrich traffic data to improve the detection, categorization, and description of non-recurrent traffic events.

6.2 Future Work

This work contains many opportunities for feature research to reduce some of the threats to the validity of and to extend upon our work. First, there could be made improvements to our keyword selection approach in order to determine the cause of the limited amount of traffic event-related tweets collected in this study. Second, our approach could be extended by using more data sources that provide real road-user data. Any text-based source could be usable as our traffic domain annotator and geolocation method are not Twitter bound. Third, the traffic domain annotator could be enriched with more advanced grammatical rule structures, to reduce the ambiguity problems. Fourth, traffic event classification is performed based on only two different classifier algorithms. By comparing the achieved results in this work with classifiers based on other classifier algorithms, overall results could improve. Also, pre-processing elements and features that were not implemented because of time constraints could be included in feature work. Fifth, in this work a tweet can be assigned to multiple event categories as well as multiple locations. However, it is assumed that all locations refer to all event categories, whereas there could exist multiple location-category couples within a tweet. It would be interesting to make a distinction between these two in feature work. Sixth, due to time constraints, we did not implement a temporal linking approach based on the text of a tweet. Such temporal linking approach could be useful, as the creation date of a tweet is not necessarily a reflection of the time a traffic event described in that tweet occurred. For example, an event could have happened in the past, is still ongoing, or could happen in the future. In this work, we already have created an annotator that is able to extract temporal expressions from tweet text. As future work, one could combine these temporal expressions with the creation date of a tweet to calculate the most probable datetime range for the traffic events in a tweet. In order to achieve this, one would need to create a parser that is able to parse human readable temporal expressions to machine-readable dates. With this approach another challenge arises, as tweets can then be linked to a datetime range instead of a single datetime. As Waze and TomTom event reports always contain a single datetime this could cause matching problems and the traffic event description method has to be adapted to deal with this mismatch problem. Seventh, our current version of the pipeline used for the SocialTerraffic application is not able to process data in real time. With some adaptions a real time version of SocialTerraffic could be developed. Lastly, in our SocialTerraffic application we already showed how traffic data can be linked to traffic events and used to create speed/flow heat maps. Therefore, it would be interesting to integrate SocialTerraffic into the application from DiTTLab.

Bibliography

Ahmed, M. S., & Cook, A. R. (1979). *Analysis of freeway traffic time-series data by using Box-Jenkins techniques* (No. 722).

Aköz, Ö., & Karsligil, M. E. (2014). Traffic event classification at intersections based on the severity of abnormality. *Machine vision and applications*, *25*(3), 613-632.

Bosch, A. V. D., Busser, B., Canisius, S., & Daelemans, W. (2007). An efficient memory-based morphosyntactic tagger and parser for Dutch. *LOT Occasional Series*, *7*, 191-206.

Chawla, N. V. (2009). Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*(pp. 875-886). Springer, Boston, MA.

Chen, P. T., Chen, F., & Qian, Z. (2014, December). Road traffic congestion monitoring in social media with hinge-loss Markov random fields. In *Data Mining (ICDM), 2014 IEEE International Conference on* (pp. 80-89). IEEE.

Cui, J., Fu, R., Dong, C., & Zhang, Z. (2014, October). Extraction of traffic information from social media interactions: Methods and experiments. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on* (pp. 1549-1554). IEEE.

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

D'Andrea, E., Ducange, P., Lazzerini, B., & Marcelloni, F. (2015). Real-time detection of traffic from twitter stream analysis. *IEEE transactions on intelligent transportation systems*, *16*(4), 2269-2283.

Van der Veer, N., Boekee, S., & Peters, O. (2016). Nationale social media Onderzoek 2016. *Newcom Research & Consultancy BV.*

Dokter, E. (2015). Characterization of traffic events using social media.

Dudek, C. L., Messer, C. J., & Nuckles, N. B. (1974). Incident detection on urban freeways. *Transportation Research Record*, *495*, 12-24.

Forbes, G. J., & Hall, F. L. (1990). The applicability of catastrophe theory in modelling freeway traffic operations. *Transportation Research Part A: General*, *24*(5), 335-344.

Giridhar, P., Amin, M. T., Abdelzaher, T., Kaplan, L. M., George, J., & Ganti, R. (2014, March). Clarisense: Clarifying sensor anomalies using social network feeds. In Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on (pp. 395-400). IEEE.

Giridhar, P., Amin, M. T., Abdelzaher, T., Wang, D., Kaplan, L., George, J., & Ganti, R. (2016). ClariSense+: An enhanced traffic anomaly explanation service using social network feeds. *Pervasive and Mobile Computing*, *33*, 140-155.

Gu, Y., Qian, Z. S., & Chen, F. (2016). From Twitter to detector: Real-time traffic incident detection using social media data. *Transportation research part C: emerging technologies*, *67*, 321-342.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, *3*(Mar), 1157-1182.

Hawas, Y. E. (2007). A fuzzy-based system for incident detection in urban street networks. Transportation Research Part C: Emerging Technologies, 15(2), 69-95.

He, J., Shen, W., Divakaruni, P., Wynter, L., & Lawrence, R. (2013, August). Improving Traffic Prediction with Tweet Semantics. In *IJCAI* (pp. 1387-1393).

Hürriyetoglu, A., Oostdijk, N., & den Bosch, A. (2014, 3). (Data set For) "Estimating Time to Event from Tweets Using Temporal Expressions".

Ikeda, H., Kaneko, Y., Matsuo, T., & Tsuji, K. (1999). Abnormal incident detection system employing image processing technology. In *Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on*(pp. 748-752). IEEE.

Ivan, J. N., Schofer, J. L., Koppelman, F. S., & Massone, L. L. (1995). Real-time data fusion for arterial street incident detection using neural networks. *Transportation Research Record*, (1497), 27-35.

Joachims, T. (1998, April). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning* (pp. 137-142). Springer, Berlin, Heidelberg.

Kamran, S., & Haas, O. (2007, June). A multilevel traffic incidents detection approach: Identifying traffic patterns and vehicle behaviours using real-time gps data. In *Intelligent vehicles symposium, 2007 ieee* (pp. 912-917). IEEE.

Kon, T. (1998). Collision warning and avoidance system for crest vertical curves.

Kumar, A., Jiang, M., & Fang, Y. (2014, July). Where not to go?: detecting road hazards using twitter. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval* (pp. 1223-1226). ACM.

Levin, M., & Krause, G. M. (1978). Incident detection: A Bayesian approach. *Transportation Research Record*, (682).

Li, R., Lei, K. H., Khadiwala, R., & Chang, K. C. 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.

Li, X., & Porikli, F. M. (2004, October). A hidden Markov model framework for traffic event detection using video features. In *Image Processing, 2004. ICIP'04. 2004 International Conference on* (Vol. 5, pp. 2901-2904). IEEE.

Murphy, K. P. (2006). Naive bayes classifiers. *University of British Columbia, 18.*

Nguyen, H., Liu, W., Rivera, P., & Chen, F. (2016, April). TrafficWatch: Real-Time Traffic Incident Detection and Monitoring Using Social Media. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 540-551). Springer, Cham.

Niver, E., Mouskos, K. C., Batz, T., & Dwyer, P. (2000). Evaluation of the TRANSCOM's system for managing incidents and traffic (TRANSMIT). *IEEE Transactions on intelligent transportation systems*, *1*(1), 15-31.

Oostdijk, N. H. J., Hürriyetoğlu, A., Puts, M., Daas, P., & van den Bosch, A. P. J. (2016). Information extraction from social media: A linguistically motivated approach.

Ribeiro Jr, S. S., Davis Jr, C. A., Oliveira, D. R. R., Meira Jr, W., Gonçalves, T. S., & Pappa, G. L. (2012, November). Traffic observatory: a system to detect and locate traffic events and conditions using Twitter. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Location-Based Social Networks* (pp. 5-11). ACM.

Tostes, A. I. J., Silva, T. H., Duarte-Figueiredo, F., & Loureiro, A. A. (2014, September). Studying traffic conditions by analyzing foursquare and instagram data. In *Proceedings of the 11th ACM symposium on Performance evaluation of wireless ad hoc, sensor, & ubiquitous networks* (pp. 17-24). ACM.

Schulz, A., Ristoski, P., & Paulheim, H. (2013, May). I see a car crash: Real-time detection of small scale incidents in microblogs. In *Extended Semantic Web Conference* (pp. 22-33). Springer, Berlin, Heidelberg.

Schulz, A., Schmidt, B., & Strufe, T. (2015, August). Small-scale incident detection based on microposts. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media* (pp. 3-12). ACM.

Sermons, M. W., & Koppelman, F. S. (1996). Use of vehicle positioning data for arterial incident detection. *Transportation Research Part C: Emerging Technologies*, *4*(2), 87-96.

Sethi, V., Bhandari, N., Koppelman, F. S., & Schofer, J. L. (1995). Arterial incident detection using fixed detector and probe vehicle data. *Transportation Research Part C: Emerging Technologies*, *3*(2), 99-112.

Silva, T. H., de Melo, P. O. V., Viana, A. C., Almeida, J. M., Salles, J., & Loureiro, A. A. (2013, November). Traffic condition is more than colored lines on a map: characterization of waze alerts. In *International Conference on Social Informatics* (pp. 309-318). Springer, Cham.

Speer, R., & Havasi, C. (2012, May). Representing General Relational Knowledge in ConceptNet 5. In *LREC* (pp. 3679-3686).

Stephanedes, Y. J., & Chassiakos, A. P. (1993). Freeway incident detection through filtering. *Transportation Research Part C: Emerging Technologies*, *1*(3), 219-233.

Systematics, C. (2005). Traffic congestion and reliability: Trends and advanced strategies for congestion mitigation. *Final Report, Texas Transportation Institute.*

Tignor, S. C., & Payne, H. J. (1977). Improved freeway incident detection algorithms. *Public roads*, *41*(1).

Wang, S., He, L., Stenneth, L., Yu, P. S., & Li, Z. (2015, November). Citywide traffic congestion estimation with social media. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems* (p. 34). ACM.

Wanichayapong, N., Pruthipunyaskul, W., Pattara-Atikom, W., & Chaovalit, P. (2011, August). Social-based traffic information extraction and classification. In *ITS Telecommunications (ITST), 2011 11th International Conference on* (pp. 107-112). IEEE.

Willsky, A., Chow, E., Gershwin, S., Greene, C., Houpt, P., & Kurkjian, A. (1980). Dynamic model-based techniques for the detection of incidents on freeways. *IEEE Transactions on Automatic Control*, *25*(3), 347-360.

Yaguang, K., & Anke, X. (2006, November). Urban traffic incident detection based on fuzzy logic. In *IEEE Industrial Electronics, IECon 2006-32nd Annual Conference on* (pp. 772-775). IEEE.

Young, S. (2007, August). Real-time traffic operations data using vehicle probe technology. In Proceedings of the 2007 Mid-Continent Transportation Research Symposium (pp. 16-17).

Appendix A: Source Code Repository

The source code for the SocialTerraffic System, the pipeline and all other code used for analysis purposes, is available on the following GitHub repository:

https://github.com/BdeBock/SocialTerraffic

However, please note that this repository is private and an invite is needed to access it.

Appendix B: Rule-based Traffic Domain Annotator Grammar

Please note that the rules in this appendix provide a simplified overview of the actual implementation, and that the dictionaries are only defined once in order of their appearance.

1. No Applicable Category (N/A)

Goal: annotate tokens that are not matched by other categories

2. Media Attachment

Goal: annotate indicators of date and time.
Rules:

- R1: Combination of (https://t.co/ + Word(Alpha token + Number token))

Examples:

- E1: #A16 Li 16,9 https://t.co/ovmSUIHLMv

3. Temporal (Timex)

Goal: annotate indicators of date and time.
Dictionary:

- Timex: gistermiddag rond 12:00 uur, 10 minuten geleden, daarnet.

Rules:

- R1: Timex token

Examples:

- E1: Daarnet langs een kopstaartbotsing gereden richting Middelburg.

4. Advice

Goal: annotates announcements or guidance related to traffic.
Dictionary:

- Advices: e.g., pas je snelheid aan, gevaarlijke situatie
- Advice activities: e.g., omrijden, keer om, wijk uit

Rules:

- R1: Advices token | Advice activities

Examples:

- E1: Pas je snelheid aan er heeft net een ongeluk plaatsgevonden op de A10

5. Road user transport

Goal: annotate indicators of groups of traffic.
Dictionary:

- Traffic: e.g., bestemmingsverkeer, vrachtverkeer, colonne.

Rules:

- R1: Traffic token

Examples:

- E1: Veel vakantieverkeer richting Amsterdam vandaag.

6. Road User Casualty

Goal: annotate injured road users.
Dictionary:

- Traffic road users: e.g., automobilist, inzittende, tegenligger.
- General road users: e.g., man, personen, volwassenen.
- Casualties: e.g., slachtoffer, doden.
- Injuries: e.g., gewonden, verongelukt, letsel.
- Injury adjectives : e.g., ernstig, eenzijdig, lichte.
- Quantifiers: e.g., twee, alle, meerdere.

Rules:

- R1: Optional(Quantifiers token) + Optional(Traffic road users token | General road users token | Casualties token) + Optional(Injury adjectives token) + Injuries token
- R2: Optional(Quantifiers token) + Optional(OneOrMore(Injury adjectives token)) + (Traffic road users token | General road users token | Casualties token)
- R3: Optional(Quantifiers token) + Casualties token

Examples:

- E1: Meerdere inzittenden ernstig gewond bij kettingbotsing op de A10.
- E2: Tweetal ernstig gewonde voetgangers.
- E3: Er is een slachtoffer gevallen bij een ongeluk op de A5.

7. Road User Traffic

Goal: annotate road user persons.
Dictionary:

- Traffic road users: e.g., automobilist, inzittende, tegenligger.

Rules:

- R1: Optional(Quantifiers token) + Traffic road users token

Examples:

- E1: Meerdere automobilisten betrokken bij ongeval.

8. Road User General

Goal: annotate general persons.
Dictionary:

- General road users: e.g., automobilist, inzittende, tegenligger.

Rules:

- R1: Optional(Quantifiers token) + General road users token

Examples:

- E1: Meerdere automobilisten betrokken bij ongeval.

9. Road User Vehicle

Goal: annotate vehicle names and their brands.

Dictionary:

- Vehicle names: e.g., auto, vrachtwagen, caravan.
- Vehicle brands: e.g., Nissan, Volvo, Hobby.
- Vehicle colors: e.g., red, blue, grey.

Rules:

- R1: Optional (Quantifiers token | Vehicle colors token) (Vehicle names token | Vehicle brands token).

Examples:

- E1: Een rode Auto met caravan achterop #busje geklapt.

10. Road User Emergency service

Goal: annotate road user emergency services and their status.

Dictionary:

- Road user emergency services: e.g., ambulance, anwb, politie.
- Road user emergency service status: e.g., aanrijdend, ter plaatse, op locatie.

Rules:

- R1: Optional(Quantifiers token) + Road user emergency services token + Optional(zijnLit) + Optional(Road user emergency service status token)

Examples:

- E1: Ongeval bij knooppunt Amstel politie is ter plaatse.

11. Place Location

Goal: annotate exact locations on road infrastructure, amenities, buildings, etc. A location must contain a geopoint, geoline, or geoshape.

Dictionary:

- Places: Combination of buildings, amenities, places, etc., e.g., Zeeland, TU Delft, de Kuip
- Road numbers: e.g., A10, N12, s101.
- Infrastructures (suffix based): Combination of a custom word + infrastructure suffix, e.g., Lndbergstraat, Wstrschldetunnel, Leidddseplein

Rules:

- R1: (Places token | Infrastructures token | Road numbers token)

Examples:

- E1: Ongeluk met twee auto's #A10.

12. Place Location Combination

Goal: annotate combinations of areas that have unique physical and human characteristics, and locations.

Dictionary:

- Mile markers: Combinations of road number, marker, and road side tokens, e.g., hmp 10.2, hectometerpaaltje 13.1
- Road lanes: e.g., linker rijbaan, spitsstrook, greppel
- Infrastructure types: refer to rule *16. Place Infrastructure Type*
- Infrastructure suffix: Lndbergstraat, Wstrschldetunnel, Leidddseplein

Rules:

- R1: Optional(tennoordenvanLit | tenwestenvanLit | tenzuidenvanLit | tenoostenvanLit | thvLit | opzijvanLit | nabijLit | bijLit | vlakLit | naastLit |opLit | inLit | linksrechtsvanLit) + Optional(~(Places token) + deLit) + (Road numbers token | Infrastructure suffix token | Mile marker token | Places token |Infrastructure types token) + Optional(deLit) + (Road numbers token | Infrastructure suffix token | Mile marker token | Places token | Infrastructure types token)
- R2: R1 + (thvLit | opzijvanLit | nabijLit | bijLit | vlakLit | naastLit |opLit | inLit | linksrechtsvanLit) + Optional(~(Road numbers token | Infrastructure types token | Infrastructure suffix token | Mile marker token | Amenities token | Buildings token | Places token | Leisure token) + Optional(~(Places token) + deLit) + Road numbers token | Infrastructure types token | Infrastructure suffix token | Mile marker token | Places token)
- R3: Optional(tennoordenvanLit | tenwestenvanLit | tenzuidenvanLit | tenoostenvanLit | thvLit | opzijvanLit | nabijLit | bijLit | vlakLit | naastLit |opLit | inLit | linksrechtsvanLit) + Optional(~(Places token) + deLit) + (Road numbers token | Infrastructure suffix token | Mile marker token| Places token) + (thvLit | opzijvanLit | nabijLit | bijLit | vlakLit | naastLit |opLit | inLit | linksrechtsvanLit) + (Road numbers token | Infrastructure suffix token | Mile marker token| Places token)

Examples:

- E1: Ten noorden van de A10 is een ongeluk gebeurd.
- E2: Autobotsing bij de McDonald's in Amsterdam.
- E3: Op de A10 staat een auto stil #Vlissingen.

13. Place Road Section

Goal: annotate places that make a reference to a location

Dictionary:

- Directions: e.g., ->, =&gt; , in Noordelijke richting

Rules:

- R1: tssnLit + Optional(deLit) + (Road numbers token | Infrastructure suffix token | Mile marker token | Infrastructure types token | Places token) + enLit + Optional(deLit) + (Road numbers token | Infrastructure suffix token | Mile marker token | Infrastructure types token |Places token)
- R2: Optional(vanafLit | vanLit | vanuitLit) + Optional(deLit) + (Road numbers token | Infrastructure suffix token | Mile marker token | Infrastructure types token | Places token) + Optional(deLit) + (totLit | totAanLit | naarLit | Directions token) + Optional(deLit) + (Road numbers token | Infrastructure suffix token | Mile marker token | Infrastructure types token| Places token)

Examples:

- E1: Tussen hmp 21.2 en 56.3 staat een file.
- E2: File vanaf knooppunt Amstel tot aan hmp 13.5

14. Place Road Direction

Goal: annotate a specific road direction

Dictionary:

- Countries: e.g. Belgie, Duitsland

Rules:

- R1: (Directions token | naarLit) + Optional(~(Places token | Countries token | Road numbers token | Infrastructure suffix token | Mile marker token) + Arbitrary token) + (Places token | Countries token | Road numbers token | Infrastructure suffix token | Mile marker token)
- R2: (inLit | vanuitLit) + Directions token

Examples:

- E1: In de richting van Amsterdam staat het vast.
- E2: Vanuit tegenovergestelde rijrichting rijden ambulances aan.

15. Place Road Mile Marker

Goal: annotate a specific road direction

Dictionary:

- Road markers: e.g., hectometerpaal, htm, paaltje
- Road side: e.g., links, li, re

Rules:

- R1: Road numbers token + (((Optional(Road marker token) + Float number token + Optional(Road side token)) | (Road side token + Optional(Optional(Road marker token) + Float number token))))
- R2: Float number token + (Road numbers token + Optional(Road side token) | Road side token + Optional(Road numbers token))
- R3: (Road side token | linksrechtsVanLit) + ((Optional(Road marker token) + Float number token + Optional(Road numbers token))
- R4: Road marker token + Float number token + Optional(Road side token | Road numbers token)

Examples:

- E1: A10 10.2 li
- E2: 12.3 A5 re
- E3: links van hmp 12.3
- E4: Bij hectometer 12.3 op de A10 staat een auto stil.

16. Place Infrastructure Type

Goal: annotate various types of road infrastructures.

Dictionary:

- Infrastructures: e.g., knooppunt, knp, tunnel
- Roads: e.g., autobaan, ringweg, parallelweg

Rules:

- R1: (Infrastructure token | Road token)

Examples:

- E2: Knooppunt Amstel staat weer vast.

17. Place Road Lane

Goal: extract a specific lane of a road.

Rules:

- R1: Optional(opLit | opzijvanLit | naastLit | linksrechtsvanLit) + Optional(deLit) + Road lane token

Examples:

- E1: Olie op de vluchtstrook nabij Utrecht.

18. Event Accident

Goal: annotate traffic collisions (including consequences) between vehicles and other vehicles, pedestrians, animals, road debris, or other stationary obstructions.

Dictionary:

Vehicle status:	e.g., brand, problemen, slip
Accident types:	e.g., autobrand, aanrijding, frontale botsing
Accident types adjectives:	e.g., dodelijk, ernstige, levensgevaarlijke
Accident adjectives:	e.g., gekantelde, geschaarde, vastgelopen
Accident verbs:	e.g., omgevallen, gekanteld, van de weg geraakt
Objects:	e.g., boom, obstakel, punaises
Traffic lights:	e.g., stoplicht, lantaarnpaal
Traffic signs:	e.g., bewegwijzering, matrixbord, wegmarkering
Animals:	e.g., zwaan, hert, wild

Rules:

- R1: Optional(Quantifiers token) + Accident adjectives token + Optional(Vehicle colors token) + (Vehicle names token | Vehicle brands token)
- R2: Optional(Quantifiers token) + Optional(Vehicle colors token) + (Vehicle names token | Vehicle brands token) + Accident verbs token
- R3: Optional(vanwegeLit | ivmLit | doorLit | alsgevolgvanLit | metalsgevolgLit | naLit | bijLit) + Optional(Accident type adjective token) + Accident type token + Optional(metLit | tegenLit | opLit | tussenLit) + Optional(Quantifiers token) + Optional(Accident type adjective token) + (Road user persons token | Road user persons token | Road user emergency services token | Vehicle names token | Vehicle brands token | Objects token | Traffic lights token | Traffic signs token | Animals token) + Optional(enLit + (Road user persons token | Road user persons token | Road user emergency services token | Vehicle names token | Vehicle brands token | Objects token | Traffic lights token | Traffic signs token | Animals token))
- R4: Optional(Quantifiers token) + Optional(Vehicle colors token) + (Vehicle names token | Vehicle brands token) + Optional(~(Vehicle status token) + SM token) + Optional(inLit | tegenLit) + Optional(~(Vehicle status token) + SM token) + Vehicle status token
- R5: Optional(Accident type adjectives token) + Accident token

Examples:

- E1: Twee geschaarde zwarte auto's op de A10.
- E2: Mercedes van de weg geraakt bij knooppunt Amstel.
- E3: Vanwege ernstig ongeluk met overstekend hert.
- E4: Auto half in de berm gelukkig geen gewonden.
- E5: Rij net langs een zorgelijke aanrijding bij hmp 12.6.

19. Event Traffic jam

Goal: annotate indicators of a traffic jam

Dictionary:

- Flows: e.g., doorstromen, langzaam rijden, verkeersopstopping
- Intensities: e.g., drukte, spits, verkeersintensiteit
- Intensity adjectives: e.g., erg, enorme, korte
- Durations: e.g., min, minuten, uren
- Distances: e.g., m, km, kilometer

Rules:

- R1: Optional(Intensity adjectives token) + (Flows token | Intensities token) + (vanLit | voorLit) + Optional(~Quantifiers token + SM token) + Optional(Quantifiers token) + (Distances token | Durations token)
- R2: Optional(Quantifiers token) + Distances token + Flows token
- R3: Optional(Quantifiers token) + Durations token + Optional(~(Flows token) + Arbitrary token) + Flows token
- R4: Optional(Intensity adjectives token) + (Flows token | Intensities token)

Examples:

- E1: Korte file van 10 minuten voor de Kuip.
- E2: Kilometer stapvoets rijdend richting Amsterdam.
- E3: 10 min wachten vanwege ongeluk voor ons.
- E4: Mega paasdrukte op de A10.

20. Event Closure

Goal: annotate indicators of road/lane closures

Dictionary:

- Closures: e.g., afsluiting, afkruizing, wegversperring
- Closure status: e.g., afgesloten, dicht, geblokkeerd
- Closure adjective: e.g., dichte, afgebakende, versperde
- Closure signs: e.g., ✕, rood kruis

Rules:

- R1: Optional(Quantifiers token) + Optional(deLit) + (Road lanes token | Infrastructures token | Roads token) + Optional(~Closure status token + Arbitrary token) + Closure status token
- R2: Optional(Quantifiers token) + Closure adjective token + (Road lanes token | Infrastructures token | Roads token)
- R3: Optional(inLit | vanuitLit) + Directions token + (R1 | R2)
- R4: (R1 | R2) + Optional(inLit | vanuitLit) + Directions
- R3: (Closures token | Closure signs token| Closure status token)

Examples:

- E1: Meerdere rijbanen zijn afgesloten
- E2: Afgebakende vluchtstrook richting knooppunt Amstel.
- E3: In westelijke richting rijbaan afgesloten.
- E4: Rijbanen gesloten in beide richtingen.
- E5: Doorgaand rijverkeer gestremd tot 21:00 vanavond.

21. Event Enforcement

Goal: annotate indicators of activities held by traffic enfocement agencies
Dictionary:

- Monitoring: e.g., alcoholcontrole, snelheidscontrole, flitser

Rules:

- R1: Monitoring token

Examples:

- E1: Alcoholcontrole op de A2 richting Den Bosch.

22. Event Hazard Violation

Goal: annotate indicators of law violating activities
Dictionary:

- Violations: e.g., bumperkleven, spookrijden, afsnijden
- Speeds verbs: e.g., scheuren, racen, rijden, scheurt
- Speed limits: e.g., adviessnelheid, snelheidslimiet
- Speed adjectives: e.g., hoge, maximum
- Speed adverbs: e.g., te hard, te snel

Rules:

- R1: Speed adverbs token + Speed verbs token
- R2: Speed verbs token + Speed adverbs token
- R3: Optional(Speed verbs token) + Speed adverbs token + Optional(Speed verbs token) + danLit + Optional(~(Speed limits token) + Arbitrary token) + Speed limits token
- R4: (Violations token | Speed adverbs token)

Examples:

- E1: Auto kwam hard aanrijdend en botste met voorligger.
- E2: Zie auto's weer veel te hard rijden op de A10.
- E3: Meerdere auto's rijden harder dan is toegestaan op de A58.
- E4: Auto achter me loopt weer lekker te bumperkleven.

23. Event Hazard Traffic Sign

Goal: annotate indicators of broken or unreadable, or missing traffic signs.
Dictionary:

- Traffic signs: e.g., bewegwijzering, matrixbord, wegmarkering
- Traffic sign defects: e.g., geen zicht op, onduidelijk, missend

Rules:

- R1: Optional(Quantifiers token) + Traffic signs token + Optional(~(Traffic sign defects token) + SM token) + Traffic sign defects token
- R2: Optional(Quantifiers token) + Traffic sign defects token + Optional(~(Traffic sign token) + SM token) + Traffic sign token
- R3: Optional(Quantifiers token) + Optional(Traffic sign defects token) + Optional(erLit) + Optional(deLit) + Traffic signs token + (opLit | nabijLit | langsLit | bijLit | vlakLit | naastLit | bovenLit | inLit | halverwegeLit | overLit) + Optional(deLit) + (Road lanes token | Infrastructures token | Roads token)
- R4: Optional(Quantifiers token) + Traffic signs token + Optional(Traffic sign defects token) + (opLit | nabijLit | langsLit | bijLit | vlakLit | naastLit | bovenLit | inLit | halverwegeLit | overLit) + Optional(deLit) + (Road lanes token | Infrastructures token | Roads token)

Examples:

- E1: Verkeersbord niet duidelijk
- E2: Onduidelijk verkeersbord
- E3: Defect matrixbord boven de rechter rijbaan
- E4: Matrixbord defect boven de rechter rijbaan

24. Event Hazard Traffic Light

Goal: annotate indicators of malfunctioning or broken traffic lights
Dictionary:

- Traffic lights: e.g., stoplicht, verlichting, lantaarnpaal
- Traffic light defects: e.g., defect, brandt niet, op hol

Rules:

- R1: Optional(Quantifiers token) + Traffic lights token + Optional(~(Traffic light defects token) + Arbitrary token) + Traffic light defects token
- R2: Optional(Quantifiers token) + Traffic light defects token + Optional(~(Traffic lights token) + Arbitrary token) + Traffic light token
- R3: Optional(Quantifiers token) + Traffic lights token + Optional(~(Traffic light defects token) + Arbitrary token) + Traffic light defects token + (opLit | nabijLit | langsLit | bijLit | vlakLit | naastLit | bovenLit | inLit | halverwegeLit | overLit) + Optional(deLit) + (Road lanes token | Infrastructures token | Roads token)
- R4: Optional(Quantifiers token) + Traffic light defects token + Optional(~(Traffic light token) + Arbitrary token) + Traffic light token + (opLit | nabijLit | langsLit | bijLit | vlakLit | naastLit | bovenLit | inLit | halverwegeLit | overLit) + Optional(deLit) + (Road lanes token | Infrastructures token | Roads token)

Examples:

- E1: Stoplicht op hol
- E2: Defect verkeerslicht
- E3: Lantaarnpalen branden niet langs de rechter baan A10
- E4: Defecte straatverlichting nabij uitvoegstrook richting Middelburg A58.

25. Event Hazard Weather

Goal: annotate indicators of bad weather conditions.
Dictionary:

- Weather types: e.g., bliksem, hagel, mistbank, windhozen
- Weather vision: e.g., beperkt zicht, slecht zicht
- Weather adjectives: e.g., beperkt, felle, laagstaande

Rules:

- R1: Optional(vanwegeLit | metLit | doorLit | tijdensLit) + Optional(deLit) + Optional(Weather adjectives token) + (Weather types token | Weather vision token)
- R2: Optional(vanwegeLit | metLit | doorLit | tijdensLit) + Optional(deLit) + Optional(Weather adjectives token) + (weerLit)
- R3: R1 + (opLit | nabijLit | langsLit | bijLit | vlakLit| naastLit | bovenLit |inLit | overLit) + Optional(deLit) + (Road lanes token | Infrastructures token | Roads token)
- R4: (Weather types token | Weather vision token) + (opLit | nabijLit | langsLit | bijLit | vlakLit| naastLit | bovenLit |inLit | overLit) + Optional(deLit) + (Road lanes token | Infrastructures token | Roads token)

Examples:

- E1: Harde windstoten zorgen voor gevaarlijke situaties op de A5.
- E2: Belabberd weer in Zeeland verstoort het verkeer op de A58.
- E3: Vanwege heftige ijzel op de vluchtstrook is deze afgesloten.
- E3: Ijs op de rechter baan richting knooppunt Amstel.

26. Event Hazard Stopped Vehicle

Goal: annotate indicators of stopped vehicles due to breakdown.
Dictionary:

- Stopped car verbs: e.g., stopt, staat stil, tot stilstand
- Stopped car causes: e.g., klapband, motorpech, lege tank

Rules:

- R0: Optional(Vehicle color token) + (Vehicle token | Vehicle brand token)
- R1: Optional(Quantifiers token) + R0 + metLit + Stopped car causes token+ Optional((opLit | nabijLit | langsLit | bijLit | vlakLit| naastLit | inLit) + Optional(deLit) + (Road lanes token | Infrastructures token | Roads token))
- R2: Optional(Quantifiers token) + Stopped car verbs token + R0 + (vanwegeLit | ivmLit | doorLit | tgvLit | alsgevolgvanLit | metalsgevolgLit | naLit | metLit) + Stopped car causes token + Optional((opLit | nabijLit | langsLit | bijLit | vlakLit| naastLit | inLit) + Optional((opLit | nabijLit | langsLit | bijLit | vlakLit| naastLit | inLit) + Optional(deLit) + (Road lanes token | Infrastructures token | Roads token))
- R3: Optional(Quantifiers token) + R0 + Stopped car verbs + (vanwegeLit | ivmLit | doorLit | tgvLit | alsgevolgvanLit | metalsgevolgLit | naLit | metLit) + Stopped car causes token + Optional((opLit | nabijLit | langsLit | bijLit | vlakLit| naastLit | inLit) + Optional(deLit) + (Road lanes token | Infrastructures token | Roads token)
- R4: Stopped car causes token

Examples:

- E1: Meerdere auto's met pech.
- E2: Stilstaande auto met rookontwikkeling op de vluchtstrook.
- E3: Auto staat stil door klapband.
- E4: Pechgevalletje op de A10 richting Amsterdam.

27. Event Hazard Roadwork

Goal: annotate indicators of unplanned roadwork activities.

Dictionary:

- Roadwork: e.g., opruimingswerkzaamheden, spoedreparatie, onderzoek

Rules:

- R1: Optional(vanwegeLit | ivmLit | doorLit | tgvLit | alsgevolgvanLit |alsgevolgvanLit | metalsgevolgvanLit | naLit) + Optional(~Roadwork token + Arbitrary token) + Roadwork token + Optional(aanLit | vanLit | opLit | nabijLit | langsLit | bijLit | vlakLit| naastLit) + Optional(~(Vehicle names token | Vehicle brands token | Road lanes token | Infrastructures token) + Arbitrary token) + (Vehicles names token | Vehicle brands token | Objects token |Road lanes token | Infrastructures token | Roads token)
- R2: Roadwork token + Optional(aanLit | vanLit) + Optional(~(Vehicles names token | Vehicle brands token | Objects token |Road lanes token | Infrastructures token) + Arbitrary token) + (Vehicles names token | Vehicle brands token | Objects token |Road lanes token | Infrastructures token | Roads token)
- R3: Roadwork token

Examples:

- E1: Vanwege spoedreparatie aan het wegdek
- E2: Opruimingswerkzaamheden van brokstukken.
- E3: Spoedreparatie knooppunt Amstel hou rekening met je snelheid.

28. Event Hazard Object

Goal: annotate indicators of foreign objects and road debris that could cause dangerous situations.

Dictionary:

- Object adjectives: e.g., loshangend, kapotte, omgewaaide

Rules:

- R1: Optional(Quantifiers token) + Object adjectives token + Optional(erLit) + Optional(deLit) + Objects token
- R2: Optional(Quantifiers token) + Objects token + Object adjectives token
- R3: Optional(Quantifiers token) + Optional(Object adjectives token) + Optional(erLit) + Optional(deLit) + Objects token + (opLit | nabijLit | langsLit | bijLit | vlakLit| naastLit | bovenLit | inLit | halverwegeLit | overLit) + Optional(deLit) + (Road lanes token | Infrastructures token | Roads token))
- R4: Optional(Quantifiers token) + Objects token + Optional(Object adjectives token) + (opLit | nabijLit | langsLit | bijLit | vlakLit| naastLit | bovenLit | inLit | halverwegeLit | overLit) + Optional(deLit) + (Road lanes token | Infrastructures token | Roads token))

Examples:

- E1: Gevaarlijke restanten worden momenteel opgeruimd bij afslag Delft.
- E2: Boom omgevallen pas op afslag Goes #A58
- E3: Honderden punaises op het wegdek richting knooppunt Amstel.
- E4: Boom omgewaaid op de rechterbaan A10.

29. Event Hazard Animal

Goal: annotate indicators animals or roadkill on the road.

Dictionary:

- Animal adjectives: e.g., overreden, loslopend, overstekende

Rules:

- R1: Optional(Quantifiers token) + Animals adjectives token + Optional(~(Animals token) + Arbitrary token) + Animals token
- R2: Optional(Quantifiers token) + Animals token + Animals adjectives token
- R3: Optional(Quantifiers token) + Animals adjectives token + Optional(Animals token) + (opLit | nabijLit | langsLit | bijLit | vlakLit | naastLit | inLit | halverwegeLit) + Optional(deLit) + (Road lanes token | Infrastructures token | Roads token)
- R4: Optional(Quantifiers token) + Animals token + Optional(Animals adjectives token) + (opLit | nabijLit | langsLit | bijLit | vlakLit | naastLit | inLit | halverwegeLit) + Optional(deLit) + (Road lanes token | Infrastructures token | Roads token)

Examples:

- E1: Er ligt een aangereden gans.
- E2: Gans aangereden richting Amsterdam.
- E3: Meerdere aangereden ganzen op de vluchtstrook.
- E4: Hond loslopend op in de berm.

30. Event Hazard Roadcondition

Goal: annotate indicators of a road condition hazard event.

Dictionary:

- Road conditions: e.g., spoorvorming, aquaplaning, staand water
- Road condition adjectives: e.g., beschadigd, gaten, gevaarlijk

Rules:

- R1: Road condition adjectives token + Optional(inLit | opLit) + Optional(deLit) + (Road lanes token | Infrastructures token | Roads token)
- R2: Optional(deLit) + (Road lanes token | Infrastructures token | Roads token) + Optional(metLit) + Optional(~(Road conditions) + Adjective token) + (Road conditions token | Road condition adjectives token)

Examples:

- E1: Gat in wegdek #A5 li 13.24 afrit Middelburg
- E2: Vluchtstrook met veel staand water thv hmp 12.3