UNIVERSITY OF TWENTE.

**Faculty of Electrical Engineering,
Mathematics & Computer Science**

Detection and tracking of events using open source data

Jordy M. van der Zwan

M.Sc. Thesis
December 2020

Supervisors:
dr.ir. M. van Keulen
dr. M. Theune

Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Detection and tracking of events using open source data

Jordy M. van der Zwan

Abstract

This research focuses on designing a generic event detection system that uses open source data. Good performance is also a requirement to ensure that private individuals are able to use the system as well. The system must be able to detect events in real time based on messages from a message stream. To achieve this goal, we firstly explore what should be considered an event by looking at existing definitions and building our own definition based on observed components. Secondly, an overview is created of which pieces of information can be displayed to the user of the system in order to communicate the event to the user. An event detection system was designed which relies on a user defined reference model supported by Named Entity Recognition. The reference model plays a key part in the linking of keywords with the same meaning and the extraction of meaning from the messages from the message stream. The design was evaluated on both recall and precision using a Twitter datastream as the message stream. Taking into account the limitations of the available data, the design reached a peak recall of 80% and precision of 66%. The design performed sufficiently and still has potential to be improved in future work.

Contents

Chapter 1

Introduction

The world is increasingly generating an abundance of information, some of which is relevant to an user, but most of which is irrelevant. Finding the relevant information among the vast amount of data is a task that has become impossible for humans. This work focuses on detecting real world events that happen and are relevant to the user of the system.

Aims The aim of this work is firstly, to analyse the factors that need to be considered when deciding what should be considered an event in the context of event detection. The second goal is to provide an overview of how these events can be represented. Based on these answers to these questions, a light-weight generic event detection system which can be configured to be useful in multiple use cases will be designed. Although not specifically aimed at Twitter data, due to the extensive related work already done in Event Detection using Twitter data, Twitter will be used as an example in many cases.

1.1 Motivation

A tremendous amount of data is available on the Internet to everyone who wants to use it. The amount of openly available data is increasing in places like online social media such as Twitter. The users on such platforms produce enormous amounts of messages, the record being 143,199 tweets per second in August of 2013 [1]. This is a drastic difference with the daily average of 500 million tweets per day which translates to 5700 tweets per second. Among the messages about what people had for breakfast, more 'valuable' information is tweeted as well. Journalists and other people use Twitter to disseminate news about things that happen in the world. Detecting these happenings through the messages that are being sent through data

[1] https://blog.twitter.com/engineering/en_us/a/2013/new-tweets-per-second-record-and-how.html

streams like Twitter can provide a valuable source of information for a multitude of stakeholders.

When something is happening, people who are close to it can instantly tweet about what is occurring. This provides the opportunity for an almost real time presentation of happenings all around the world. Both organisations and individuals can gain valuable advantages when they are aware of these developments in the world. Only if you are aware of happenings in the world can you act on them.

The third goal to create an light-weight generic event detection system was born from a personal desire to be more aware of real world happenings without a manual search for all the related information. An automated system which is able to retrieve data and process it in such a way that current events are detected and communicated to the user in a clear manner would solve this problem for me. The hardware and financial resources that are available to me are however limited as they are for many other private individuals as well. Paying thousands of euros in order to get my hands on data streams is not possible and neither is upgrading my hardware to a ridiculous standard or renting hardware in a data center. This is the reason that the light-weight nature of the event detection system is part of the aim of this research and considered a requirement.

The second requirement of a generic event detection system originates from the same desire as described in the previous paragraph. As much as I don't want to manually search through the data, I do not want to search through hundreds of events either. Only relevant events should be provided to me, when I am looking at international conflicts, an event about the Oscars is not relevant to me and should not be shown. In addition to the efficient use of my time, the light-weight nature of the system is likely to demand a more targeted detection process as processing all the data might not be possible on limited hardware.

1.2 Use cases

Real time information about current events is useful for a wide range of stakeholders. Every stakeholder will be interested in different events for different reasons. This section will touch upon a few examples of cases in which event detection may be useful.

1.2.1 Private users

Event detection can be used by private individuals who want to remain aware of current events, perhaps in an area that is not covered often by the traditional media. This would be a powerful tool when fully developed, allowing anyone to be alerted when events happen that they are interested in. The generic nature of the event detection system gives these users the freedom to detect events that are relevant for them.

1.2.2 Natural disasters

Information during natural disasters is extremely valuable in the decision making process of crisis response. Multiple works [1]–[3] have looked at event detection in the context of a earthquake reporting system using Twitter data. The perspective of Twitter accounts being sensors in a global sensor network gives an insight into the potential wealth of information that event detection can tap into. Information of a decent quality can be vital in helping authorities prioritise where to deploy their resources and thus save lives.

1.2.3 Conflicts in the world

Multi nationals

Access to an event detection system is valuable for multi national companies that operate in volatile areas. They need to stay aware of current events to determine potential risks to their assets and employees. When conflicts create unstable environments due to violence or a changing political landscape, action might be needed to reduce risk or protect assets and people. This could be a reason to hold off on expanding into a new country or to halt operations and/or investments in a certain country.

Oil companies

The oil industry can also be heavily swayed by conflict due to a significant amount of oil being in volatile areas in the Middle East. An example of this is the unrest in the Strait of Hormuz. The interference of Iran in international waters affects international shipping lanes and therefore the prices of the materials that are transported on these lanes, such as oil. Awareness of events in this area is important for oil companies and other companies which are dependent on the resources shipped through these lanes.

1.2.4 Sports

A sports coach or team manager needs to be aware of events such as athletes being transferred or injured, a record being set, facilities being built/modified and new rules. This data can then be used by them to predict which team will be the most difficult opponent and to determine which players/athletes are most valuable. Whether teams are placed in a certain league can also influence local economies as the loss of such a status might mean the disappearance of a significant amount of visitors.

Determining who will win is also important as the winning or losing of a game might trigger the loser's fans to vandalise in the surrounding area. When these elements are detected and recorded it might help with predicting such events in the future and allow the authorities and locals to prepare for such behaviour as a timely warning might be able to be provided.

1.2.5 Journalism

A journalist who is focused on global events would greatly benefit from an overview of global events. Creating this overview of global conflicts would allow the journalist to consume more information by helping in the consolidation of data from multiple sources.

1.3 Challenges for event detection

In order to design an event detection system, a number of challenges need to be faced. This section lays out the challenges that come with these systems.

Understanding texts One of the largest challenges is that detection systems cannot understand what the incoming data. The system will look at the words but cannot understand the meaning like a human can. This means that the system requires a way to capture the meaning of a text by indicators to determine which messages refer to the same event.

Multiple meanings This task is complicated as natural language is complicated for machines with words that mean multiple things or different words that mean the same thing. The smallest difference, such as the use of synonyms, which would be easy for a human to spot, can cause an event to go by undetected by a machine.

Limited information and quality Certain sources such as Tweets are very short and lack in quality which makes it very difficult to extract valuable information. When the quality of spelling and grammar is bad, automated methods of extracting meaning become incredibly difficult.

Noise Social media sources such as Twitter do not only contain valuable information but also a lot of noise which does not pertain to a relevant real world event. Sadly, the messages do not come labelled, which adds the challenge of identifying the tweets that are valuable and filtering out the rest. Another form is noise is when a piece of information occurs multiple times from the same source or by different sources. This means that a message that is spammed is likely to be detected unless something is in place to prevent this.

Evaluation Another challenge is to evaluate the techniques that are designed. There is not a ground truth available for real world events that contains all the events that you would like to capture. There have been works that have created a corpus for event detection using Twitter data. However Twitter does not allow for tweets to be distributed which forces people to only publish the id's of the tweets. Due to tweets being deleted over time, the corpus becomes useless over time as you don't know if the deleted tweets were essential and manually checking millions of tweets is very costly [4]. Weiler et al. [4] (2019) tried to retrieve a set of 1,850,000 tweets and was only able to retrieve about 740,000. The second problem is the fact that it takes a long time to retrieve the tweet using their API which is the only legal method this thesis is aware of. According to Weiler et al. it would take roughly 69 days to retrieve the 1,850,000 tweets. Another work from 2017 could only retrieve 65.6% of a corpus which was published in 2015 [5] which shows that a corpus can degrade in a relative short time. These problems make it virtually impossible to compare event detection methods.

1.4 Research questions

Event detection is an interesting topic but the name itself provides little detail about what exactly is detected. In the context of Twitter, every tweet could be considered an event in itself. An event detection systems which detects tweets would be relatively easy to implement but that is obviously not what is meant with event detection. When talking about event detection, everyone has an intuition for what would be detect, mostly news articles come to mind as how a result would look like. As part of this research, some attention will be spent to look into what can and will be considered an event.

Now we have determined what will be considered an event in the scope of this research, a second question needs to be answered. Given an event, what do we want to know about the event in order to understand it clearly. The desired pieces of information can then inform certain design choices during the last research question. However, the extraction of these desired pieces of information will not be implemented and is left for future work.

Now we have determined what needs to be detected and what we need to know about the detected events, a method must be designed which can detect the events. As stated earlier in the introduction, the method should be usable by multiple stakeholders and not only focus on a specific area. In order to truly be able to service all stakeholders, the system should be able to be run by these stakeholders given a reasonable minimum specification.

These three clear steps in the research brings us to the following research questions:

- R.Q. 1. What is an event?
- R.Q. 2. What do we want to know of an event?
- R.Q. 3. How can we detect and track relevant events?

Research questions 1 and 2 will be able to shed light on the more conceptual questions regarding event detection where the third research question aims to design a solution.

1.5 Research method

To answer the first research question, we will start by looking at the dictionary definitions of an event as well as definitions of events in related work. The next step is to identify the basic components of an event and construct our own definition around those. We will then look at the properties of an event and how they relate to each other.

The first step to finding an answer to the second research question is an informal user study to determine what is deemed important information. We will then built on the aspects that are found in the responses and look at existing work to provide a full picture of what we want to know about an event.

The design of an event detection system will be based on elements that are identified during the answering of the first two research questions. We will then find a solution through experimentation in the environment already built during the pilot that was done prior to the project proposal. This environment takes care of the data collection and helps displaying the information to the user.

The event detection system will be evaluated based on recall using news articles and an evaluation of the quality of the detected events themselves.

1.6 Thesis overview

Defining an event The second chapter seeks to answer the first research question. This is done by looking at existing definitions of events in English dictionaries as well as in related work. The chapter also further lays out the problem of detecting events based on message streams. Lastly, the chapter describes the properties of and hierarchies around these events and provides the definition of an event that is used in the thesis.

Communicating an event The third chapter seeks to answer the second research question. This is done by looking at related work and a small informal user study in order to get an indication of what people want to know. The chapter further discusses both internal and external information as well as coding systems for events. Lastly, the chapter looks at what kind of big picture overviews of events are desirable.

Detecting and tracking an event The fourth chapter describes the design that was made in order to answer the third research question. It gives an overview of existing work in the field of event detection and lists some of the requirements of an event detection system. Lastly it lays out the design both on a global scale and more detailed.

Evaluation The fifth chapter shows the evaluation of the design that is described in chapter 4. The evaluation consists of both the recall and precision evaluation.

Discussion This chapter firstly discusses the recall and precision evaluation, mostly focused on the limitations of the methodology. Secondly, this chapter discusses the performance of the implementation of the design as well as the limitations of the design.

Conclusion The last chapter concludes provides the reader with a short problem description as well as the answers to the research questions. Lastly it also summarises the conclusion of the evaluation of the design and contains the future work.

Chapter 2

Defining an event

A fundamental step before an event detection system can be made is to determine what is and what is not an event. This chapter seeks to firstly, determine the definition of an event which will be used in the thesis. This chapter also serves to highlight the problems and choices that need to be made in regard to what should be considered an event.

Existing definitions Many definitions of events exist both in- and outside the context of event detection. The first section of this chapter examines both technical as non technical definitions of events. This is a starting point from which to pick and choose which elements constitute an event.

2.1 Existing definitions

This section looks at the existing definitions of events and breaks them down into basic elements which are to be used later.

2.1.1 Dictionary definitions

A number of dictionary definitions show how vague the boundaries of an event are defined. Three definitions were used from Cambridge, Merriam Webster and Oxford learners dictionary. They are the following:

1. **Cambridge dictionary:** anything that happens, especially something important or unusual. [6]

2. **Merriam Webster dictionary:** something that happens : occurrence [7]

3. **Oxford learners dictionary:** a thing that happens, especially something important [8]

First impression Although all three definitions feel intuitively correct and a good description of an event, they lack in clear boundaries. An event as defined by "something that happens" encapsulates every action ever undertaken by humanity as well as every tree that ever fell down in a forest.

Breaking down the definitions The first definition by Cambridge dictionary consists of three basic elements. The first element is "anything that happens" which refers to a real world occurrence of something. The second element is the importance of the occurrence and the third element is the unusualness of the occurrence. The latter two elements are however optional, they do not make an event but rather lift an event to a higher level.

A thing that happens

All dictionary definitions contain the notion of a happening which forms the basis of the definition and is the only mandatory condition for an event to exist.

A thing that doesn't happen The only things that is excluded from the definitions is anything that did not happen. False reports or descriptions of events that did not happen should thus be ignored. This poses a challenge as this means that false reports should thus be filtered from the input data stream. In order to achieve this goal, the system would need to determine whether the input data describes a real or fake event. This out of the scope of this research.

Perfect representation required When we make the assumption that we only want to detect events that happened, we need to think about how good the description of that event needs to be. Consider the event "A good meeting between Alice and Bob occurred", would the description "A good meeting occurred" be sufficient or does the description need to be more specific? A perfect representation is an unachievable goal as that would require complete and correct information to be provided to the event detection system. The question then becomes: at what point is the representation of an event sufficient to be classified as correct?

Importance

Both the definition by Cambridge dictionary as well as Oxford learners dictionary include the element of importance as an optional element of an event. They both state that the occurrence of something qualifies as an event *especially* if the occurrence is important. This suggests the existence of different degrees of events.

Unusualness

The definition by the Cambridge dictionary puts forward an element which stands out. In addition to the notion of importance being a qualifying factor it also looks at whether a happening is unusual. The use of unusualness can rule out more common events like sunrise and sunset which seems like a positive addition to the definition of an event. However, firefights in a war zone are no longer unusual after a while and would then become excluded from being an event. How unusual an event needs to be is a relative question without an objective or obvious line in the sand which can be easily drawn.

2.1.2 Definitions in related work

In addition to looking at the dictionary definitions of an event, this section looks at a number of more technical definitions from related work in the field of event detection.

1. "An event, in the context of social media, can be regarded as something of interest that occurs at a specific point in time in the real world and instigates a discussion about associated topics by social media users." [9]
2. "real world occurrence e with 1) an associated time period T_e and 2) a time-ordered stream of Twitter messages M_e, of substantial volume, discussing the occurrence and published during time T_e" [10]
3. "This occurrence is characterised by topic and time, and often associated with entities such as people and location" [11]
4. "a meaningful event usually consists of six elements, when, where, who, what, why and how" [12]
5. "Something that happens at specific time and place along with all necessary conditions and unavoidable consequences" [13]

Identifying recurring elements The elements that occur in all definitions are an occurrence of something (of interest) and a specific time (period). Other elements which can be found in one or more definitions are: a location, a reason, a method, actors, conditions, consequences and a substantial amount of messages about the event.

An occurrence of something The occurrence of something is easily recognised as the equivalent to "A thing that happens" in the dictionary definition of an event.

This element remains a vague concept and it not made concrete by any of the definitions.

Time (period) All definitions of an event from the related work contain the element of a time (period). This is a logical element as every event that has occurred, must have occurred at some point in time and for a certain duration.

Location Definitions 3, 4 and 5 include a location as an element of an event. In many cases, location provides valuable insight into an event but is this always the case? When we consider an announcement of a product, does this always have an associated location? In the case of a press conference, the location of the conference can be used. But what if the announcement was posted on the internet? In this case we either do not have an associated location or we must include the digital world and define a location as either in the real world or digital world.

Reason, method and actors In many instances of real world events, the user will be interested in who was involved in the event. Many of the events of interest will be in regard to actors interacting with each other or their environment. Secondly, the user will be interested in why they did what they did and how they did it. If there are no actors involved the user will most still be interested in why the event happened, what the cause was and how that came to be.

Conditions and consequences As named in the fifth definition, events may have conditions that need to be met in order for them to occur and can have consequences which may be events themselves. It seems a reasonable assumption that an event does have necessary conditions and unavoidable consequences, however they do not need to be detected as they are external forces that work on the event as opposed to the event itself.

Substantial amount of messages The first two definitions from the related work suggest that a substantial amount of messages is required in order for an event to exist. This research rejects that notion as will be further detailed in this chapter. This work considers an event not to be directly related to a discussion which may or may not take place.

2.2 Problem description

To help us understand the problem, we look at Figure 2.1 which introduces the three streams we are dealing with when detecting events.

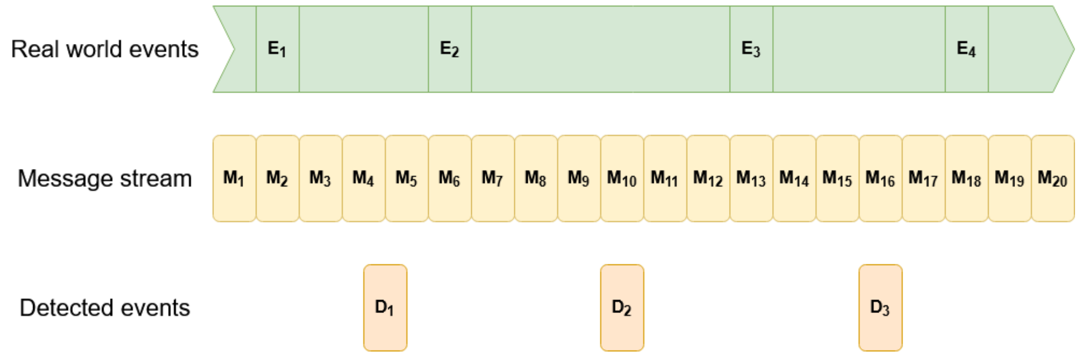

Figure 2.1: Timeline of real world events, messages and detected events.

Real world event stream The first stream, displayed in green, is the real world event stream, this stream consists of all the events that happen in the world. This is a stream with virtually infinite events, in the figure, four events are shown but the stream is much more dense and consists of many more real world events.

Message stream The second stream, displayed in yellow, is the message stream. These messages can be from any data source and in regard to anything. This message stream consists only of tweets for the purposes of this research.

Detected event stream The last stream, displayed in orange, is the detected event stream. This is the stream we are trying to create based on the messages we receive in the message stream. The position of the detected event indicates the moment at which the event was detected based on the message stream.

Goal The goal of the thesis is to create a system which can detect real world events based on a message stream that is provided. This means that we are not observing the thing we want to detect, but rather look at a stream of messages which may or may not describe observations about real world events. We can see this flow in figure 2.2. The arrows from the real world events stream to the message stream represent that the message mentions or describes the real world event. The arrows from the message stream to the detected event stream represent the detection of an event based on the messages.

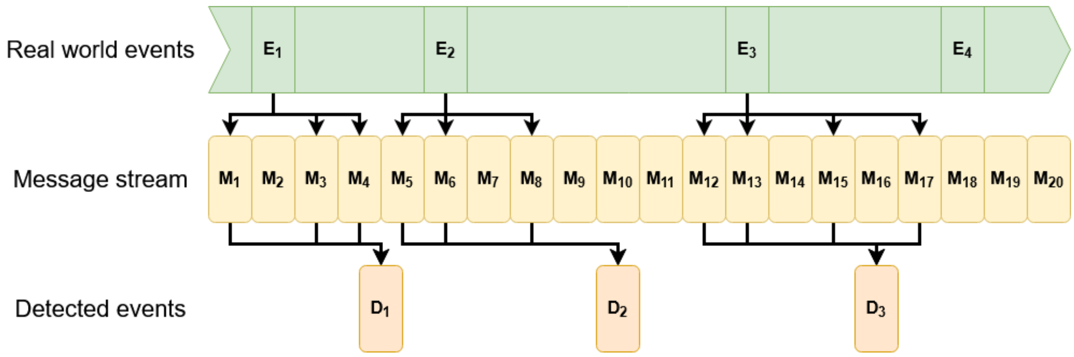

Figure 2.2: Illustration of the flow from real world event to the detection of an event.

The messy real world Although the real world event stream might look nice in a diagram, it causes a lot of complication. Firstly, it is still unclear what the boundaries of an event actually are. Secondly, we do not know what this stream looks like, there is no ground truth available for reality.

2.3 Event properties

2.3.1 Discussion events vs Real world events

When we look at the first and second existing technical definition, we observe an important distinction. There is a fundamental difference between whether the system should detect real world events or simply the discussion of a real world event. The main difference between these definitions is how such a system should be evaluated.

Real world events As described in the previous section, we have the three streams: real world event stream, message stream and detected event stream. The difference in definition lies in what stream you want to observe. In the case of real world events, you want to observe the real world events stream. However this not possible given that a software system that detects real world events always has a message or other data stream that communicates the observations made by sensors or other people.

Discussion events The alternative is to say that we want to detect the events that are in the message stream. This simplifies our problem as we limit our problem to two streams by eliminating the real world event stream. However, the assumption of this work is that the user is interested in the actual event that is being discussed as opposed to the discussion.

Evaluating the definitions When we look at evaluating discussion events, the recall will only look at the events that were available in the data. However, the user of such a system will mainly be interested in the recall of events in the real world.

2.3.2 Event detectability

When we take a second look at figure 2.2 we can see that every real world event was detected. The detection process passed from the real world event, through the message stream and produced a detected event. However, what if the real world event is not reflected in the message stream?

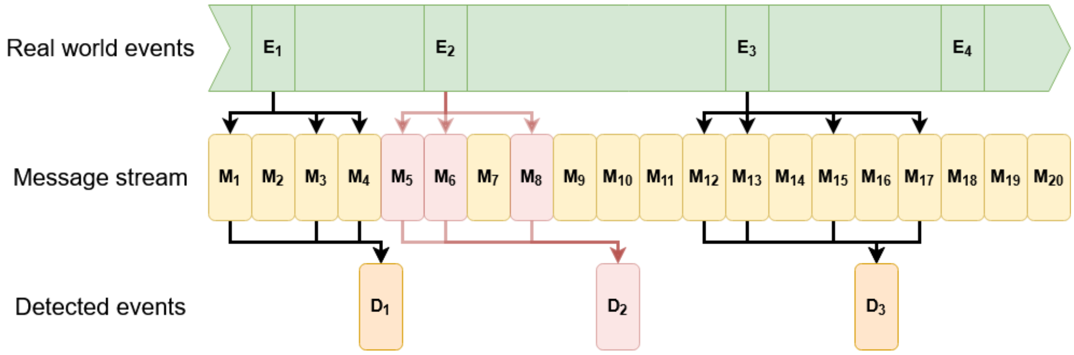

Figure 2.3: Timeline of real world events, messages and detected events with one unmentioned real world event

Event without messages Figure 2.3 shows what happens when there are no messages about an event. The diagram shows a timeline of real world events (E), messages (M) about these events and detected events (D) which represent the real world events based on the messages. It is logical that if there are no messages in your datastream, the system will not detect the real world event. The question is then raised whether this should be considered an inherent problem with the event detection system itself.

Part of evaluation In the case of an event detection system that aims to detect real world events, the event detection system has missed the event. This should thus be part of the evaluation.

Incomplete view Not all real world events will produce messages which can be picked up by the event detection system. We therefore must take into account that event detection will never provide a complete view of all real world events. Even when messages are produced about a real world event, this does not necessarily mean that an event detection will receive the messages, especially when resources are limited. An actor with unlimited resources might be able to pay to receive all the data and be able to handle the incoming data stream. However when you do not have access to unlimited resources, you will not receive all the messages which makes it even harder to detect the real world events. This thesis will only look at English messages as further explained in subsection 4.2.2. This does have further implications on what the detection system will be able to detect.

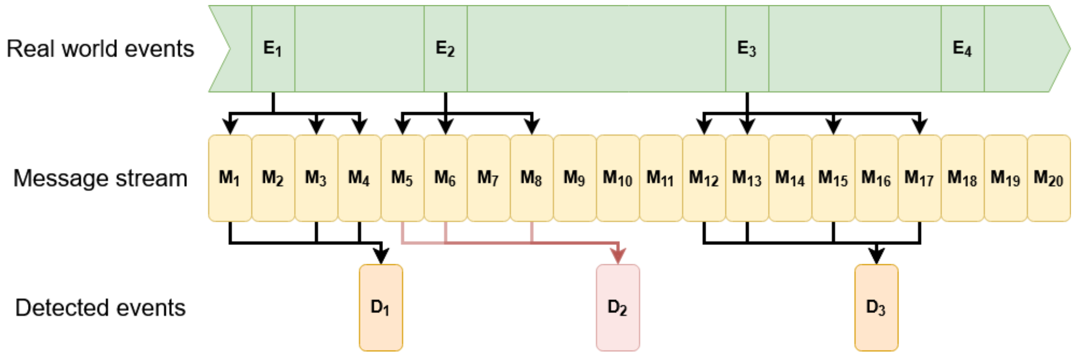

Figure 2.4: Timeline of real world events, messages and detected events with one undetected real world event

Figure 2.4 shows us a case where the event could not be detected because the detection system was unable to link the messages back to the real event and thus detect the event. This is the main problem we are trying to solve. If an event occurred and we have messages that are available to the system, the system should detect the event.

2.3.3 Reliability

When we look back at figure 2.2 we can see that the real world event is detected through the messages in the data stream. However, the event detection system only looks at the data stream as that is the only thing it can do. This does mean that we cannot check whether the real world event actually occurred.

When we look at figure 2.5, a scenario is presented where a detected event was produced while there isn't a corresponding real world event. When the goal is to detect discussions in the data stream, no problems occur since the discussion happened. However in the case of real world event detection, this poses a major problem which is not easily solved.

There is no fool proof solution for determining whether a statement is true or false. Even humans are not able to agree on what is true in normal conversation or even governments. Finding a solution to the problem of reliability is therefore out of the scope of this research as it is simply too difficult.

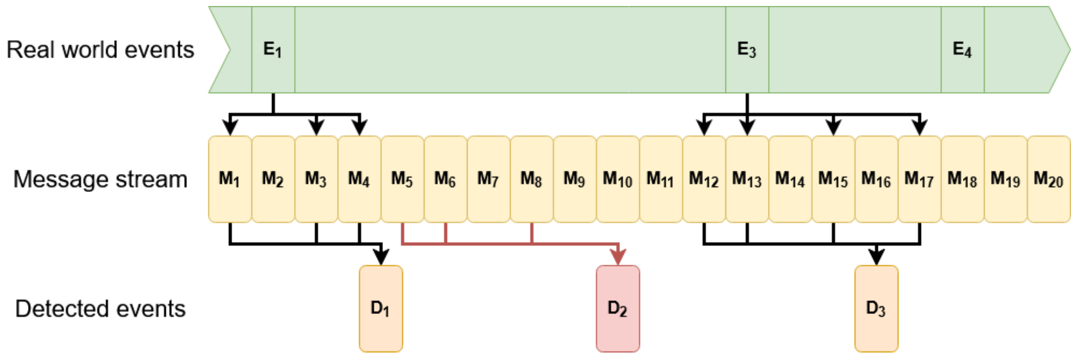

Figure 2.5: Timeline of real world events, messages and detected events with one fake real world event

An alternative to determining whether an event or statement is true or false is to communicate both perspectives to the user. However, this task is not trivial either as you would want to group multiple messages which belong to the same statement as this is important to judge the likelihood of reliability.

2.3.4 Relevance

Relevancy is dependent on the purpose for which you want to detect the events. This can be to detect the outcome of a soccer game or to detect where violence occurs. Neither of these is more relevant then the other without a purpose for which you want to detect that type of event.

If your goal is to protect employees who might be in an volatile area then every event which might endanger the employees is relevant. That does not mean that every event about conflict in the area is relevant, as long as it is not aimed at the employees. This shows how far relevancy can narrow down the events that are important to a user.

2.3.5 Burstiness

The burstiness of an event and/or keyword describes how much the discussions about or mentions of a certain event or keyword are more than normal i.e. if an event or keyword is suddenly discussed more than normal, it is a bursty event or keyword. Another word for bursty, especially in the context of social media, is trending. This property is widely used in order to identify which events and keywords are important [14] [15] [16].

Burstiness as importance In order to illustrate how burstiness is used to indicate importance in certain cases we look at an example [17]. Here, an event is defined in part as: ”a significant thing is happening when a group of people are talking about it in a magnitude that is different from normal levels of conversation about the matter, or in other words, it is trending”. The first issue that rises is the exclusion of events that are important but not discussed in a different magnitude or simply not at all. The second issue is the vagueness of the amount of discussion needed in order to qualify as trending. If a person tweets about their dinner, the discussion about that event (the dinner) has increased by a factor of infinity, yet in most cases is not a significant event. In this example, it is clear that the event is not significant as it is still only one person talking, however this becomes less clear when going from 20 messages to 50. This can be seen as bursty as the increase is significant, but is this still the case from 300 to 330? You would need to create a parameter which determines how bursty an event must be in order to qualify.

Burstiness as an indicator Although burstiness is not capable of providing an all encompassing definition for detecting events, it undoubtedly is a helpful indicator in determining certain significant happenings. Under the assumption that a significant amount of discussion and or mentions are required in order for events and/or keywords to become trending. In the case of the explosion in the Port of Beirut in Lebanon on the 4th of August 2020 [18], the event is clearly observable by looking at the trending topics on Twitter which contained the keywords ”Beirut” and ”Lebanon” [19].

2.3.6 Local and Global events

The definition of local and global hot events according to [12] is as follows: A local hot event captures the interests of a particular user community while a global hot event reflects the general focus of general users. The notion of local and global events is closely related to the question of relevance of an event. In this case the relevance determination is not made from the perspective of the user (is the event relevant to the user) but from the perspective of the event (to which users is this event relevant).

2.3.7 Completeness

Another property that should be considered is to what degree the information about an event is complete (enough). In the perfect case, a completely detected event would contain all the information about a real world event that can be known. However, this is often far from the case when trying to detect real world events.

Detection requires completeness Assuming that a detected event cannot be 100% complete, the question rises whether there is a minimum degree of completeness that is required in order for a detected event to be considered a detected event. This is in a sense a question of how specific do you need to be in order to qualify as a detected event. If we look again at the explosion in the Port of Beirut in Lebanon on the 4th of August 2020 [18], the keywords ”Beirut” and ”Lebanon” [19] could be considered to be complete enough, yet does not even mention anything about an explosion. If these two keywords were detected and classified as a detected event, should this be considered a correct classification or not?

Relevance requires completeness The second aspect that is influenced by completeness is relevance. If a detection system for people in emergency situations after a natural disaster detects the event: ”Man with broken leg in India”, the event is not relevant to authorities as it cannot be acted upon. However if the event: ”Man with broken leg in front of the Antop Hill Police Station in Mumbai, India” is relevant as it can be acted upon. This shows that an event needs to have a minimum level of completeness in order to be relevant.

2.3.8 Actionable

A major aspect of an event, especially a detected event is whether it is actionable. The real value in event detection lies in the ability of the user to learn of events when they happen so they can decide whether they need to act upon this event.

2.4 Event hierarchies

2.4.1 Detail hierarchy

An example of a detail hierarchy in the context of real world events is the example of a war. A war could be the lowest level of detail, the battles in that war the second level of detail and the firefights in those battles can be the third level of detail. Every level has smaller events but more detail.

An example of a detail hierarchy in the context of detected events is the example of a meeting. The lowest level of detail is: ”A meeting occurred”. The second level is ”A meeting between X and Y occurred”. But in the case of detected events, multiple interpretations can be reported and thus detected. Two events on the third level could be: ”A positive meeting occurred between X and Y” and ”A negative meeting occurred between X and Y”. Both events add details that the parent event does not have.

2.4.2 Causality hierarchy

When we look at the fifth related work definition: Something that happens at specific time and place along with all necessary conditions and unavoidable consequences [13], another hierarchy follows. The definition states that events have conditions and consequences. When we interpret these conditions and consequences as other events, we could create a hierarchy based on these elements.

Focus on causes One of the options in a causality hierarchy is to have the events that caused a event are considered to be the children of the event. This would allow people to dissect why an event has occurred and possibly which other events might be responsible for the occurrence of an event.

Focus on consequences Alternatively, you could consider the events that have been caused by an event as the children of the causing event. This allows for a quick overview of how large the reach of an event is.

2.5 Reframing an event

As stated in section 2.1.1 it is very hard to narrow down the notion of an happening without focusing on a specific stakeholder or case. However, instead of narrowing the definition down, it might be possible to reframe the definition in such a way that we can use it differently.

The main problem you run into when trying to determine whether something is a happening is whether it is in enough detail. Take for example a war, is this a single event or a collection of multiple events? When we then look at the battles fought in the war, should these battles be detected separately or should they always be part of the war event?

When we look at an example of a meeting between two actors, say actor X and Y. The meeting has four parts which can be identified as A: the actors arrived, B: actor X spoke, C: actor Y spoke and D: the actors left. It might be possible to detect all four parts which would result in the top representation in Figure 2.6. However, if the meeting is behind closed doors, we might only be able to say that A: the actors arrived, BC: the actors talked and D: the actors left. Lastly, in the case of a meeting, it might only be reported and acceptable to simply state that ABCD: a meeting has occurred between X and Y as displayed by the bottom representation.

It is important to note that all three representations show the same event(s) but in different levels of abstraction. In the case of a real meeting, more actors might be present and more talking will be done which cannot all be detected.

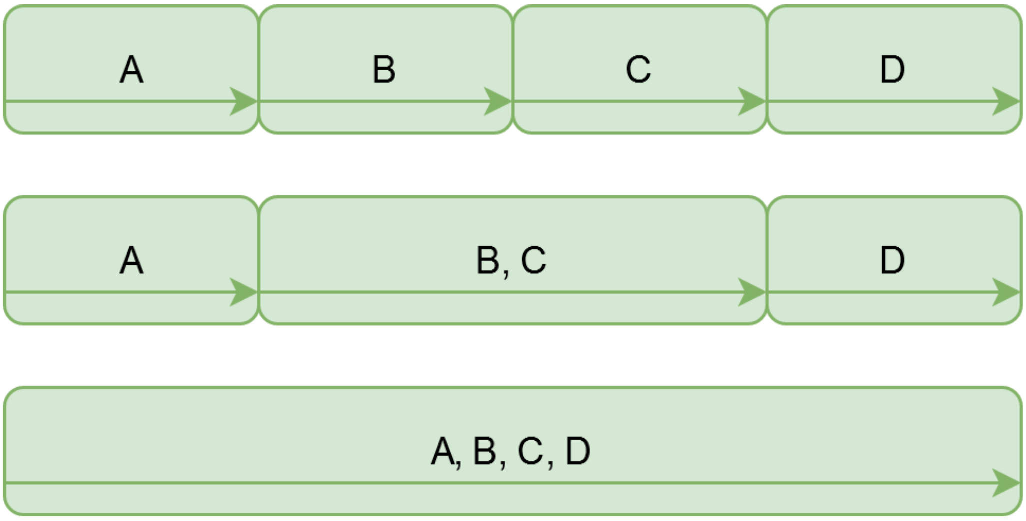

Figure 2.6: Visualisation of how events can be represented

2.6 Conclusion

This chapter has dissected definitions from both English dictionaries as well as related work in the area of event detection. Furthermore, it has laid out the difference between real world events and discussion events and discussed properties of events as well as hierarchies of events.

Event definition This work defines an event as a real world event as opposed to a discussion event. This means that the goal of the event detection system is the detection of occurrences in the real world which are relevant to the user. This will mainly affect the considerations in the evaluation of the event detection system. When we mention an event in the rest of this study, this refers to a real world event. In all other cases, it will be specified as in the case of a detected event.

Detection limits An important thing to note is that the system will not be detecting what we would like to detect, because the system detects real world events indirectly based on the message stream. We therefore need to acknowledge that we can only detect a small portion of the real world events as only a small portion will show up in the messages stream and only a subset of those events will be detected from the message stream.

Chapter 3

Communicating an event

Now we have a grip on what will be considered an event, we want to evaluate what we want to know about the events we detect. This may influence how we will detect the events and what information must be kept and what can be discarded due to resource allocation.

Goals The overall goal of this chapter is to determine what we want to know about an event. To answer this question, we need to determine why we want to know anything about the event in the first place.

Understanding the event The main goal of gathering information about an event is to make it possible for the user to understand the event that has been detected. The system would be incomplete if the output was limited to: ”An event was detected”. Although this qualifies as event detection in the strictest terms, this is useless for a user of the system.

Judging the event Once the user understands an event, it enables the user to form an opinion or other judgement. This judgement can be in regard to the reliability of the detected event, the actors involved or the event itself.

Acting on the event The ultimate goal of detecting events is the option to act on the information. If a system would detect the events but nothing happens with the information, the information is useless. The understanding and judgement should ultimately lead to a choice of whether to act on a detected event or not. This could be to evacuate based on detection of an impending natural disaster or violent conflict.

3.1 Informal user study

A small informal user study was conducted among four OSINT (Open Source INTelligence) enthusiasts and someone without an express interest in OSINT. The OSINT enthusiasts were known to be interested in OSINT through their presence in online meeting places for OSINT discussions. The other person was an acquaintance of the author.

Method They were asked via online instant messages what they would want to know about an event. This was asked without initially providing the precise context in order not to steer the response in a particular direction. However they were aware that the author was working on an event detection system. In some cases further questions were asked in order to clarify or expand their answer. An example of a clarifying question was to clarify what the person meant by saying: ”... and other information would be good”. In other cases, where the person misunderstood the question and context, these were further explained in order get an usable answer.

goal The main goal of the study was to get an impression of what people might want to know about an event. This in turn provides a starting point for determining what information should be extracted from an event and displayed to the user.

3.1.1 Results

All the pieces of information that were reported by the people who were interviewed are enumerated below. It also shows by how many individuals each of the questions were mentioned.

1. Who, what, where, when, why and how? (4x)
2. Is it fake news? (1x)
3. Is it reliable? (1x)
4. Is it accurate? (1x)
5. Is it credible? (1x)
6. Who reported the event? (2x)
7. What is the nature of the source? (Official/Unofficial) (2x)
8. Are there conflicting sources? (1x)

9. Has the event happened before? (2x)
10. Will the event happen again? (2x)
11. Has the event happened somewhere else before? (3x)
12. Has the event happened to other actors before? (2x)
13. What are the consequences of the event? (1x)
14. What will happen next? (1x)
15. What will not happen next because of the event? (1x)
16. Who benefits from the event? (1x)
17. What caused the event? (2x)
18. Who is now put at a disadvantage? (1x)
19. Why did it not happen before? (1x)
20. Why will it not happen again? (1x)
21. Is the event unique? (1x)

3.1.2 Discussion

When we look at the desired pieces of information as described in subsection 3.1.1 we can identify two main types of information that people want to know about events. These two types are internal and external information.

Internal information The first type of information is internal information of events e.g. which actors were involved. Internal information is limited to a single event and can be looked at in isolation.

External information The second type is external or contextual information e.g. has an event like this happened before or will it (likely) happen again. These pieces of information tell the user how the state of the world has caused and/or influenced the event or how the event has influenced or will influence the world.

3.2 Internal information

This section discusses more clearly what is considered internal information and lays out examples of internal information that can be useful. For certain pieces of information, a more detailed analysis of the challenges is given as well.

The key property of internal information is that it pertains to the event itself as opposed to information about relations to other events. If the information or question pertains to the real world event or the detected event, then it is considered to be internal information.

Five Ws The most obvious pieces of internal information are the five W questions. These are who, what, when, where and why. In addition to these five questions, the question how can be asked as well. These provide the basic information about the event. However they are also very open questions to answer.

3.2.1 Who?

This question pertains mainly to the actors that are involved in the event. There can be multiple actors connected to a single event which makes it difficult to assess the completeness of the list of actors.

Complexities Actors can be involved in events in a number of ways and which actors are relevant is dependent on the user and the particular event. It might be important to know who initiated an event, who was targeted, who was affected, who witnessed the event or who is responsible for the event. In the case of a conflict between to parties, you might have an initiating actor and a targeted actor. However, in the case of a natural disaster, the initiating actor does not exist.

3.2.2 What?

The question as to what has happened requires some description of the occurrence itself. This occurrence can be an action that has been taken by an actor, or another kind of happening. The main issue with this question is that it is hard to standardise the answer to this question given the desired generic nature of the event detection system.

Verbs as actions A basic form of the what question is to look at the verbs used in the messages. This works well in an example such as: ”The two trains crashed into each other on Monday morning at Utrecht Centraal”, where the question: ”what happened?" can be answered by: a crash. This is a very broad statement but would allow detected events to be compared to each other.

3.2.3 When?

The question of when can be approximated by looking at the dates of the messages that are linked to the event, but this is only an approximation. The event is not necessarily detected at the time it is happening.

Extracting time As can be seen in Figure 3.1 it is difficult to determine when a event has occurred or will occur based on the messages in the datastream. The reason for this is that the messages about an event and the actual occurrence of an event do not need to align with each other. There are multiple cases in which can be seen in the figure and are explained below.

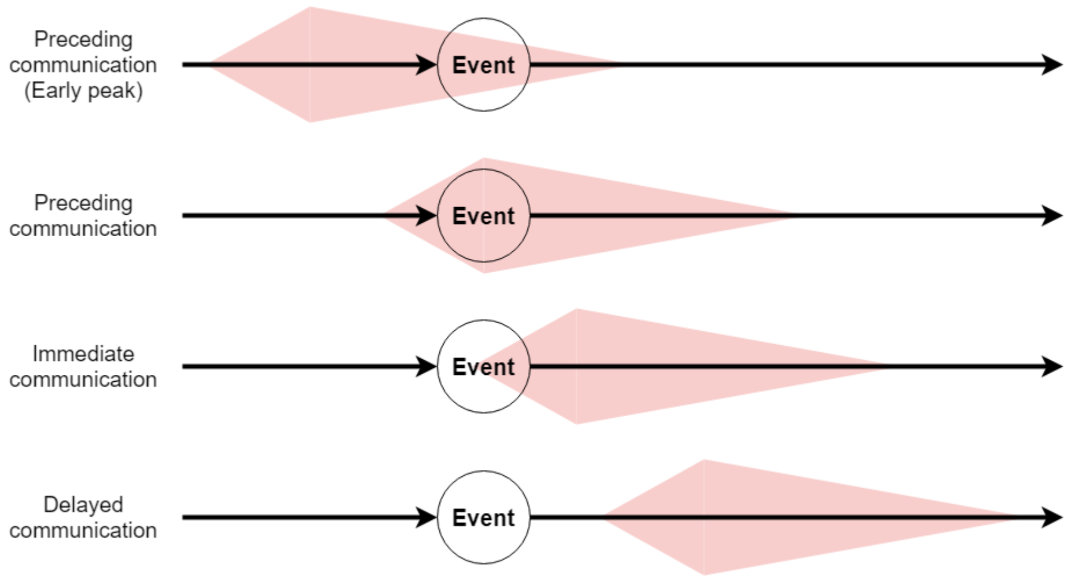

Figure 3.1: Event Message Interaction

Preceding communication Events which have been announced can occur in the data stream before the events themselves have occurred. Most messages about an event may be send even before the event has started.

Immediate communication The third case is where an unexpected event occurs e.g. a fire breaks out and messages start to occur in the data stream.

Delayed communication Another option is where an event already took place but is announced or discovered after the fact.

No guarantees The examples given in this subsection illustrate the fact that determining an exact time period in which an event occurred is a difficult task. The most obvious choice for a time is the first time you detect the event in the data stream. This is likely not the exact time of the event, but should be close enough for most purposes, especially when analysing events on larger times scales.

3.2.4 Where?

The location of an event is often important in order to make a detected event actionable. If you do not know where an event took place, it becomes harder to respond. It also provides context to an event as a firefight in Afghanistan is different than a firefight in Canada.

Physical and digital locations The where question does not necessarily result in a physical location but can also result in a digital location such as a place where an announcement has been posted.

Unavailable or irrelevant A location is not necessarily available or relevant for an event. In the case of an announcement or the release of a product, the location of the announcement is unlikely to be relevant.

3.2.5 Why?

This question looks a the motivations of the actors or the reason that an event occurred. An example of this is when an official gives reasons as to why a certain policy decision was made. Which judgements were made and why they reached that particular conclusion.

3.2.6 How?

Aside from a description of what has occurred, the user is likely to be interested into how the event happened. This can help a user decide whether an event is plausible or provide insight into how an event was able occur.

Means The how question should also shed light on what means the actors used in order to do what they did. In the case of a shooting, the type(s) of gun(s) is valuable information in order to judge and in order for the event to be actionable in the case of trying to prevent the event from repeating.

3.2.7 Reliability

In the era of big data and fake news, reliability of information becomes easily questionable. False reports, either purposefully or accidentally spread, pose a big issue in event detection. If 99% of the detected events are fake, they are all useless to the actor without further investigation.

Rumour detection Rumour detection can help in detecting whether information is suspicious. This can either be determined by comparison to external sources or the analysis of interactions with the messages e.g. replies to tweets.

Tracking the information flow Using the timestamps of the tweets themselves we can also determine where reports of events originated and whether similar language was used which may indicate that people are copying the information. This could be assisted by models of who follows who which may provide insight in the information flow.

3.2.8 Enriching internal information

Examples of enriching internal data are the lookup of coordinates based on location names. Knowing which cities certain events have occurred in is valuable in itself but can be more valuable when projected on a map. In order to do this, coordinates must be provided which can be acquired using external sources.

3.3 External information

Another interesting aspect is the notion of external information. The context in which events happen is important for the user to understand and judge the event. External information are pieces of information that do not describe the event itself but rather the relation to preceding events as well as subsequent events.

3.3.1 Relevancy through consequence

A tree falling down in a forest has little value to a stakeholder while a tree falling down on a high voltage line would spike more interest. These events differ in internal information too, as the high voltage line should be mentioned in the second event.

However, it is not necessarily the collision with the power line that is interesting to the user but rather the result of the collision. If the collision causes a huge blackout, the event becomes very relevant to the user. If the collision is on the other side of the world, it becomes much less relevant to the user. This has to do with the consequences of the event, the more it affects actors and in particular the user of the system, the more relevant it becomes to the user.

3.3.2 Other contextual questions

From the results of the informal user study, we see questions like: ”Has the event happened somewhere else before?” and ”Has the event happened to other actors before?”. These questions are interesting to users as they show whether an event is a novel development or a common occurrence. This is important information that is needed to judge the importance of the event as well as decide whether actions need to be taken.

3.4 Coding systems for events

Multiple event coding systems have been devised in order to analyse real world events. The main purpose of these systems is to standardise the representation of real world events.

Existing coding systems Examples of existing coding systems are: the World Event/Interaction Survey (WEIS), The Codebook of the Conflict and Peace Data Bank (COPDAB), the Integrated Database for Event Analysis (IDEA), the Protocol for the Analysis of Nonviolant Direct Action (PANDA), the Behavioral Correlates of War (BCOW) and the Conflict and Mediation Event Observations (CAMEO) [20]. Where IDEA, PANDA and BCOW were extensions on WEIS and COPDAB, CAMEO is a seperate framework that set out to solve issues with previous coding systems.

Origin and purpose The main use of these coding systems is to analyse global conflict and cooperation events such as state actors interacting on a geo-political scale. Both WEIS and COPDAB were developed during the Cold War and were aimed at sovereign states interacting through diplomacy and military threats. These systems are not perfect and still leave a level ambiguity instead of absolutely clear borders. Improvements such as the addition of sub-state actors are proposed in works that expand on these coding systems [21].

CAMEO structure As CAMEO is the most recent system, we take a look at how CAMEO creates an encoding for events. The main structure of these codes is roughly: ”Actor Verb Actor” where the first actor is the initiator and the second actor is the target. The verb describes an action such as: an appeal, a consultation, an investigation or demand. Both the actors and verbs exist in a hierarchy where the actor or action becomes more defined further down the hierarchy.

3.4.1 Examples

CAMEO Actor example An example of an actor code is the code: ”NGOHRIAMN” which represents Amnesty International [22]. These codes consist of multiple three letter combinations and narrow down. The example consists of three codes: NGO, HRI and AMN which stand for Non-governmental organizations, Human Rights and the specific code for Amnesty International respectively.

CAMEO Verb example An example of a verb hierarchy has as root the verb: 170 (Coerce, not specified below). This has two more defined verbs namely 171 (Seize or damage property, not specified below) and 172 (Impose administrative sanctions, not specified below). Lastly 171 has two more defined verbs, namely 1711 (Confiscate property) and 1712 (destroy property). It should be noticed that the first two numbers represent one of the 20 base verbs and all subsequent numbers represent a more specific verb.

3.4.2 Drawbacks

It is important to note that these systems lose information about the event itself. It is not feasible nor desirable to create unique codes for every actor or action. The main purpose of these systems is to be able to analyse how often certain events occur in the world. When these coding systems are applied to huge amounts of events it becomes possible to detect patterns. This can be used to see whether certain actions occur more or less often over time which can be narrowed down by actor as well. A very useful tool in order to analyse the big picture.

3.5 Communicating the big picture

An insight from the informal user study is that not only the events themselves are seen as valuable, but also the patterns that these events create.

Context is key Most real world events do not occur in a vacuum. They often are a consequence of, or in response to one or more previous events. Showing these dependencies provides a broader insight into the question why an event has occurred.

3.5.1 Temporal

As discussed in paragraph **??**, determining correct timestamps for events is not a trivial task.

Detecting escalation Temporal data in events is vital in order to be able to detect whether events are escalating. When more and more riots start taking place, this is a valuable warning. However, this requires reliable temporal data about the events themselves. When this is not available, false positives can occur. When a riot sparks discussions of past riots, the system may think they are happening currently and falsely detect an escalating amount of riot when only one occurred.

3.5.2 Spatial

Spatial information about events is very useful in order to determine whether events are limited to certain regions or wide spread problems.

Detecting hotspots Spatial data can also be used to determine hotspots of activity. In the case of tracking conflicts, detecting hotspots can provide information needed to decide whether to avoid an area.

Detecting spreading When hotspots are detected, we can also track whether these hotspots are getting bigger or moving. This provides actionable information which can be used to prepare for whatever event type might be spreading.

3.5.3 Causal

Causal information of events can be valuable in order to determine whether certain event types always result in a particular response from a certain actor. When a lot of events have been detected long term, patterns could be found where certain events or event types often precede another event (type).

3.5.4 Towards prediction

When we look at spreading and escalation of events over a long period of time, we can start making predictions about what will happen next based on previous patterns. These predictions will still be vague but historic patterns can give insights into the future.

3.6 Conclusion

This chapter has described how the communication of an event is important for a user to understand, judge and act on a detected event.

Informal user study Although the user study was small, it did provide a starting point for the overview of pieces of information that can be communicated and are desired. It shows that aside from information about a single event, there is a desire to understand the event in a broader context.

information For both internal and external information, an overview was given with the possible pieces of information where challenges were highlighted as well.

Coding systems An alternative method of representing events was discussed as well based on related work. The origin and use of coding systems was discussed as an option for representing events.

Communicating the big picture Given that there is a desire to understand the detected events in a broader context, a number of examples were given in which a bigger picture is painted for the user. These overviews are based on multiple events and can show larger trends over time.

Chapter 4

Detecting and tracking an event

Section 4.1 discusses the related work where both more abstract considerations regarding event detection will be discussed as well as specific implementation examples will be discussed. Section 4.2 lays out all the requirements that a real world event detection system should meet. Section 4.3 provides a global overview of the method that has been implemented and evaluated in this thesis. Section 4.4 expands further on the global design and provides more detailed insight into the method as well as insight into why certain choices were made.

4.1 Related work

Event detection techniques are generally categorised in two main categories which are not entirely consistent over all related work but are roughly as described in this section.

Feature pivot Techniques that are feature pivot are considered to be temporal based [23] [24], to involve ”grouping entities within documents according to their distributions” [17] or ”computing the co-occurrence patterns between pairs of terms selected among different documents” [25].

Document pivot Techniques that are document pivot are considered to rely on tweet/document features [23] [24], consist of ”clustering on documents based on their semantic distance” [17] or involve ”create groups of documents according to a specific document representation and some document-to-document or document-to-cluster similarity measures” [25].

4.1.1 Deterministic vs Indeterministic

When using keywords in order to cluster tweets together, a choice has to be made whether to have a pre-defined set of keywords which will be used during the event detection process or to have let the system determine which keywords should be used based on for example burstiness. When burstiness is used to determine keywords, a string will be considered a keyword if it occurs more often than it is normally occurs. When a predefined set is used, the technique is deterministic. If this is not the case, it is considered an indeterministic technique [26].

External data as keywords An example of a deterministic event detection system is [27]. In this case, Wikipedia titles are used as keywords in order to create clusters of tweets which represent a detected event. Another example of using external information is [28] which uses data from DBpedia and WordNet as keywords in order to detect events.

Bursty event detection Detecting bursty events is inherently an indeterministic method as you do not know which events/keywords will be mentioned more than usual.

4.1.2 Named Entity Recognition

The main purpose of Named entity recognition is to identify entities that are in the texts. These entities can then be used as keywords and used for event detection. One of the advantages of doing this is that it can combine multiple words into a single entity. Another use is that it also identifies the type of entity (person/location/organisation) which allows for automated enhancement information e.g. looking up coordinates of locations or biographies of people.

Conventional NER solutions Stanford CoreNLP is one of the most used NLP toolkits [29] and is often used in event detection systems when Named Entity Recognition is used in the design of the event detection system. Apache OpenNLP also provides Named Entity Recognition [30] but is not often used in related work.

TwiNER Both StanfordNLP and Apache OpenNLP are not targeted specifically towards Twitter. The short nature of tweets does however bring its own set of challenges to the problem of Named Entity Recognition. TwiNER tries to create an unsupervised model which is aimed towards NER in tweets and shows promising results but needs more work [31].

TwitterNEED Another option for Named Entity Recognition is TwitterNEED [32] which is also targeted specifically towards tweets. This method uses Named Entity Disambiguation in combination with Named Entity Extraction in order to improve the extraction process. This method also uses external data from YAGO Knowledge Base and Google searches.

4.1.3 Filtering of messages

A method used to filter irrelevant messages as proposed by [14] uses a Random Forest classifier which is pre-trained and then used to filter out irrelevant messages. The following features were used in the classifier:

- The number of words
- The ratio of capitalized words
- The number of name entities
- The ratio of hashtag words
- The ratio of mentioned users
- The ratio of non-English words
- The ratio of opinion words
- The ratio of personal sentiment words
- The number of question expression word

The idea behind this filtering method is the fact that certain types of spam tweets have a quite similar features. One of these features that was also encountered during this study is the mentioning of many users when spamming messages. If these messages are filtered out quickly, this can help boost performance as more expensive tasks do not have to be executed for every irrelevant message.

4.2 Requirements

Given the goal of a light-weight generic event detection system, a number of requirements are set based on this goal. This section will enumerate, describe and justify these requirements.

4.2.1 Functional requirements

This subsection discusses the basic functional requirements of the event detection system. This list contains all the requirements you want in a completely worked out system. These requirements are not all worked into the prototype that has been evaluated in the thesis. Some requirements might be impossible to fully implement, but serve as a goal to aim for.

Detect real-world events The system should detect all real-world events that are relevant to the user. This means that an event should be detected no matter how much it was discussed. This requirement is impossible to achieve as you cannot detect what you don't see, however this should be seen as an inherent failure of the system as opposed to moving the goalposts and only considering events that are discussed. The detected events should be actual events that happened, not false reports.

Track real-world events The system should add new messages about an already detected event to the detected event.

Automatic learning The system should pick up locations, organisations, people and other entities that have not been manually inputted when they occur a significant amount in the data stream. Although the initial setup of the detection system is done by the user and thus deterministic, the system should pick up new information.

Reliability The system should determine whether an event is reliable.

Perspectives The system should identify and show all different perspectives on an event when multiple are available. An example is to show both the progressive and conservative arguments used in a political discussion.

Generic data structure The design should not use elements unique to or focus solely on a single source of data. The main concern when using only a single source of data is that the data may become unavailable due to any number of reasons.

A current trend is that open source data is disappearing more and more as companies and organisations are becoming more strict in data security. Large scandals such as the incident with Cambridge Analytica cause companies to be careful with the data that they share which is the responsible course of action. However the downside for actors that rely on open-source data is that they lose access to sources which have decided to clamp down on their data security.

Another reason that data may become unavailable is that the provider of the data no longer wants or is able to provide the data. This can happen when companies switch to selling the data instead of giving it away for free. It can also happen that a company goes out of business and therefore is no longer able to provide the data.

Due to these concerns, the event detection system should not be tailored specifically to Twitter data but be usable when other data is provided as well.

4.2.2 Performance requirements

The aim of the research is to build a light system which can be run by most people or at least be able to run on a system that most people could afford if they were interested in event detection. This excludes the use of huge models which require a lot of memory.

Due to the limitation in resources, we also need to consider other limitations such as an inability to pay for translation services. Some event detection systems process multiple languages, this will not be possible during this research. The scope of this research is therefore limited to message streams that are only in English.

4.3 Global design

In this section, the main workings and components of the event detection system are named and explained. At the core of the system is the reference model which is used to translate a message into the meaning of the event. We will firstly look at the structure of the reference model and then show how it is used in the event detection system.

Introducing reference items A reference item represents a single entity or action. It consists of a set of strings that represent the single entity or action. An example of a single reference item is one that represents the country: United States of America. In this example, the reference item would be a set that contains the following strings: ”United States”, ”U.S.”, ”USA” and ”United States of America” as well as more strings that represent the country.

Reference item matching When one of these strings occur in a message, the reference item will be linked to this message. Due to this, the system is able to link the abstract concept of the country: ”United States of America” to every message that contains any of the strings in the reference item. Figure 4.1 illustrates how the

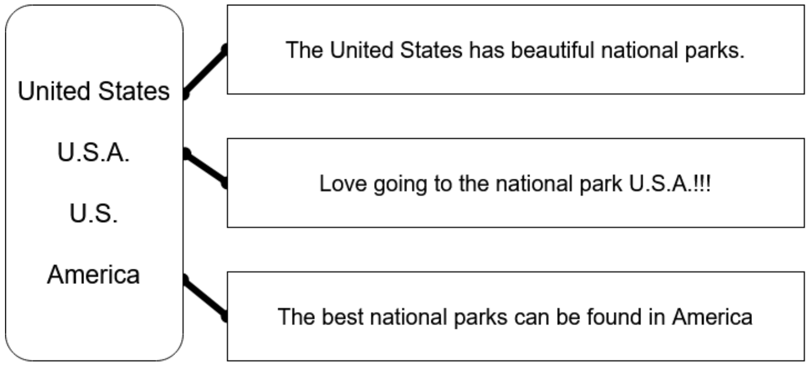

Figure 4.1: Example of a reference item linking multiple message to the country: United States of America.

system is able to link messages to the country: ”United States of America” while they do not share the same keyword.

Reference model All the entities and actions that are relevant to the user must be captured in a reference item. If this is not the case, the system will not detect these entities and actions in the messages. The collection of all the reference items is called the reference model. Now we can formally define the reference item and reference model:

Definition 4.3.1 (Reference item)**.** A reference item is a set of strings which all represent the same real-world entity.

Definition 4.3.2 (Reference model)**.** A reference model is a collection of reference items.

4.3.1 Single message life-cycle

We will now look at the life-cycle of a single message in context of the event detection system. Only a subset of the steps and conditions are given in this example, further explanation can be found in the detailed design in section 4.4. This example assumes that the message concerns an actual event.

Message creation The first step is that the message is created in some form. In our case, a message is tweeted that describes an event. We will now look at the following message as an example: ”The United States has send aid to Beirut after the explosion”.

Message capture Secondly, the message has to be captured by the system. A message will be captured when there is an overlap between the keywords the system is looking for and the content of the message. If a lot of messages are in the stream, the system will only receive a subset due to API limitations.

Reference model matching The third step is to match the message against the reference model. This step results in a set of reference items that have been linked to the message. In the example, the following reference items would be detected: 'United States of America', 'Aid', 'Beirut' and 'Explosion'. The number of linked reference items must be equal to or greater than a configurable threshold as to prevent all messages that are only linked to a single reference item such as: 'United States of America' to be put in a cluster. If the message does not match enough reference items, the message is ignored on the basis that it does not contain enough (relevant) information.

Clustering The fourth step is to put all the messages with identical reference item sets into a cluster. In this example, the message would be put in the same cluster as the message: "Beirut has been sent aid by the USA after a large explosion" as both messages will link to the same reference items: {'United States of America', 'Aid', 'Beirut', 'Explosion'}. The assumption being that when the reference item sets are identical, the meaning can be assumed to be identical as well. If the number of messages in a cluster reaches or exceeds a configurable threshold, the cluster is considered a detected event.

4.3.2 Detected event example

When we look at the messages: "The United States has send aid to Beirut after the explosion" and "Beirut has been sent aid by the USA after a large explosion", the following reference items would be linked: 'United States of America', 'Aid', 'Beirut' and 'Explosion'. Assuming we only need 2 messages in order for a cluster to qualify as a detected event, we have detected the event that is represented by the reference item set: {'United States of America', 'Aid', 'Beirut', 'Explosion'}. The event is thus represented by the reference item set they have in common and the messages themselves.

4.4 Detailed design

This section contains a detailed description of all the steps in the event detection system including the steps that were not discussed in the global design. The steps can be seen in Figure 4.2. This also shows a step that was left out in the global design: entity extraction. This step tries to detect entities in the incoming messages and add them to the reference model if they occur a significant amount. This is further explained in subsection 4.4.3. We start this section with a deeper explanation of the purpose of the reference model.

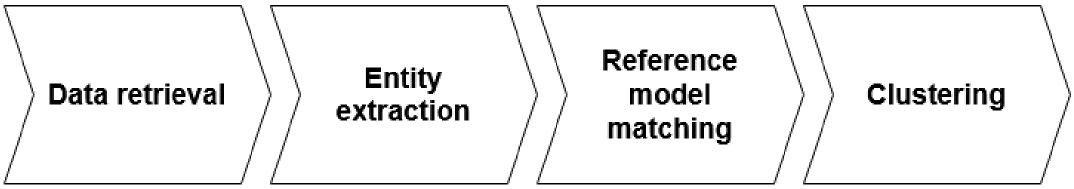

Figure 4.2: Event Detection System Overview

4.4.1 Reference model

This subsection explains in more detail why the reference item was used and introduces the concept of reference item types.

Modelling relevance In order to satisfy the requirement of detecting events that are relevant to the user, the user must be able to state what they consider relevant. The user can do this by creating a collection of relevant keywords which can represent any entity or action the user deems relevant. Because the system only looks at linked reference items for the clustering, only messages with relevant entities and actions will result in a detected event. This should thus result in detected events which are relevant as well.

Connecting the dots As the user needs to create this list of relevant keywords, we take this opportunity to have the user do more of the work in advance. Instead of a simple list with many keywords, the user will create a collection of reference items which contain multiple keywords that refer to a single entity. An example of a reference item that was given previously is: {"United States of America", "United States", "USA", "America"}, these keywords all refer to a single entity. This is more valuable than a flat list of keywords, some of which might refer to the same entity. This is because it allows for messages that do not have any keyword in common, but refer to the same entities, to be recognised as similar by the system.

Introducing reference item types Every reference item is also given a type i.e. person, organisation, country, etc. which is used to organise the reference model. These types can be linked to one of the six 5W1H questions (Who, What, Where, When, Why and How). Using the types, we can display the answers to these questions on the screen. An example would be the reference item: ”Donald Trump” which would be of type: ”person”. He could then be put as (part of) the answer to the question: ”who?”.

Used reference item types The types that were used during the research are: ”activities”, ”locations”, ”countries”, ”organisations”, ”people”, ”professions” and ”keywords”. The activities type captures what has happened such as: ”Aid”. The locations and country type captures where the event has occurred. The country, organisation, person and profession types show which actors were involved in a detected event. The keyword type was used to represent multiple objects such as equipment like tear gas or pepper spray which can be important to an event.

Reference item type ambiguity As stated in the previous paragraph, a country can be used as a location and as an actor. This type can thus not be linked to only: ”who?” or ”where?” but should be linked to either or both questions based on the context.

Reference model creation The reference model must be created before event detection can begin. With an empty reference model, nothing will be detected by the system as all messages are seen as irrelevant to the user. It then becomes an iterative process of increasing the reference model by adding reference items manually and through the automated entity detection which will add to the reference model as well.

4.4.2 Data retrieval

The initial step in the event detection system is the data retrieval. The system retrieves Tweets in real time using the Twitter Streaming API via the Twitter4J library. This is a restricted stream and will only provide a fragment of the actual data stream.

Filtering the stream A number of keywords can be provided in order to filter the stream. It is also possible to leave it empty which will generate a random sample of all the tweets but the filter helps to provide more relevant messages. This is likely undesirable (depending on what the user wants to detect) as this will introduce a lot of noise whereas a filtered stream would contain much less noise.

Other data sources The data retrieval in the implementation used in this research only uses the Twitter API by using the Twitter4J library. However the other steps in the event detection system could also accept other data sources. Properties that a single message must have are: a source, unique identifier and the actual content of the message.

4.4.3 Entity extraction

In addition to the manual creation of the reference model, entity extraction provides assistance by detecting people, locations and organisations.

Named Entity Recognition Although actions such as ”exploded”, ”attacked” and ”won” can be thought of in advance, the people, organisations and locations involved are more difficult if not impossible to predict. Named Entity Recognition is therefore applied on the incoming tweets in order to update the reference model with people, organisations and locations which occur often in the data stream.

False positives Both libraries had significant trouble with the detection of entities. Certain words such as: ”Breaking” would be detected as a location, which would cause trouble as it would match to many messages.

Entity extraction occurrence threshold In order to reduce the amount of false positives that would be added as reference items to the reference model, a minimum occurrence threshold was added. This technique does not filter out 100% of the false positives but reduced it significantly. as they would be accepted if multiple tweets contained the same string which would create the false positive.

Used libraries Two Named Entity Recognition libraries (Stanford NLP and Apache OpenNLP) were considered for the purpose of this research. These were chosen as they are both well established and already had implemented libraries which could be used in the implementation of the event detection system.

Quality vs Performance Apache OpenNLP was chosen as it uses significantly less memory and is also much faster. The speed was the main factor as it was needed to keep up with the amount of incoming data which was not possible with Stanford NLP. Stanford NLP did seem to have less false positives, but the speed of OpenNLP took priority.

4.4.4 Reference item matching

During the reference model matching, the system tries to link the messages to reference items in the reference model. If a piece of text can be matched to multiple reference items, the largest entity is chosen. An example for this is in the case of the text: "The Federal Bureau of Investigation concluded their inquiry.". Both the keyword 'Investigation' and 'Federal Bureau of Investigation' can be matched to the text. In that case 'Federal Bureau of Investigation' will be matched and 'Investigation' will not.

4.4.5 Clustering

Once a message has been matched to the reference model, the system will determine whether it belongs to a existing cluster, a new cluster or a detected event. This is done based on a number of parameters given below:

- **Minimum number of reference items:** how many unique reference items are needed in a set in order to be detected as an event.
- **Minimum number of messages:** how many messages are needed in a cluster in order to qualify as a detected event.
- **Minimum number of unique sources:** how many unique sources are needed in a cluster in order to qualify as a detected event.

Minimum number of reference items A message will only be used if it has matched with a minimum amount of reference items from the reference model. If this is not the case, the message is not processed further as it is deemed not to be relevant. The message either does not contain enough (relevant) information and can therefore not match with enough reference items. Another reason can be that the reference model is incomplete.

Minimum number of messages and unique sources All messages with the same set of reference items are then clustered together. If a cluster contains enough messages from enough unique sources it is considered to be an event.

Hierarchy After the clustering, a hierarchy in clusters is established. Every cluster has a set of reference items which was used to create the cluster. A cluster where the set of reference items is a subset of the set of reference items from another cluster will be assigned as the parent. This is a hierarchy of detail as described in subsection 2.4 and shows easily whether more detailed versions of the event exist.

4.5 Summary

This chapter has provided the reader with an overview of some of the related work in regard to event detection. The related work was namely focused on the use and detection of keywords that can be used in event detection.

Requirements Secondly, the chapter provides a list of requirements which are desirable for an event detection system. This contained both functional requirements as the performance requirements.

System design Lastly, the chapter has described the event detection system design that this work proposes. The core concept being the reference model which is the basis of the system.

Chapter 5

Evaluation

The goal of the evaluation is to determine both how well the event detection system detects real world events as well as how well the system deals with noise. How well the system can detect real worlds events is evaluated in section 5.2 and how well it deals with noise is tested in section 5.3. For every evaluation, the method is described and the results are shown in their respective sections. The more generic details that hold for every evaluation will be discussed first.

All evaluations used the same machine with the following specifications:

- **Operating system:** Windows 10 Education (64 bit)
- **Motherboard:** Z370 GAMING PRO CARBON AC
- **CPU:** Intel Core i7-8700K CPU @ 3.70GHz
- **RAM:** 32,0 GB DDR4 2666MHz
- **GPU:** NVIDIA GeForce GTX 970

Evaluator The evaluations have been primarily executed by an external person who studied International Relations specialised in Terrorism and Political Violence. During the execution evaluation process, the researcher has observed the evaluation and asked for clarification when reasoning was unclear. Certain parts of the evaluation were done by the author due to the limited amount of time the evaluator was available. Which parts were executed by which person is clearly indicated in the description of the method of both evaluations.

Use case In order to conduct the evaluation, the evaluator was asked: ”to take the role of someone who is interested in (relatively large) conflict in the world. In this context, what would be relevant are large protests, military violence, etc.”. This was also the context in which the system was configured and tested. This includes the creation of the reference model and the use of keywords for the retrieval of tweets.

Keywords The following keywords were used: rocket, radar, launcher, mines, armed, invade, missile, destroyer, cruiser, damaged, killed, battle, combat, clash, engaged, bomb, nuclear, fire, force, threat, intercept, jet, carrier, strike, fighters, defense, cluster, riot, fired, border, incursion, civilians, protest, demonstration. The evaluator was asked to advise on the choice of keywords during testing. However, the author did add additional keywords. When shown to he evaluator, no objections were made to the added keywords.

Used parameters A number of parameters are required in order to run the event detection system. These parameters were determined during the research based on trial and error and have not been formally evaluated. As executing the evaluation multiple times for multiple configurations is time consuming, a formal parameter optimisation and evaluation is left for future work. The parameters that were used for the evaluation of the event detection system are the following:

- **Minimum number of messages:** at least 4 messages
- **Minimum number of reference items:** at least 4 reference items
- **Minimum number of unique sources:** at least 3 unique sources
- **Entity extraction occurrence threshold:** at least 20 occurrences

Data capture window The data was collected overnight on the night of 22-7-2020 to 23-07-2020. The reason for collecting data overnight was that this was the time at which the machine was available to be dedicated to this task. The data stream was enabled and collected tweets during the whole night and into the morning. During the window which was used for the evaluation, 716327 tweets were collected.

Entity extraction The entity extraction was also active during the capture of tweets with an occurrence threshold of 20. Entities which were added due to the automatic entity extraction and deemed to be wrongly detected were not deleted. This was done in order not to disturb the potential disruptive effects of automatic entity extraction and keep this as part of the evaluation. Manual addition of entities was also done on relevant entities which were below the threshold. All the tweets were then run through the event detection system. The resulting events were then used for the evaluations.

5.1 Event metrics

This subsection discusses the metrics which will be used in the evaluation of the method.

Singular events A detected event is singular when the keywords and underlying messages discuss a single event. When keywords or messages of different events are present in a single detected event, this detected event is not singular. If different versions of the same event are given, it is also a singular event. An example of this is with the messages: ”A good meeting took place” and ”A bad meeting took place”. In that case, the event is singular and can be described as: ”A meeting took place”. This is an easy example and real cases are likely to be more complex and ambiguous.

Relevant events A detected event is relevant when the event is of interest to the user. This depends on the use case and the user themselves. This is a very subjective metric. Due to the subjective nature of relevancy and the vague border between relevant and irrelevant, the relevancy question is split into two. We consider whether an event is roughly relevant and whether it is strictly relevant. The definitions of these terms are as follows:

Definition 5.1.1 (Strictly relevant event)**.** A detected event is strictly relevant when it directly references an event which is relevant.

Definition 5.1.2 (Roughly relevant event)**.** A detected event is roughly relevant when it mentions an event which is relevant either indirectly or in the form of an opinion.

The roughly relevant events can be considered relevant in most cases as they do provide some information and/or insight into the desired area. However, if the user is only interested in actual reports of events, then this would not suffice.

5.2 Evaluating Recall

In order to assess the recall of real world events of the event detection system, news articles from BBC World and CNN were used to determine whether the event detection system would also detect the real world events. It is the assumption of the evaluation that the articles of the BBC World and CNN represent real world events. It is therefore assumed that the article is true regardless of what event it describes.

Relevance The event detection system should only detect events which are relevant to the user and the baseline should thus contain only relevant events. The evaluator was thus asked to determine relevancy for each of the events that were reported by CNN and the BBC. If the article was deemed not to be relevant, it was ignored for the following steps. Whether irrelevant events were detected by the system is evaluated in section 5.3. When referring to events reported by BBC World or CNN, this only applies to relevant events from this point forward.

Detected or Undetected For every relevant event, the evaluator was asked to determine whether the event was detected by the event detection system by looking through the detected events. This was done by matching the content of the article to the content of a detected event. If the detected event described or mentioned the event that was described or mentioned in the article, it was classified as a match.

Undetected or Undetectable If a matching detected event could not be found, further analysis was done by the researcher to determine whether the event could have been detected and was not detected or whether the event was undetectable given the captured data. This consisted of manually analysing the dataset of 716327 tweets to determine whether the data stream contained data about the event. In this analysis, the provided parameters were also taken into account. As an event is only detected after 4 tweets have been found, only when 4 tweets about an event were found in the datastream is it considered to be significantly in the datastream. If the data was found to be significantly present in the data stream, the event detection system could have detected the event. The problem then lies with the design itself or the configuration that was used. If the data was not significantly present in the data stream, the data was either never created (tweeted) by users, not delivered by the Twitter API or not captured by the keywords used by the data stream.

5.2.1 Results

Figures 5.1 and 5.2 show the recall of relevant events reported by BBC World and CNN respectively.

Excluding undetectable events We will first look at the results when we set aside the events which were undetectable based on the captured data. In this case we detected 33% and 80% of the relevant events reported by BBC World and CNN respectively as seen in figures 5.1 and 5.2. It is interesting to note that the difference in recall between the two news agencies is quite large given the recall for events reported by CNN is more than double compared to events reported by BBC World.

Including undetectable events We will now include the undetectable events into our calculations and count the undetectable evens as undetected events. In this case we detected 15% and 57% of the relevant events reported by BBC World and CNN respectively as seen in figures 5.1 and 5.2. The difference in recall between the two news agencies remains quite large and the recall for events reported by CNN is now more than triple compared to events reported by BBC World.

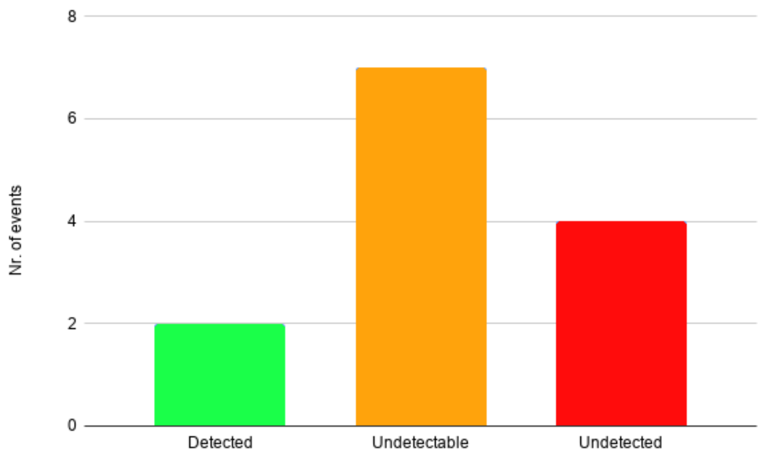

Figure 5.1: Recall of relevant events reported by the BBC

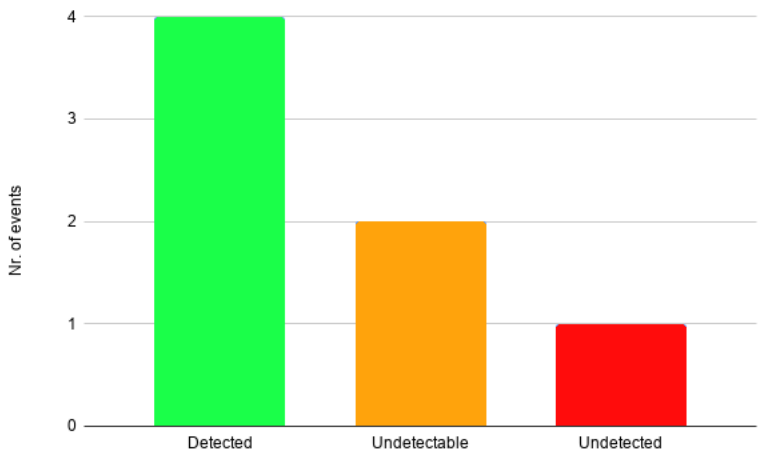

Figure 5.2: Recall of relevant events reported by CNN

Detailed results The results for each of the individual events are listed in tables 5.1 and 5.2 for BBC World and CNN respectively. These show the title of the article as well as whether the event was detected by the system, deemed undetectable based on the captured data or deemed undetected.

Title	Status
Why Turkey-Greece tensions have flared in Med	Detected
Girl 'kills two in fightback against Taliban'	Detected
Abood Hamam seen filming IS takeover of Raqqa	Undetectable
Fugitive Chinese researcher 'hiding in consulate'	Undetectable
Chinese construction workers abducted in Nigeria	Undetectable
Where police may be more deadly than coronavirus	Undetectable
West African leaders head to Mali to mediate crisis	Undetectable
US orders China to close Houston consulate	Undetectable
Lorry driver extradited to UK over Essex deaths	Undetectable
Islamist militants kill five aid workers in Nigeria	Undetectable
In pictures: Portland's 'Wall of Moms'	Undetected
Timeline: Events after Harry Dunn crash death	Undetected
US orders China to close Houston consulate	Undetected
Prominent Zimbabwean journalist charged with inciting violence	Undetected

Table 5.1: Results of event detection system on BBC World articles

Title	Status
Portland mayor tear gassed after speaking with protesters on presence of federal agents	Detected
15 injured in Chicago drive-by shooting at funeral for man killed in drive-by shooting	Detected
Trump accused of deploying 'secret police' as part of 'authoritarian' law enforcement surge	Detected
NYPD disperses protest encampment outside City Hall	Detected
China harboring military-linked fugitive scientist at San Francisco consulate, FBI says	Undetectable
Philadelphia police officer charged for pepper spraying Black Lives Matter protesters, prosecutors say	Undetectable
AG Barr calls reaction to George Floyd death 'extreme'	Undetected

Table 5.2: Results of event detection system on CNN articles

News agency	Detected events	Undetectable events	Undetected events
BBC	2	7	4
CNN	4	2	1
Total	6	9	5

Table 5.3: Overview of the performance of the event detection system

A full overview of the data can be seen in Table 5.3. When we disregard the undetectable events, 54% of the reported events by both BBC World ore CNN were detected by the event detection system. When we include the undetectable events, this number drops down to 30%.

5.2.2 Discussion

When looking at the results, it becomes instantly clear that there is a significant difference between the recall for BBC World and CNN. This difference is likely caused by the chosen data capture window. The data was captured late in the evening and at night for the UK (21:35-02:45) and during the day and early evening for the US (16:35-21:45). A large amount of the tweets will therefore have been from the united states as opposed to the united kingdom. This is confirmed when looking at the Twitter user locations from tweets in the captured data. The term "USA" occurred 63,867 times in the data and the term "UK" only occurred 3,191 times in the data. This seems like a sound explanation as to why events reported by an American news agency, which likely to be relevant to Americans, are more prevalent in the captured data.

Undetectable events Looking at both BBC World and CNN combined, 45% of the relevant reported events were deemed undetectable due to insignificant data in the data stream. This is almost half of all the reported event and raises some questions as to the performance. The two situations in which events are undetectable are when no data created by users or not delivered by Twitter. The latter can be caused due to limit on the amount of messages you can retrieve per second that are imposed on the free live data stream or because the filter that is put on the data stream filters out the relevant messages. Unfortunately, it is not possible to determine whether the data was never created or not retrieved without paying for full access to the datastream in order to check.

Strict reference requirements When looking at whether an event had a corresponding detected event, the reference had to be precise in order to be considered a corresponding pair of events. An example of a relevant reported event which has no strictly corresponding detected event but could be considered to be loosely referenced has the headline: ”US orders China to close Houston consulate”. This order was given after documents were being burned in the consulate. The burning of the documents was detected, however the closure order of the consulate was not. It is then a question at what level you want to define the event. When using the strict descriptions as mentioned, these do not correspond to a single event. However when considering the event: ”A diplomatic conflict between the US and the Chinese consulate in Houston”, both the reported and detected event would refer to the same event. In this case the reported event was recalled by the system. This example illustrates how the results have to be interpreted with the strict evaluation choice in mind.

Why were detectable events missed? The main reason for missing the events which were detectable is that they contained keywords which were not in the reference model. This mainly happened due to certain events being unforeseen when the reference model was build. The undetected event: ”Timeline: Events after Harry Dunn crash death” describes a fatal car crash but was deemed relevant due to a conflict with diplomatic immunity for the suspect. However, this scenario was not incorporated into the reference model and thus not deemed relevant by the event detection system. This shows that the reference model is an integral part of the system.

5.2.3 Conclusion

When we assume that the undetectable events were caused by shortcomings of the system and look at all the relevant reported events, the resulting recall of 30% is quite underwhelming. This recall would be unusable in real world scenario for most stakeholders as the majority of the events would not be detected.

Best case scenario However, when we assume that the undetectable events were not in the data, the recall of 54% is already a significant improvement. Furthermore, if we focus on events reported by CNN based on the data capture window mentioned earlier, we reach a potential recall of 80%. This result would be considered a satisfactory result as this captures a large majority of the real world events.

This evaluation does not provide a hard result for the recall, but shows that, when taking the circumstances into consideration, the event detection system detects a large majority of the real world events. To what degree these considerations assume too much is open to interpretation.

5.3 Evaluating precision

The purpose of the precision evaluation is to determine the quality of the detected events that were produced by the event detection system. Three metrics were used: strictly relevant, roughly relevant and singular. These metrics were used to evaluate all 99 detected events. The events were sorted in order from the event with the most amount of tweets to the event with the least amount of tweets.

Strictly relevant A tweet was deemed strictly relevant when it referenced an event which was relevant. This required a direct description of a real world event. This metric was determined by the evaluator.

Roughly relevant Tweets that do not directly reference an event or do so in the form of an opinion about an event are not considered strictly relevant. However, these tweets could still be considered relevant for the stakeholder as described at the start of this chapter. In order to create a distinction between these two situations, the separate relevancy metrics were used. This metric was determined by the evaluator.

Assumption of reliability In order to determine the relevancy of a detected event, all the tweets the event is based on were assumed to be reliable and thus true. The reliability of messages and events is out of the scope of the research and thus relevancy is determined under the assumption that the detected event corresponds to a real world event. This means that an event would not be considered to be irrelevant due to the determination by the evaluator that an event is obviously false and thus not relevant.

Singular A detected event is considered to be a singular event when it references a single real world event as opposed to multiple events. This does not mean that every tweet within a detected event must have the same view on the corresponding real world event nor that they discuss the same aspect of the real world event but it should concern only one real world event. This metric was determined by the evaluator.

5.3.1 Results

This subsection contains the results of the precision evaluation.

Strictly relevant Figure 5.3 shows the percentage of detected events which were considered strictly relevant to the stakeholder by the evaluator. Roughly one third of the detected events was considered strictly relevant. If we look at only the 50 largest detected events, this goes up to 51%.

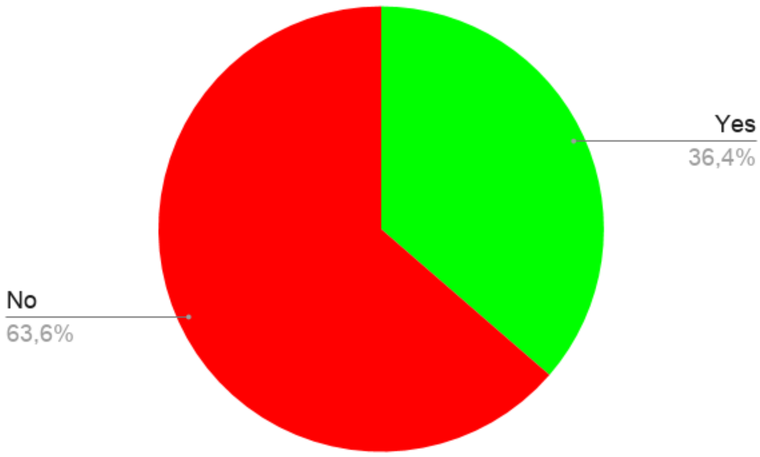

Figure 5.3: Percentage of events that are strictly relevant

Roughly relevant Figure 5.4 shows the percentage of detected events which were considered roughly relevant to the stakeholder by the evaluator. Approximately two thirds of the detected events were found to be roughly relevant which is almost double the amount of strictly relevant detected events. If we look at only the 50 largest detected events, this goes up to 69%.

Singular Figure 5.5 shows the percentage of detected events which were considered singular by the evaluator. Approximately 90% of the detected events were found to be singular. If we look at only the 50 largest detected events, this goes down to 83%.

Detailed analysis Figure 5.6 shows the percentage of events that are relevant/singular over the nr. of events in calculating the average of the metrics. On the left side of the chart, only a single event is included whereas on the right, all detected events are included. We can see that the smaller events are more singular than the larger events as the percentage of singular events goes up. We can also see that the percentage

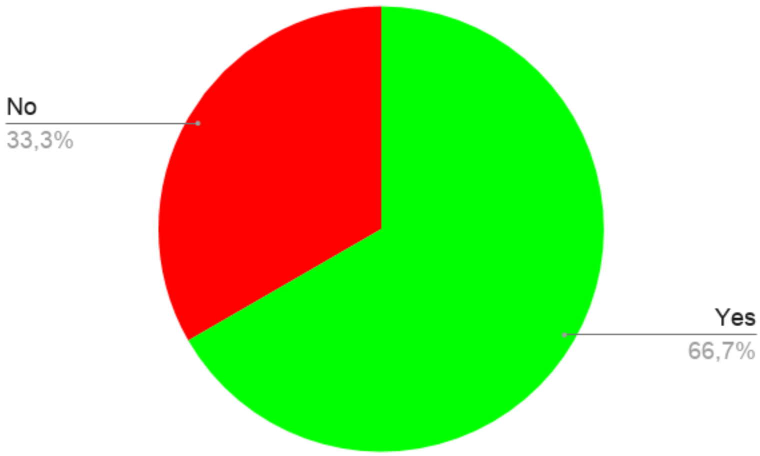

Figure 5.4: Percentage of events that are roughly relevant

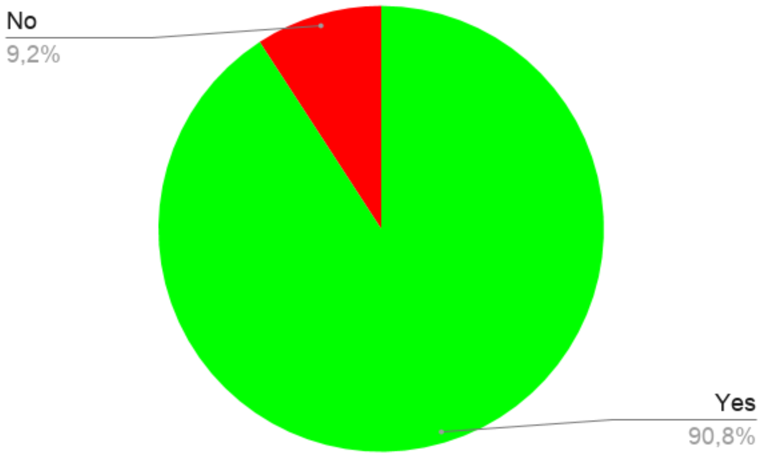

Figure 5.5: Percentage of events that are singular

of roughly relevant events starts to stabilise between 65 and 70 percent whereas the percentage of strictly relevant events keeps dropping when more smaller events are added.

5.3.2 Discussion

Roughly vs Strictly relevant It is important to determine whether an event should be considered relevant when it is roughly or strictly relevant. In most use cases, as well as the use case that was used in this evaluation, roughly relevant should be considered as relevant. Even an indirect or subjective mention of a relevant event can provide key insights and inform the user about contextual information. Although this should be merged into a single event about the mentioned event in future, the

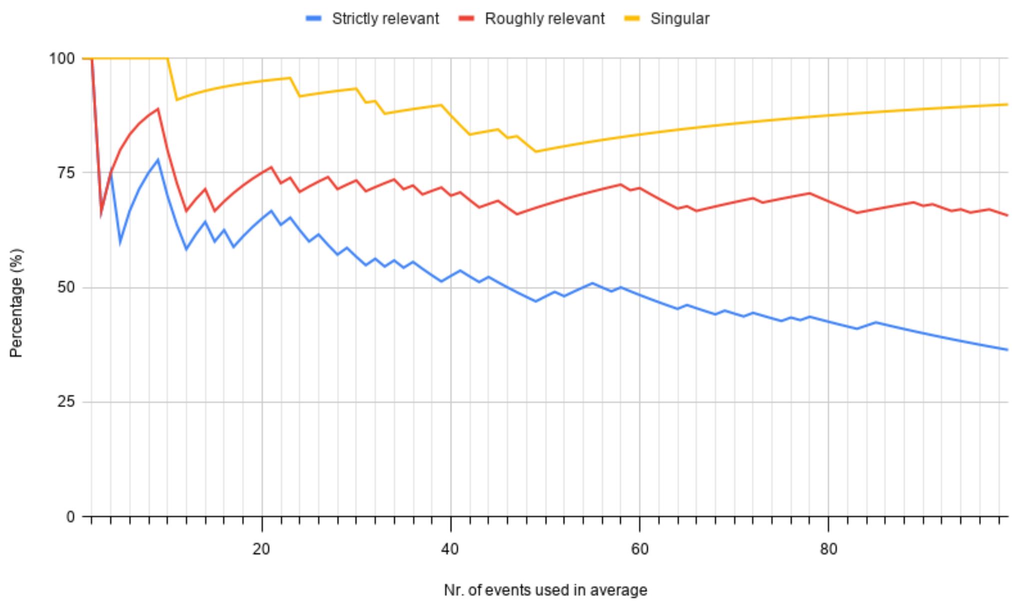

Figure 5.6: Percentage of events that are relevant/singular over nr. of included events

roughly relevant events still contain relevant information.

Duplicate detection There were 5 events which were detected multiple times by the event detection system. This is due to the different descriptions of the same event. In some of these cases, the events could be combined through the detail hierarchy structure as described in subsection 2.4.1.

5.3.3 Conclusion

Looking at events that are strictly relevant, 36.4% is a rather low percentage though not completely unusable. When we consider roughly relevant events as relevant, we consider 66% a sufficient percentage. This means that only 1 in 3 events are not relevant to the user which should still be reduced in future work but is quite usable in a real world scenario.

Chapter 6

Discussion

This chapter will reflect on the research that was done, discuss the results of the evaluations that were done and look at what can be done next in order to improve the event detection system.

6.1 Discussion of recall evaluation

This section discusses the results of the recall evaluation and also gives more insight into some of the choices made in setting up the evaluations.

6.1.1 Methodology

Undetectability analysis In order to determine whether an event was undetected or undetectable, an analysis was done as described in section 5.2. This analysis was done based on querying the dataset of captured tweets. The keywords were determined based on the words used in the articles as well as words which were considered likely to be used in tweets in regard to the event. This analysis does not completely prove that an event is undetectable as the queries might have been incomplete. However, a lot of care was taken in order to prevent a false positive.

Dependency on correct configuration The event detection system heavily depends on the model of entities and other keywords in order to detect anything. The cases in the evaluation on news articles where there were a significant amount of messages in the data but the event was not detected might be solved by updating the model. This was intentionally not done in order to properly test the system. In a real use scenario, it would also not be possible to change the model to detect events that you have missed because you would not know they exist. By not tailoring the model to the evaluation, a more realistic view of the performance of the event detection system is provided.

Baseline choice The choice was made to use news articles from English news organisations instead of a snapshot of existing event datasets such as ACLED as these either are capable of processing messages in multiple languages or use data which is not open source. These significant differences were considered to be so great that the comparison to articles from English news organisations was a fairer evaluation.

Potential baseline alternative Another option that was considered was to use an existing set of tweets that contained a number of events which were known to be in the dataset. However, due to policies set by Twitter, datasets are only allowed to provide the ids of the tweets and not the tweets themselves. Due to this, it is necessary to manually retrieve the tweets based on the ids in the dataset. This was no longer possible since a large portion of the tweets were no longer available. It could therefore no longer be assumed that the dataset contained the events. This option was therefore ruled out and an evaluation on news articles was chosen as the best time efficient option.

6.1.2 Results

Whether to include or exclude undetectable events is a difficult decision. Whether these events could have been detected is unknowable for the purposes of this research. The only method to fully check whether an event is undetectable requires access to the full Twitter Firehose which includes all tweets. As this is very expensive, this is not an option. This leaves us with the limited Twitter gardenhose which only receives a fraction of all the tweets in the firehose. The chosen keywords on which we filter the firehose could limit the tweets in such a way to cause an event to become undetectable while it would have been detectable in the firehose. This could be seen as a limitation of the event detection system or as a limit of the Twitter gardenhose. Either way, the real world event cannot be detected. In order to be in line with the goal of detecting real world events, we must include undetectable events. It is however valuable to look at the performance excluding undetectable events in order to assess the capabilities of the system better.

6.2 Discussion of precision evaluation

6.2.1 Methodology

Relevancy The distinction of strictly relevant and roughly relevant was developed during the evaluation in response to comments made by the evaluator. Certain events might not directly pertain to the stakeholders main aim but could still be important. An example of this is when we look at the following tweet: ”Acting Border Protection Commissioner Says Chicago Mayor Lori Lightfoot’s Claims About Federal Agents In Portland ‘Devoid Of Any Fact Or Truth’”. The event ’person A says that person B is lying’ is not directly related to conflict or violence. However, the event does indirectly reference the conflict that was happening in Portland and does provide information about the event in Portland. This would thus be considered to be roughly relevant and not strictly relevant.

Avoiding petitions One example of detected events which were not roughly relevant are petitions. These occur often in the datastream as they consist of a large amount of messages and will be considered events when they match enough reference items. This can be avoided in a next iteration by requiring the texts of the tweets to be at least slightly different.

Duplicate events An important observation when looking at the detected events is that real world events were detected multiple times which resulted in duplicate detected events. This can be attributed to the use of different keywords in describing the same event. Whether this is desirable is debatable. The different detected events provided each a different set of details about the real world event. To solve this, perhaps some clustering needs to be done on the detected events as well.

6.3 Performance

During the runs of the event detection system done for the evaluations as described in chapter 5, the execution time was also evaluated. As the system is supposed to be light weight, it would be problematic if the system were pushing the limits of the hardware or unable to keep up with the incoming data stream.

The application was able to process the tweets at a speed of roughly 1000 tweets per second at the end of processing the data given the commodity computer system specified at the beginning of chapter 5. When the system was just started, the speed was much greater as it did not have to look through many existing events to see whether it matched. When almost all data was processed the speed was dipping just below 1000 tweets per second. The average amount of tweets per second on Twitter is 5700[1]. However this is not how many tweets are available to developers who do not pay to use the Twitter API. During the research, the Twitter stream provided roughly 60 tweets per second at maximum. Twitter allows for two streams to be able to run simultaneously which would cap the amount of tweets to 120 per second. This is well within the capability of the event detection system.

The CPU usage of the application itself hovered around 14% at a clock speed of roughly 4.35 GHz.

6.4 Limitations

6.4.1 Heavy reliance on reference model

The system relies heavily on the quality of the reference model. Creating a good reference model is not a trivial task but must be done by every user in order to detect relevant events. Once such a model has been constructed, it could be shared and improved slowly, however this still needs a lot of work and the quality of the model is hard to assess.

6.4.2 Linking of similar reference item sets

The reference model was used in the design in order to link closely linked terms. Apart from that, the system does not merge detected events based on how closely related they are to other events. If there are detected events which have more reference items or events which differ slightly, they are simply different events. The choice was made to prefer more smaller detected events about the same real world event as opposed to a single detected event that is a mix of multiple real world events.

[1] https://blog.twitter.com/engineering/en_us/a/2013/new-tweets-per-second-record-and-how.html

Chapter 7

Conclusion

In this research we set out to design an event detection system that can be used by most stakeholders. In order to reach this goal, we looked at defining an event, created an overview of what we want to know about an event and designed a system which can detect these events based on a messages stream.

7.1 Answers to the research questions

7.1.1 What is an event?

An analysis of event definitions, properties and mechanisms was done by looking at existing definitions and expanding on the elements found in those definitions. We defined an event as a real world event as opposed to a discussion event. This brought different considerations to the evaluation of the event detection system as is normally done when evaluating discussion events. The goal of the event detection system was thus to detect real world events that are relevant to the user.

7.1.2 What do we want to know of an event?

An informal user study was done among OSINT enthusiasts in order to find what pieces of information are important to know about an event. The identification of internal and external information showed that aside from information about a single event, knowledge about the context in which events occur is desired as well.

7.1.3 How can we detect and track relevant events?

An event detection system which relies on a user defined reference model supported by Named Entity Recognition was designed. The reference model plays a key part in linking keywords with the same meaning and extraction of meaning from the messages from the message stream.

7.2 Evaluation

Although the event detection design performed moderately, steps have been identified in the future work which could improve the performance of the system. The designed event detection system was evaluated on both recall and precision.

Recall evaluation The overall outcome of the recall evaluation is quite underwhelming with a recall of 30%. The recall of the system was expected to be quite low given the definition of an event as a real world event. When we shift from real world events to discussion events (which excludes events that did not appear in the message stream) and take into account the data capture window by looking only at events reported by US media, the recall jumps to 80%. This is considered a very good result for the recall evaluation.

Precision evaluation When we look at results for both relevancy metrics we observed 36.4% and 66% for strictly (directly related to the event) and roughly (directly or indirectly related to the event) relevant respectively, both outcomes will be acceptable for most users. Although it is very inconvenient to filter 2 out of every 3 messages, this is still doable given that 1 out of three events are relevant to the user. The percentage of 66% is significantly higher and provides mostly relevant events to the user. For most purposes, roughly relevant can be considered relevant 'enough' which means that most users will experience a precision of 66%.

7.3 Future work

Although the results are sufficient and the event detection system is usable, there is still plenty of room for improvement. This section discusses the future work which is still open after the conclusion of the thesis.

7.3.1 Reference item types

A possible improvement to the reference item is to use the reference item types to determine whether a event was detected. The current system only checks that there are a minimum amount of linked reference items in a message. This becomes a problem when a message contains a number of location names above the threshold.

If the system required a minimum number of linked reference item of certain types i.e. one location, one actor and one action, this problem could be negated.

7.3.2 Better entity extraction

Improvements are still desired on the front of entity extraction as not all entities are detected and entities are detected which are not entities. There are alternative NER options such as TwiNER but perhaps a custom solution is required to achieve the needed quality of entity detection.

7.3.3 Multiple languages

In order to support multiple languages we do not need to translate every message but we can translate the reference model instead. If we translate all the reference item keywords to other languages, then use those to match to messages of their respective language, we can detect events from all languages. Once an event has been detected, you can translate the underlying messages which saves in computing power in comparison to translating all non-English messages to English.

7.3.4 Better deduplication

The reference model items could be extended to include the parts of speech (word classes) e.g. VERB, NOUN, etc. in order to make more make duplicates less common. An example of where this could have been useful is the keyword fire. This was meant to only refer to an actual fire, but was also detected in the context of firing an person. The addition of part-of-speech tagging has the potential to solve some of these conflicts.

Bibliography

[1] Sakaki, T., Okazaki, M., Matsuo, Y. (2010). Earthquake shakes Twitter users: real-time event detection by social sensors. In Proceedings of the 19th international conference on World wide web (pp. 851-860). ACM.

[2] Sakaki, T., Okazaki, M., Matsuo, Y. (2013). Tweet analysis for realtime event detection and earthquake reporting system development. IEEE Transactions on Knowledge and Data Engineering, 25(4), 919- 931.

[3] Ekta, P., Bundela, P., Dewan, R. (2017). Tweet Analysis for Real- Time Event Detection and Earthquake Reporting System Development. International Research Journal of Engineering and Technology (IRJET), 4(5).

[4] A. Weiler, H. Schilling, L. Kircher, and M. Grossniklaus, "Towards reproducible research of event detection techniques for twitter," in *2019 6th Swiss Conference on Data Science (SDS)*, 2019, pp. 69–74.

[5] K. Lee, A. Qadir, S. A. Hasan, V. Datla, A. Prakash, J. Liu, and O. Farri, "Adverse drug event detection in tweets with semi-supervised convolutional neural networks," in *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 705–714. [Online]. Available: https://doi-org.ezproxy2.utwente.nl/10.1145/3038912.3052671

[6] "Event — definition in the cambridge english dictionary." accessed: 13 July 2020. [Online]. Available: dictionary.cambridge.org/us/dictionary/english/event

[7] "Event — definition of event by merriam-webster," accessed: 13 July 2020. [Online]. Available: https://www.merriam-webster.com/dictionary/event

[8] "event noun - definition, pictures, pronunciation and usage notes — oxford advanced learner's dictionary at oxfordlearnersdictionaries.com," accessed: 26 July 2020. [Online]. Available: https://www.oxfordlearnersdictionaries.com/us/definition/english/event

[9] M. Hasan, M. A. Orgun, and R. Schwitter, “Real-time event detection from the twitter data stream using the twitternews+ framework,” *Information Processing & Management*, vol. 56, no. 3, pp. 1146 – 1165, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306457317305447

[10] H. Becker, M. Naaman, and L. Gravano, “Beyond trending topics: Real-world event identification on twitter,” *ICWSM*, vol. 11, 01 2011.

[11] W. Dou, X. Wang, W. Ribarsky, and M. Zhou, “Event detection in social media data.” IEEE, 2012, p. 971–980.

[12] Z. Tan, P. Zhang, J. Tan, and L. Guo, “A multi-layer event detection algorithm for detecting global and local hot events in social networks,” *Procedia Computer Science*, vol. 29, pp. 2080 – 2089, 2014, 2014 International Conference on Computational Science. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S187705091400369X

[13] J. Allan, *Topic Detection and Tracking: Event-Based Information Organization*. Springer Publishing Company, Incorporated, 2012.

[14] W. Ma, Z. Liu, and X. Hu, “Online event detection in social media with bursty event recognition,” in *Security and Privacy in Social Networks and Big Data*, W. Meng and S. Furnell, Eds. Singapore: Springer Singapore, 2019, pp. 181–190.

[15] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu, “Parameter free bursty events detection in text streams,” in *Proceedings of the 31st International Conference on Very Large Data Bases*, ser. VLDB ’05. VLDB Endowment, 2005, p. 181–192.

[16] J. Li, Z. Tai, R. Zhang, W. Yu, and L. Liu, “Online bursty event detection from microblog,” in *Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, ser. UCC ’14. USA: IEEE Computer Society, 2014, p. 865–870. [Online]. Available: https://doi-org.ezproxy2.utwente.nl/10.1109/UCC.2014.141

[17] M. Fedoryszak, B. Frederick, V. Rajaram, and C. Zhong, “Real-time event detection on social data streams,” *CoRR*, vol. abs/1907.11229, 2019. [Online]. Available: http://arxiv.org/abs/1907.11229

[18] “Beirut explosion: What we know so far,” accessed: 18-9-2020. [Online]. Available: https://www.bbc.com/news/world-middle-east-53668493

[19] “Trending words on 4th august, 2020,” accessed: 18-9-2020. [Online]. Available: https://us.trend-calendar.com/trend/2020-08-04.html

[20] D. Gerner, R. Abu-Jabr, P. A. Schrodt, and Ö. Yilmaz, “Conflict and mediation event observations (cameo): A new event data framework for the analysis of foreign policy interactions,” 2002.

[21] P. A. Schrodt, Ömür Yilmaz, D. J. Gerner, D. Hermrick, A. Bron, A. Gregory, A. Ingram, M. Jekic, L. Mcmullen, L. Prather, and T. Price, “Coding sub-state actors using the cameo (conflict and mediation event observations) actor coding framework,” in *in Annual Meeting of the International Studies Association*, 2008.

[22] “Cameo event data codebook version 1.1b3,” accessed: 26 July 2020. [Online]. Available: http://eventdata.parusanalytics.com/data.dir/cameo.html

[23] Quanzhi Li, Armineh Nourbakhsh, Sameena Shah, and Xiaomo Liu. 2017. Realtime novel event detection from social media. In 2017 IEEE 33rd International Conference on Data Engineering (ICDE). IEEE, 1129–1139.

[24] A. Farzindar and W. Khreich, “A survey of techniques for event detection in twitter,” *Comput. Intell.*, vol. 31, pp. 132–164, 2015.

[25] S. Gaglio, G. L. Re], and M. Morana, “A framework for real-time twitter data analysis,” *Computer Communications*, vol. 73, pp. 236 – 242, 2016, online Social Networks. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366415003631

[26] F. B. Shannag and B. H. Hammo, “Lessons learned from event detection from arabic tweets: The case of jordan flash floods near dead sea,” in *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, 2019, pp. 806–811.

[27] K. Morabia, N. L. Bhanu Murthy, A. Malapati, and S. Samant, “SEDTWik: Segmentation-based event detection from tweets using Wikipedia,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 77–85. [Online]. Available: https://www.aclweb.org/anthology/N19-3011

[28] Tonon, A., Cudr-Mauroux, P., Blarer, A., Lenders, V., Motik, B. (2017). ”ArmaTweet: detecting events by semantic tweet analysis”. In European Semantic Web Conference (pp. 138-153). Springer, Cham.

[29] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, “The Stanford CoreNLP natural language processing toolkit,”

in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60. [Online]. Available: http://www.aclweb.org/anthology/P/P14/P14-5010

[30] Apache Software Foundation, "openNLP Natural Language Processing Library," 2014, http://opennlp.apache.org/. [Online]. Available: http://opennlp.apache.org/

[31] C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B.-S. Lee, "Twiner: Named entity recognition in targeted twitter stream," in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 721–730. [Online]. Available: https://doi.org/10.1145/2348283.2348380

[32] M. Habib and M. van Keulen, "Twitterneed: a hybrid approach for named entity extraction and disambiguation for tweets," *Natural language engineering*, vol. 22, no. 3, pp. 423–456, May 2016, eemcs-eprint-26014.