Technische Universität Ilmenau

Ereignisdetektion und Stimmungsdetektion im Echtzeitdatenstrom von sozialen Netzwerken

---

Dissertation zur Erlangung des akademischen Grades
Dr.-Ing.
vorgelegt der Fakultät für Informatik und Automatisierung
der Technischen Universität Ilmenau
von

Frank Zimmermann
geboren am 07.05.1979 in Mühlhausen/Thüringen

Gutachter: PD Dr.-Ing. habil. Jürgen Nützel
Prof. Dr.-Ing. habil. Kai-Uwe Sattler
Prof. Dr. Gerhard Heyer

Ort und Datum der Einreichung: Ilmenau, 01.03.2016
Datum der wissenschaftlichen Aussprache: 23.01.2017

Danksagungen

Besonderer Dank gebührt meinem Betreuer Herrn PD Dr.-Ing. habil. Jürgen Nützel, der es mir erst ermöglichte, neben meiner beruflichen Tätigkeit, eine Dissertation durchzuführen. In zahlreichen Gesprächen mit ihm, gab er mir die notwendigen Impulse mich mit dem Thema der ortsbasierten Analyse von Daten zu beschäftigen. Auch im weiteren Verlauf meiner Arbeit stand er jederzeit als Gesprächspartner zur Verfügung.
Einen weiteren Dank geht an Herrn Prof. Dr.-Ing. habil. Kai-Uwe Sattler und an Herrn Prof. Dr. Gerhard Heyer, die mir wertvolle Hinweise gegeben haben, um die vorliegende Arbeit zu verbessern.
Auch möchte ich meinen Kollegen Stefan Richter danken, der den Kampf gegen den Fehlerteufel aufgenommen hat und mich auf endlos verschachtelte Sätze hingewiesen hat.

Frank Zimmermann
Weinbergen
01.03.2016

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

Auf die einfache Frage: „Was geschieht gerade?“ kann man auf verschiedene Arten versuchen eine Antwort zu finden, je nachdem wer diese Frage stellt. Stellt z.B. eine Nachrichtenredaktion diese Frage, so ist sie auf ihre eigenen Journalisten bzw. andere externe Quellen angewiesen, die das aktuelle Geschehen der Redaktion berichten. Stellt man die Frage selbst, so kann man einen Nachrichtensender (Fernseher oder Radio) einschalten, um die neuesten Nachrichten zu sehen / zu hören oder man ruft im Internet eines der unzähligen Nachrichtenportale auf. Diese Nachrichtenquellen beantworten aber nicht ganz die Frage was gerade jetzt geschieht, sondern eher was bereits geschehen ist. Nur bei sehr wenigen Ereignissen, wie z.B. gerade stattfindenden Katastrophen, Sportereignissen oder anderen bedeutenden Ereignissen, wird der Zuschauer in Echtzeit über die gerade stattfindenden Geschehnisse informiert („Breaking-News“). Die restlichen Ereignisse sind meist schon geschehen und es wird im Nachgang, bei TV-Nachrichtensendungen nach ggf. ein paar Stunden oder in der Tageszeitung unter Umständen erst am nächsten Tag, davon berichtet. Des Weiteren liefern diese Nachrichtenquellen einen redaktionellen Blick auf das Geschehen. Auf die Redaktion der Nachrichtenquelle und auf deren Quellen und Journalisten kommt es an, welche Ereignisse von ihnen überhaupt registriert werden und wie über diese berichtet wird. Die jeweilige Nachrichtenquelle trifft somit eine Vorauswahl und entscheidet welche Themen wichtig oder unwichtig sind.
Dabei bieten heute die sozialen Netzwerke die potentielle Möglichkeit direkt auf die Erfahrungsberichte der Nutzer zuzugreifen die dem gerade stattfindenden Ereignis beiwohnen. Überall auf der Welt nutzen Menschen soziale Netzwerke und produzieren täglich eine Unmenge an neuen Inhalten über das, was gerade geschieht, was sie gerade tun oder womit sie sich beschäftigen. Sie versenden Nachrichten, veröffentlichen Positionsdaten (Checkins – man teilt mit wo man sich gerade befindet und man sieht wer sich noch hier befindet bzw. wer an dem Ort war [1]), Bilder, Videos und bewerten und teilen (sharen – das Teilen von Medien über digitale Kanäle um Freunde oder andere Nutzer auf Medieninhalte aufmerksam zu machen [2]) anderer Inhalte. Allein auf der Videoplattform YouTube werden pro Minute 300 Stunden [3] Videomaterial hochgeladen. Bei dem größten sozialen Netzwerk Facebook, mit seinem 1,44 Mrd. monatlich aktiven Nutzern [4], werden pro Minute ca 3,3 Mio. Objekte geteilt, 3,1 Mio. Objekte geliked (das Markieren von Medieninhalten um Freunde oder andere Nutzer auf diese Medieninhalte aufmerksam zu machen [5]) [6] und es kommen pro Minute 243000 neue Bilder hinzu. [7]
Diese gewaltigen Datenströme beinhalten somit eine große Menge an Informationen die die gerade stattfindenden Geschehnisse auf der Welt beschreiben. In ihnen stecken all die Informationen über die gerade stattfindenden Ereignisse, die von den Nutzern der sozialen Netze oder verbundenen Computersystemen wie z.B. Sensorennetzwerke die Umwelt- (z.B. Wetterdaten, seismische Aktivitäten usw.) und Verkehrsdaten erfassen und automatisch übermitteln. Sei es die Nachricht über ein gerade laufendes Sportereignis, eine Zugverspätung, die aktuellen Wetterdaten in New York oder die gerade wütende Unwetterkatastrophe (Abbildung 1). In diesen Daten liegt die ungefilterte Antwort auf die Frage was gerade und wo es auf der Welt geschieht.

Abbildung 1: Beispiel einer übermittelten Statusnachricht mit Foto und angehangener geographischen Koordinate an das soziale Netzwerk Instagram während des Schneesturms am 3. Januar 2014 am Times Square. [8]

Um diese große Menge an Daten zu verarbeiten und in Echtzeit die Frage, was gerade geschieht, beantworten zu können, bedarf es Algorithmen, die den Echtzeitdatenstrom eines oder mehrerer sozialen Netzwerke analysieren können, um Veränderungen in diesen, die auf ein neues Ereignis hindeuten könnten, zu erkennen. Durch diese automatische Analysemöglichkeit wäre es auch möglich, die Granularität der gewünschten zu entdeckenden Ereignisse zu steuern. So ist es denkbar, dass der normale Endnutzer nur aus bestimmten Regionen, z.B. aus seinem Heimatort, auch über regional begrenzte kleinere Ereignisse (z.B. das aktuelle Spiel im Station der Stadt) informiert werden möchte, während regional begrenzte Ereignisse die weiter weg geschehen nicht von Interesse eines Nutzers sind. Andere Nutzergruppen wie z.B. eine Nachrichtenredaktion möchte dagegen die regionalen Ereignisse aus einem größeren Gebiet erfassen um schnell auf neue Ereignisse mit eigenen Recherchen reagieren zu können.
Ein weiterer Anwendungsfall wäre, dass bei einem, durch die Algorithmen erkannten, großen Ereignis wie z.B. einer Katastrophe, die Nutzer sich über eine Push-Benachrichtigung (eine Nachricht, die den Empfänger sofort erreicht ohne zuvorige Nachfrage bei einem Server vgl. SMS) in Echtzeit informieren lassen können. Auch für die öffentlichen Einrichtungen wären diese Informationen im Falle einer Katastrophe von Bedeutung, um z.B. im Falle eines Erdbebens schnell Informationen über die Schäden oder Vermissten Menschen zu erhalten, welche durch die Nutzer eines sozialen Netzwerkes vor Ort gemeldet wurden, um so die Hilfen besser und schneller koordinieren zu können. Ein Bereich an dem EU-weit auch geforscht wird. [9]
Auch andere Aufgabenstellungen lassen sich mit Hilfe der Daten in den Datenströmen und einer ortsabhängigen Analyse bearbeiten. So ist es auch möglich, durch die Analyse von Text nicht nur neue Ereignisse zu erkennen, sondern auch die Stimmung eines Textes zu ermitteln. Lobt der Autor eines Textes gerade das gute Wetter oder ärgert er sich gerade über das beschriebene Produkt? Werden solche Analysen ebenfalls automatisch durchgeführt, so können sie interessante Dinge sichtbar machen wie z.B.

wie zufrieden sind die Nutzer des sozialen Netzwerkes gerade mit etwas, wie z.B. dem Wetter, einer öffentlichen Person, einem Produkt oder der Dienstleistung einer Firma usw. Diese Analysen können über lange Zeiträume durchgeführt werden, wodurch man Änderungen der Werte ermitteln kann. Wenn es z.B. ein Problem mit einem Produkt gibt, wird sich die Stimmung in Bezug auf das Produkt oder die Produktionsfirma verschlechtern. Somit können frühzeitig potentielle Probleme erkannt werden. Durch die ortsabhängige Analyse dieser Daten ist es auch möglich lokale Schwankungen zu registrieren und die räumliche Quelle von Stimmungsänderungen/-verteilungen zu erkennen.

1.2 Beiträge dieser Arbeit

Diese Arbeit stellt neue Algorithmen vor, welche aus einem Echtzeitdatenstrom von georeferenzierten eintreffenden Kurznachrichten eines sozialen Netzwerkes aktuell stattfindende Ereignisse und deren Ort erkennen kann. Die Detektion dieser Ereignisse wird mit einer sprachunabhängigen, auf Termhäufigkeitsänderungen basierten Echtzeit-Ereignisdetektion, welche ohne Vorverarbeitung bzw. Filterung der chronologisch eintreffenden Eingangsdaten auskommt, realisiert.
Um dies zu realisieren wurden folgende Neuerungen entwickelt:

- Einführung eines adaptiven Referenzkorpus für die Differenzanalyse zur Verbesserung der Detektionsergebnisse [10]
- Erweiterung der Korpora zu ortsabhängigen Korpora zur Verbesserung der Differenzanalyse durch die Beachtung von lokalen Termhäufigkeitsausprägungen und die zusätzliche Möglichkeit der Benennung des Ortes des detektierten Ereignisses [11]
- Ablauf der Ereignisdetektion ohne die Notwendigkeit einer Vorverarbeitung oder Filterung der Eingangsdaten und somit der Realisierung einer sprachunabhängigen Ereignisdetektion durch die Verwendung von adaptiven räumlich begrenzten Korpora (ortsabhängige Korpora)

Um die Analyse zu realisieren werden ortsabhängige adaptive Korpora (Analyse- / Referenzkorpus) erzeugt, um anschließend eine Differenzanalyse durchzuführen. Durch die Ortsabhängigkeit der Korpora und des adaptiven Referenzkorpus führt die Ereignisdetektion zu besseren Ergebnissen (siehe Kapitel 5). Durch Modifikationen der ortsabhängigen Korpora kann Einfluss auf die Ergebnismenge der Ereignisdetektion genommen werden, um unterschiedliche Einsatzszenarien zu realisieren bzw. die Analyse auf Orte mit unterschiedlicher Kurzmitteilungsaufkommen pro Fläche und Zeit anzupassen. Dabei kann zum einem die Zusammensetzung der Korpora oder aber auch die Detektionsschwellen bei der Differenzanalyse verändert werden. Die Zusammensetzung der Korpora lässt sich variieren indem man den Radius des Einflussbereiches zum Analyseort verändern kann und so z.B. auch weiter entfernte Kurzmitteilungen in die Analyse mit einfließen lassen kann.
Durch Separierung und Aufbereitung der Ergebnismenge der Ereignisdetektion können die gefunden Ergebnissen einzelnen abgrenzbaren Ereignissen zugeordnet werden. Um die Ergebnisse besser zu beschreiben, werden die Ereignisse mit zusätzlichen extrahierten Mediendaten wie z.B. Bildern angereichert und entsprechend präsentiert.
Für die Analysen selbst wurde eine verteilte skalierbare Systemarchitektur entworfen, mit der unabhängige gekapselte Echtzeit-Analysen über mehrere Rechner hinweg möglich sind und deren Ergebnisse sofort als Website aufbereitet präsentiert werden können.
Die im Rahmen dieser Arbeit entwickelten und praktisch getesteten Algorithmen nutzen für die Ereignisdetektion Daten eines Echtzeit-Kurznachrichtendienstes, die eine

geographische Koordinate vom Versendeort besitzen und somit ganz konkret einem Ort zugeordnet werden können. Die geographische Koordinate kommt meist von Applikationen (Apps) auf mobilen Geräten (Smartphones), die mit GPS oder einem vergleichbaren Ortungssystem ausgestattet sind und der zu versendenden Kurznachricht die aktuelle geographische Koordinate optional mitanfügen. Die durch die entwickelten Algorithmen gefundenen Ereignisse und die dazugehörenden Kurznachrichten werden als Ergebnis dem Nutzer, eines im Rahmen dieser Arbeit neu entwickelten Ereignisdetektionssystems, präsentiert. Im Endergebnis wird ersichtlich werden, welche Ereignisse und wo diese gerade stattfinden. Zu jedem Ereignis werden dem Nutzer dazu passenden Kurznachrichten und Mediendaten präsentiert, die das Ereignis genauer beschreiben sollen. Abbildung 2 zeigt beispielhaft einen Screenshot aus der im Rahmen der Arbeit entwickelten Applikation. Das Besondere bei dieser Ereignisdetektion ist, dass die geographischen Koordinaten direkt in die Berechnung zur Ereignisdetektion mit einfließen. Somit ist es möglich, dass für jedes detektierte Ereignis auch sein Ort angegeben werden kann. Damit ist es auch möglich sich nur Ereignisse anzeigen zu lassen, die einen bestimmten Ort betreffen.

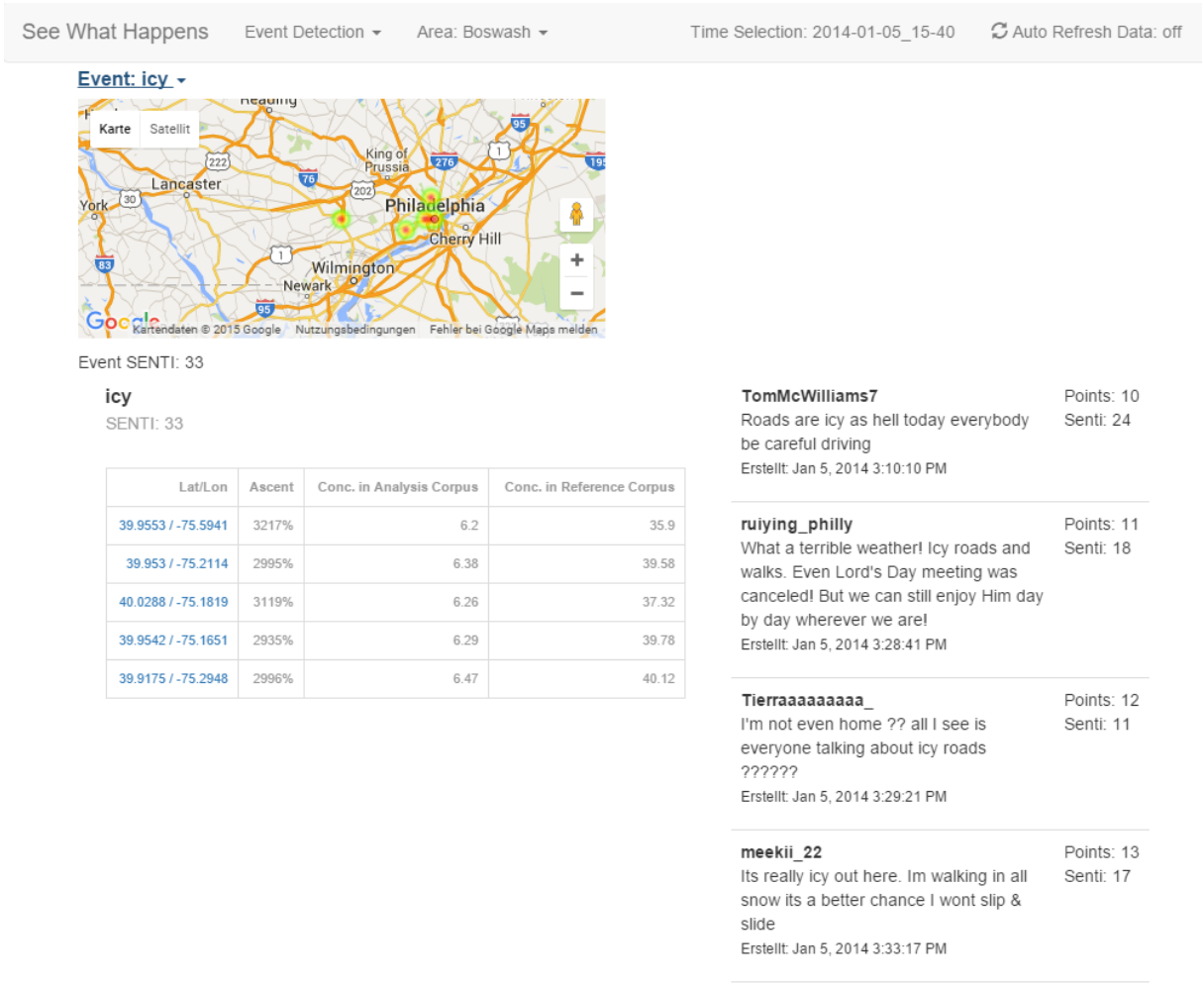

Abbildung 2: Detektierte Ereignisse „icy“ mit Beispiel-Kurznachrichten an der Ostküste der USA während der kalten Wintertage im Januar 2014 an einem Sonntagmorgen (Analyse vom 5.1.2014 15:40 Uhr (GMT +1)).

Mit den neu vorgestellten Algorithmen lassen sich, in abgewandelter Form, auch andere Analysen auf dem Datenstrom durchführen. So wird in einem weiteren Teil gezeigt, dass damit auch eine Stimmungsanalyse (Sentiment-Analyse) möglich ist.

Auch diese Analyse ist sprachunabhängig und ermittelt einen ortsabhängigen Stimmungswert für Terme auf Grundlage ihres gemeinsamen Auftretens mit Emoticons in Kurznachrichten. Weiterhin ist es auch hier möglich ortsabhängige Schwankungen der Stimmung zu erkennen.
Durch diese Analyse ist es möglich, wenn man diese Untersuchungen über einen längeren Zeitraum wiederholt ausführt, auch Änderung dieser Bewertung der Terme zu detektieren.

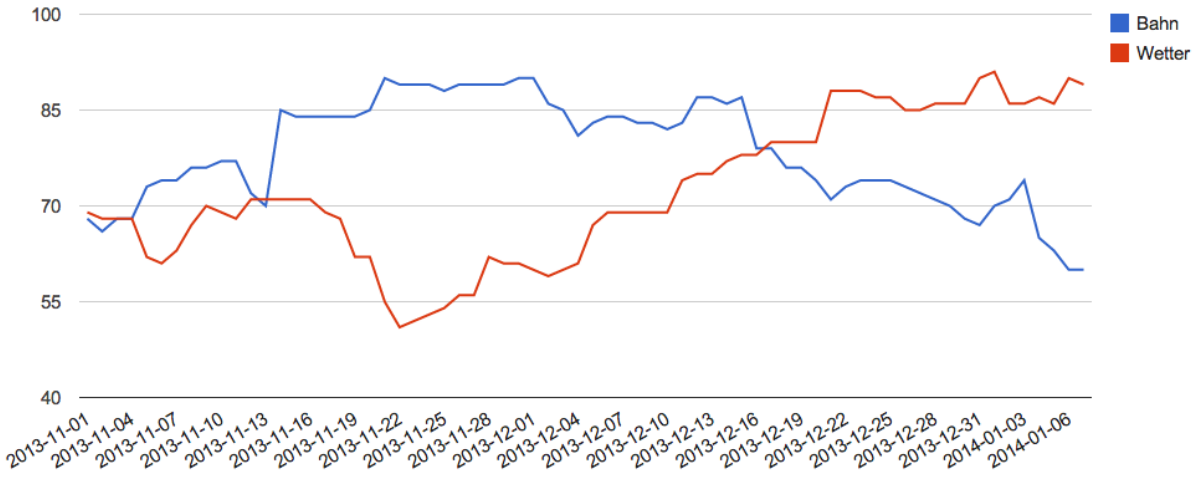

Abbildung 3: Stimmungsverläufe für die Terme "Bahn" und "Wetter" vom 01.11.2013 bis 06.01.2014. Errechnet aus dem Tweets aus Deutschland. [12]

Es ist dann nicht nur möglich zu erkennen, dass sich die Stimmung bei einem Term geändert hat, sondern, durch die Einbindung der geographischen Koordinate während der Berechnung, auch wo sich die Stimmung gegenüber den Term genau geändert hat.
In Abbildung 3 ist der errechnete/ermittelte Stimmungsverlauf für die zwei Terme „Wetter“ und „Bahn“ [12] für den Zeitraum von November 2013 bis Anfang Januar 2014 abgebildet. Der Stimmungsverlauf wurde mit den entwickelten Algorithmen berechnet. Zu sehen ist, dass die Kurve für den Term „Wetter“ ab Mitte November ansteigt. Dies könnte man z.B. mit den steigenden Temperaturen und dem milde verlaufenden Winter in Deutschland, in diesem Zeitraum, erklären. Die fallende Stimmungskurve für den Term „Bahn“ dagegen, fällt seit Ende 2013. Dies könnte einerseits auf Probleme bei der Bahn selbst hindeuten oder auf einen bekanntgewordenen Personalwechsel bei dem Vorstand der Bahn AG.

1.3 Aufbau dieser Arbeit

Das nachfolgende Kapitel 2 behandelt den Stand der aktuellen Forschung im Bereich der Ereigniserkennung. Es geht um die Entwicklungen in diesem Bereich und speziell um die Arbeiten zu der Ereigniserkennung auf den Daten des Kurznachrichtendienstes Twitter bzw. der Analyse von Streaming-Daten und der Einbeziehung von Positionsdaten. Weiterhin erfolgt ein Blick auf die Forschung zu den Stimmungsanalysen ebenfalls auf der Datengrundlage von Kurznachrichten aus Twitter.
Das dritte Kapitel erläutert die neu entwickelten Algorithmen und stellt die Ideen dahinter näher vor. Dabei geht es auch um die Visualisierung der Ergebnisse und die Untersuchung der Ergebnismenge in Abhängigkeit von Parametern in den Algorithmen.
Im vierten Kapitel werden die zuvor entwickelten Algorithmen in einer Anwendung umgesetzt. Es geht darin um die Echtzeitanalyse von einem Kurznachrichten-Datenstrom aus bestimmten zuvor festgelegten Gebieten der Erde. Da die neuen

Algorithmen auch für andere Problemfelder als die Ereignisdetektion eingesetzt werden können, wird die Analysesoftware sowohl eine Ereignisdetektion als auch eine Stimmungsanalyse in diesen Gebieten durchführen und die Ergebnisse kontinuierlich auf einer Webseite (vgl. Abbildung 2) dem Nutzer präsentieren können bzw. bei signifikanten Ereignissen dem Nutzer per Push-Nachricht zu informieren.
Kapitel 5 beschäftigt sich mit der Validierung der Ergebnisse. Hier geht es um die Frage, wie gut die Algorithmen funktionieren, wie man die Leistung der Algorithmen beurteilen kann, wie der adaptive Referenzkorpus und die Ortsabhängigkeit der Korpora das Ergebnis beeinflussen und wie schnell Ereignisse im Gegensatz zu den anderen Nachrichtenquellen detektiert werden können.
Im letzten Kapitel werden diese Ergebnisse diskutiert und es erfolgen ein Ausblick und eine Zusammenfassung der Arbeit.

2 Ereignis- und Stimmungsdetektion in sozialen Netzwerken

In diesem Kapitel soll das Thema Ereignisdetektion, ein Teilbereich des Information Retrieval Forschungsgebietes, einführend erläutert werden. Um das Problem der Ereignisdetektion zu lösen, gibt es diverse Verfahren, wovon einige in den weiteren Kapiteln näher vorgestellt werden. Grob kann man die Verfahren in drei Bereiche aufteilen, wobei es aber auch unzählige Mischformen geben kann:

- Cluster-basierte Ereignisdetektion
- LDA-basierte Ereignisdetektion
- Burst-basierte Ereignisdetektion

Die im Rahmen dieser Arbeit entwickelte Ereignisdetektion, ist dem zuletzt genannten Bereich der Burst-basierten Ereignisdetektion zuzuordnen. Zunächst einmal geht es in den nächsten Abschnitten um die Beschreibung des Forschungsgebietes und die allgemeine Vorgehensweise bei der Ereignisdetektion, welche in der Regel meist eine Cluster-basierte Ereignisdetektion ist. Weiterhin wird neben dem Stand der Technik auf die unterschiedlichen Weiterentwicklungen der Algorithmen eingegangen, um einerseits die Ergebnisse weiter zu verbessern oder die Performance des Systems zu erhöhen. Danach folgt ein Blick auf die Arbeiten zur Ereignisdetektion auf den Daten des Echtzeitkurznachrichtendienstes Twitter und den unterschiedlichen Vorgehensweisen für die Analyse dieser Daten. Da bei Twitter sehr viele solcher nutzergenerierter Daten anfallen und man relativ leicht Zugriff auf diese Daten über eine API (Application Programming Interface) hat, sind diese Daten Gegenstand sehr vieler wissenschaftlichen Untersuchungen. Hier gibt es eine große Anzahl wissenschaftlicher Veröffentlichungen, die diese Daten untersuchen und auch unterschiedliche Inhalte in den Daten detektieren. In dieser Arbeit wurde der Fokus auf die Ereignisdetektion gelegt und so wurden nur Veröffentlichungen betrachtet, die ebenfalls dieses Hauptziel hatten, nämlich neue Ereignisse zu erkennen und diese dem Nutzer zu präsentieren. Der Übergang zur Ereignisdetektion bzw. der Detektion von anderen Inhalten kann dabei sehr fließend sein. So gibt es Arbeiten wie z. B. [13], welche versuchen Veranstaltungen in den Daten zu erkennen indem die Ergebnisse mit einem Veranstaltungskalender abgeglichen werden. In [14] dagegen werden die erkannten Inhalte aus dem Twitter-Datenstrom genutzt, um abonnierte RSS-Feeds neu zu ordnen. Diese Paper ([13], [14]) führen zwar auch eine Art Ereignisdetektion durch haben aber ein anderes Ziel als die Präsentation der gefundenen Ereignisse.

In diesem Kapitel werden auch die Forschungen der Stimmungsanalyse auf den Daten von Twitter (Tweets) näher betrachtet, da später gezeigt werden soll, dass die in dieser Arbeit entwickelten Algorithmen auch für andere Aufgaben z.B. der ortsabhängigen Stimmungsanalyse eingesetzt werden können.

Zuletzt soll das soziale Netzwerk Twitter selbst näher vorgestellt werden, da dieses später als Datenquelle für die praktischen Analysen dienen soll.

2.1 Einführung in die Ereignisdetektion

Das Forschungsgebiet der Ereignisdetektion genauer gesagt des Topic Detection and Tracking (TDT) ist ein Teilbereich des Information Retrieval (IR). Das Fachgebiet des IR (Informationswiedergewinnung) befasst sich mit dem Auffinden von bestehenden Informationen in großen Datenbanken bzw. Datenbeständen (Abbildung 4). [15]

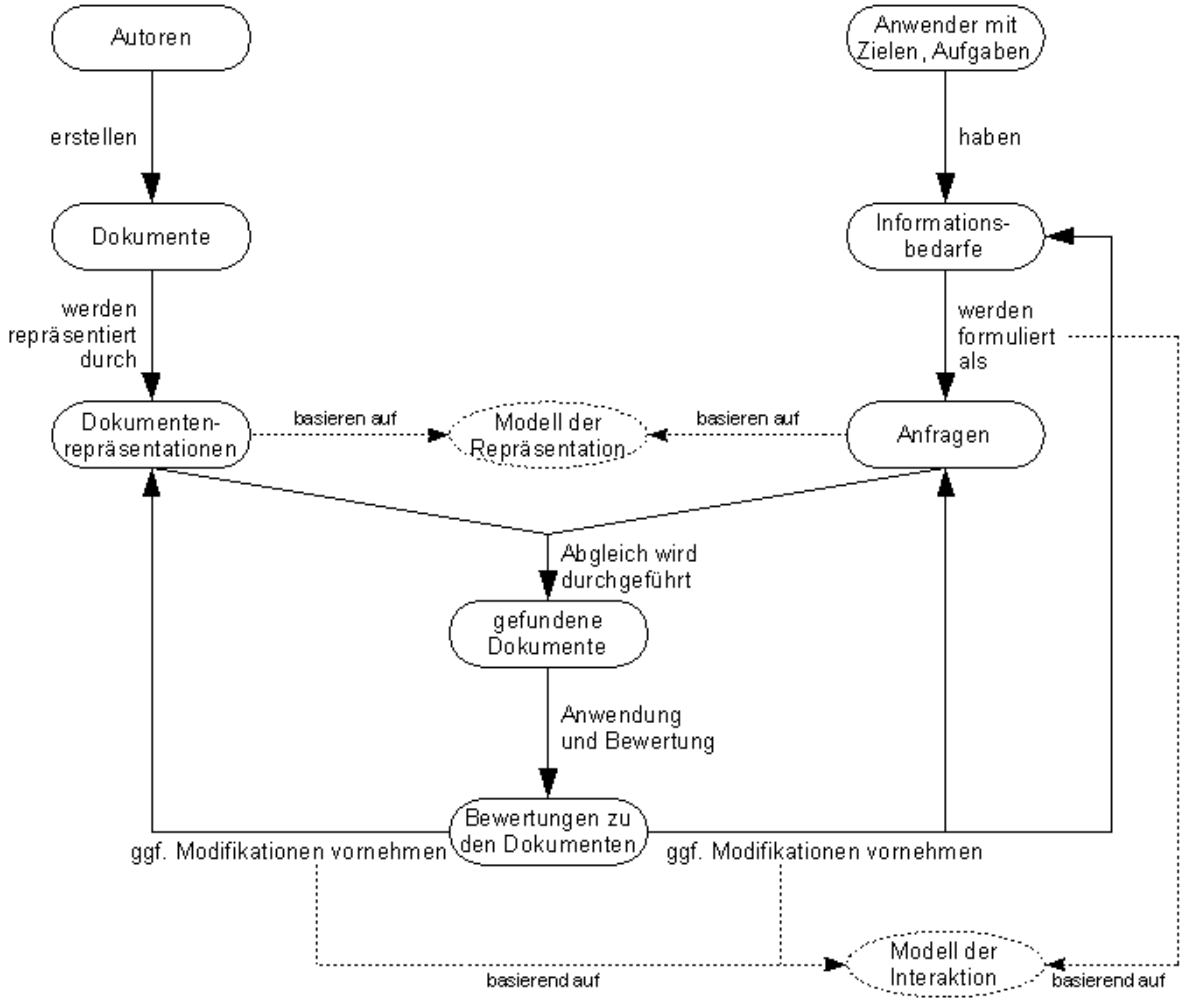

Abbildung 4: Grundlegendes Modell des Information Retrieval [15]

Der Begriff des IR geht auf von Mooers zurück, wo die Techniken des IR in den ersten digitalen Bibliotheken der 50-iger Jahre eingesetzt wurden. [16] Zu dem Gebiet des IR zählen auch die Probleme der effizienten Speicherung und der Zugriff auf diese Informationen. [16] Am bekanntesten ist der Einsatz von IR-Techniken bei den Internet-Suchmaschinen. Dort geht es darum auf eine Suche des Nutzers nach bestimmten Informationen mit einer Liste von möglichst relevanten Webressourcen zu antworten. Dabei ist die Anfrage des Nutzers eher ungenau, da er mitunter nicht exakt weiß nach welchen Termen er genau suchen soll, um die gewünschte Information zu beschreiben. Auf der Seite der Suchmaschine besteht das Problem, dass die Informationen in der Datenbank nicht vollständig sind, da typischerweise nie alle existierenden Webressourcen der Datenbank auch bekannt sein müssen.
Bei dem TDT stellt dagegen der Nutzer nicht eine konkrete Suchanfrage an das System, sondern die Anfrage an das System ist immer präsent, vergleichbar mit einem Abonnement. Die Aufgabe des Systems ist es, neue Themen aus den eintreffenden Daten zu detektieren. Analysiert man einen Strom aus Nachrichtentexten, so stellen diese

Themen Ereignisse dar, die aus diesen Nachrichtentexten erkannt werden. Was die Themen also sind, kommt somit auf die zu analysierenden Daten an. Da in dieser Arbeit nur aktuelle Kurznachrichten analysiert werden, die i.d.R. über aktuelle Geschehnisse handeln, sind die erkannten Themen mit Ereignissen gleichzusetzen. Somit können in diesem Kontext die Begriffe Ereignis (Event) und Thema (Topic) als gleichbedeutend angesehen werden.
TDT wurde durch die Arbeiten von James Allan, Ende der 90iger Jahre, geprägt. Allan definiert den Forschungsbereich folgendermaßen:
„Topic Detection and Tracking (TDT) is a body of research and an evaluation paradigm that addresses event-based organization of broadcast news. The TDT evaluation tasks of tracking, cluster detection, and first story detection are each information filtering technology in the sense that they require that 'yes or no' decisions be made on a stream of news stories before additional stories have arrived" [17], [18]
Dabei beinhaltet TDT mehrere Teilaufgaben [19]. Angenommen wird ein Nachrichtenstrom, der aus Einzelnachrichten (den sogenannten Stories) besteht, die aus dem Internet oder aus anderen Medien stammen können (es kann sich auch um Nachrichtentexte handeln, die aus Radio- oder TV-Sendungen stammen und die mittels Spracherkennungssoftware transkribiert wurden). Dabei werden die Nachrichtentexte in einzelne Nachrichten zerlegt (Story Segmentation), so dass eine Nachricht nur jeweils ein Thema (Topic) bzw. Ereignis beinhaltet.
Im Anschluss daran folgt die eigentliche Ereigniserkennung [19]. Die Ereigniserkennung kann hier auf zweierlei Arten stattfinden. Die erste Art ist die retrospektive Ereigniserkennung (Retrospective Event Detection). Dies bedeutet, dass die zu analysierenden Nachrichten alle bereits zu Beginn in einem sogenannten Text-Korpus (eine Sammlung von Texten) [20] vorliegen und die Ereignisdetektion diesen Korpus analysieren soll, um die einzelnen Ereignisse zu erkennen und die Nachrichten den einzelnen Ereignissen zuzuordnen. Eine weitere Möglichkeit ist die Online-Ereignisdetektion (Online New Event Detection). Hier treffen die einzelnen Nachrichten erst in ihrer chronologischen Reihenfolge ein und sind somit zu Beginn der Analyse nicht bekannt. Bei der Ereigniserkennung ist es nötig, die erste Nachricht zu entdecken, die ein neues Ereignis beschreibt (First Story Detection). Bei jeder weiteren Nachricht die analysiert wird, muss entschieden werden, ob diese selbst wieder ein neues Ereignis ist oder einem bereits erkannten Ereignis zugeordnet werden kann. Dieser Teil wird dann als Ereignis-/Themenverfolgung (Topic Tracking) bezeichnet. Wenn die Zuordnung der Nachricht zu einem Ereignis stattgefunden hat, so können sich noch weitere Aufgaben anschließen, wie z.B. die Benennung des Ereignisses, die Generierung einer Zusammenfassung zu jedem erkannten Ereignis oder die Sortierung der Nachrichten eines Ereignisses nach der Wichtigkeit. [17]
In dieser Arbeit und somit in den nächsten Kapiteln soll es aber vornehmlich um die TDT-Teilaufgabe der Ereignisdetektion gehen.
Bevor die Funktionsweise einer Ereignisdetektion genauer erklärt wird, sollen die zuvor schon verwendeten Begriffe Nachricht bzw. Story und Ereignis näher definiert werden.
In [17] wird eine Story folgendermaßen definiert:
„Eine ‚Story' ist eine abgrenzbare Textstelle (oder ein ganzes Dokument), in der (oder in dem) ein Ereignis besprochen wird."
In [21] wird das Ereignis als „besonderer, nicht alltäglicher Vorgang, Vorfall, Geschehnis" definiert. James Allan definiert ein Ereignis als:
„a set of news stories that are strongly related by some seminal real-world event" [22].
D.h. dass die Ereignisse zu einer bestimmten Zeit und an einen bestimmten Ort geschehen. Diese Ereignisse, die einem bestimmten Ort zugeordnet werden können,

sollen in dieser Arbeit bevorzugt detektiert werden. Dazu werden Nachrichten analysiert, die selbst Ortsdaten von ihrem Sendeort enthalten.

2.2 Allgemeine Vorgehensweise zur Ereignisdetektion

Jede zu analysierende Nachricht wird vor der eigentlichen Analyse aufbereitet. Dabei werden ggf. unerwünschte Terme, wie z.B. Stoppwörter oder andere Inhalte, die keine gültigen Terme sind, entfernt. Die Aufbereitung der Nachrichten ist ein wesentlicher Aspekt der diversen Algorithmen. Hier liegt unter anderem meist der Hauptunterschied zwischen den Verfahren (siehe Kapitel 2.3). Im nächsten Schritt wird aus den Nachrichten ein Vektor extrahiert. Der zu erzeugende Vektor beschreibt in einer konzentrierten Version die spezifische Nachricht und dient dazu die Nachricht mit anderen, bereits analysierten Nachrichten, zu vergleichen (Story Link Detection). Der Vektor beinhaltet die Information, welche Terme in welcher Konzentration in der Nachricht vorkamen und ist eine Art Fingerabdruck der Nachricht. Für die Termgewichtung in dem Vektor wird i.d.R. das TF-IDF-Gewicht (Termhäufigkeit * inverse Dokumenthäufigkeit) eingesetzt [17]. Da die folgenden Berechnungen keine ereignisdetektionsspezifischen Berechnungen sind, sondern Berechnungen, die auch für andere Analysen genutzt werden, wird hier nicht von Nachrichten gesprochen sondern von Dokumenten. In diesem Fall kann man aber den Begriff Nachricht und Dokument gleichsetzen.
Das TF-IDF-Gewicht gewichtet Terme (für eine bestimmte Nachricht) umso stärker, je häufiger sie in dieser Nachricht auftreten (Termfrequenz *tf*) und umso seltener sie in anderen Nachrichten auftreten (Inverse Dokumentfrequenz). [16] Man versucht so einen typischen „Fingerabdruck“ für jede Nachricht zu finden, um die Nachricht gut von anderen unterscheiden zu können. TF-IDF setzt sich aus zwei Teilen zusammen. Der erste Teil ist die normierte Termhäufigkeit eines Terms t_i:

$$tf_{ij} = \frac{Anzahl\ von\ Term\ t_i\ im\ Dokument\ d_j}{Anzahl\ aller\ Terme\ im\ Dokument\ d_j}$$

Formel 1: Normierte Termhäufigkeit

Die Häufigkeit wird deshalb normiert, damit die Textlänge des Dokuments keinen Einfluss auf das Endergebnis hat. Es kommt auf die relative Häufigkeit der Terme in einem Dokument an und nicht auf die absolute. Zu der Formel der normierten Häufigkeit (Formel 1) gibt es auch eine Alternativformel, die das maximale Auftreten eines anderen Terms *w* im Text nutzt, um die Termfrequenz zu normieren:

$$tf(t_i, dj) = \frac{Anzahl\ von\ Term\ t_i\ im\ Dokument\ d_j}{Anzahl\ von\ Term\ w\ im\ Dokument\ d_j}$$

Formel 2: Alternative Berechnung der Termhäufigkeit

Der zweite Teil der Formel ist die inverse Dokumentfrequenz *idf*, die sich folgendermaßen berechnet:

$$idf(t_i) = log\frac{Anzahl\ aller\ Dokumente}{Anzahl\ aller\ Dokumente\ mit\ Term\ t_i}$$

Formel 3: Inverse Dokumentfrequenz

Die inverse Dokumentfrequenz ist umso höher, je seltener der Term in anderen Nachrichten bzw. Dokumenten auftritt.
Das TD-IDF Gewicht w_{ij} für den Term t_i im Dokument d_j ist dann das Produkt der Teilformeln:

$$w_{ij} = tf(t_i, d_j) * idf(t_i)$$

Formel 4: TD-IDF Gewicht

Welche Terme genau in dem Vektor sind ist wiederum unterschiedlich und die entwickelten Algorithmen unterscheiden sich auch hier wieder.
Der erzeugte Vektor der Nachricht wird mit Hilfe des Vektorraummodells mit anderen Nachrichten verglichen. Dabei wird mit Hilfe des Cosinus-Abstands die Ähnlichkeit der Nachrichten mit anderen bereits analysierten Nachrichten verglichen. Ist die Ähnlichkeit unter einer zuvor definierten Schwelle, so stellt die gerade zu analysierende Nachricht ein neues unbekanntes Ereignis dar. Ist die Nachricht gegenüber anderen Nachrichten ähnlich, so beschreiben die Nachrichten das gleiche Ereignis und man hat somit eine neue Nachricht, die zu diesem Ereignis gehört (Topic Tracking).
Als nächster Schritt muss die neue Nachricht einem Ereignis zugeordnet werden (wenn die Ähnlichkeit über der definierten Schwelle lag). Hierbei werden die erkannten Ereignisse ebenfalls als Vektoren im Vektorraum abgebildet. Ein Ereignis ist der Durchschnittsvektor (Zentroid - das geometrische Zentrum eines zwei- oder dreidimensionalen räumlichen Objektes (in diesem Fall kann die Dimension allerdings größer als drei sein)) aller Nachrichten (*Cluster Detection*), die zu diesem Ereignis gehören. [17] Dem Zentroid, dem die neue Nachricht am nahesten ist, kann die Nachricht z.B. zugeordnet werden.
Der Zentroid eines Ereignisses kann auch für weitere Zwecke genutzt werden. So ist es möglich aus ihm einen Titel für das Ereignis zu generieren, indem die Terme mit der höchsten TF-IDF Gewichtung des Durchschnittsvektors genommen werden. Auch eine optionale Zusammenfassung des Ereignisses lässt sich aus den Zentroiden erzeugen [23].

2.3 Modifikationen der Ereignisdetektion

Die Implementationen der einzelnen Ereignisdetektionssysteme sind vielfältig. Einige Forschungsarbeiten modifizieren den Algorithmus z.B. der Vektorbildung oder des Vergleiches im Vektorraum, um die Ergebnisse zu verbessern. Andere Entwicklungen versuchen die Ereignisdetektion performanter zu implementieren oder stellen sich die Frage, wie man mit sehr großen Datenmengen umgehen kann bzw. wie man die Ergebnisse präsentieren kann. Nachfolgend sollen ausgewählte Modifikationen vorgestellt werden.

2.3.1 Modifikationen zur Verbesserung der Ergebnisse

In [17] wird aufgezeigt, dass, um die Ergebnisse zu verbessern, die Terme differenzierter betrachtet werden müssten. Die Terme werden unterschieden in Eigennamen (*Named Entities*) und die restlichen Terme (*Topic Terms*). Nur wenn beide Gruppen von Termen übereinstimmen, beschreiben die zwei unterschiedlichen Nachrichten das gleiche Ereignis. Diese Vorgehensweise begründet Allan folgendermaßen [17]:
„The intuition behind using this features is that we believe every event is characterized by a set of people, places, organizations, etc. (named entities), and a set of terms that describe the event. While the former can be described as the who, where and when aspects of an event, the latter relates to the what aspect. If two stories were on the same topic, they

would share both named entities as well as topic terms. If they were on different, but similar, topics, then either named entities or topic terms will match but not both" [24]
Dieses Vorgehen soll verhindern, dass Nachrichten, die unterschiedliche Ereignisse beschreiben aber doch ähnlich gewichtete Terme in den Vektoren haben, zu einem Ereignis zusammengefasst werden. Erst wenn die Vektoren der Eigennamen, wie z.B. Personen und Ortsnamen, und der Vektor der restlichen Terme sich gleichen, kann von demselben Ereignis ausgegangen werden.
Eine weitere Verbesserung, die [17] beschreibt, ist die Beachtung des Orts- und des Zeitbezuges. Nur wenn die Nachrichten zusätzlich über dasselbe Gebiet/Ort und über dieselbe Zeit berichten, so gehören sie zusammen. Dies wird deutlich bei Ereignissen, die immer Mal wieder auftreten, wie z.B. Flugzeugkatastrophen. Hier ist es wichtig, ob der Ort der Nachricht derselbe ist oder nicht zufällig zwei Katastrophen zur gleichen Zeit stattfanden. Dazu ist es nötig, aus dem Text die benötigten Ortsinformationen herauszufiltern bzw. muss analysiert werden, ob unterschiedliche Ortsangaben dasselbe meinen wie z.B. „Thüringen" und „Erfurt".
Bei der Zeitkomponente ist es das Gleiche. Auch hier müssen aus den zu analysierenden Texten Zeitangaben herausgefiltert werden. Dabei kann eine Zeitangabe sehr unterschiedlich aussehen. Es ist möglich die Zeit absolut anzugeben z.B. „04.08.2013" oder relativ wie z.B. „vorige Woche Mittwoch". [17]
Kann das System diese unterschiedlichen Beschreibungen von Zeiten erkennen und sie auf den gleichen Zeitbereich zuordnen, ist darüber hinaus der Ort der zwei Nachrichten auch der gleiche und damit die erzeugten Vektoren, so kann davon ausgegangen werden, dass die zwei Nachrichten über das gleiche Ereignis berichten.
In [25] wird versucht, mittels *Adaptive Tracking* die Ereignisverfolgung zu verbessern. Es wird beobachtet, ob neue charakteristische Terme bei einem Ereignis hinzukommen. Dies passiert immer, wenn sich ein Ereignis mit der Zeit weiterentwickelt und neue Aspekte bekannt werden. Es tauchen neue Begriffe oder Namen auf, die mit dem Ereignis im Zusammenhang stehen und vorher nicht bekannt waren. Diese neuen Begriffe müssen mit genutzt werden, d.h. müssten im Zentroid des Ereignisses vorhanden sein, damit die nachfolgenden Nachrichten korrekt dem Ereignis zugeordnet werden können.
In [19] wird als zusätzlicher Input mit der Zeit gearbeitet. Also wann einzelne Nachrichten erschienen sind. Alte Ereignis-Cluster [26], also hier Ereignisse, haben weniger Einfluss und fallen aus dem Vektorraum langsam heraus, wenn keine neue Nachricht dazukommen kann. Dies wird deshalb gemacht, da erkannt wurde, dass die Menge an Nachrichten, bei einem typischen Ereignis, sprunghaft ansteigt, um dann binnen 1-4 Wochen langsam auszulaufen. Es gibt aber Ereignisse die langlebiger sind und immer mal wieder neue Nachrichten hinzukommen. Dies ist z.B. der Fall bei einem langwierigen Gerichtsprozess. Je älter das Ereignis ist, umso höher wird z.B. die Schwelle gelegt, wie ähnlich eine neue Nachricht sein muss, damit sie zu diesem Ereignis noch dazu gerechnet werden kann. D.h. bei einem alten Ereignis muss die neue Nachricht, die dazugerechnet werden soll, sehr genau zu dem Zentroid des Ereignis-Clusters passen oder die neue Nachricht wird ggf. selbst zu einem neuen Ereignis, sollte sie die Schwelle zu einem anderen Ereignis-Cluster auch nicht überschreiten können.
Mit Hilfe eines sogenannten *Look-Back Window* [19] ist es möglich, herauszufinden, ob zwei Nachrichten über das gleiche Ereignis berichten. Je älter die Nachrichten in diesem Fenster sind, umso weniger werden sie gewichtet. Verglichen werden die Nachrichten hier z.B. ob sie lexigraphisch ähnlich sind. Wenn ja werden die Nachrichten demselben Ereignis zugeordnet.
Eine weiterer Ansatz in [19] überprüft, ob die Termhäufigkeit an sich in den Nachrichten verändert ist. Das bedeutet aber, dass man zuvor die Standardtermhäufigkeiten in den

Texten kennen muss. Man würde so Termhäufigkeitsabweichungen der Nachrichten erkennen und somit auf ein neues Ereignis schließen.

2.3.2 Modifikationen zur Verbesserung der Performance

Die Beachtung des Veröffentlichungszeitpunktes der Nachrichten hat nicht nur Auswirkungen auf die Qualität der Ergebnisse, sondern verbessert auch die Performance des Ereignisdetektionsschrittes. Lässt man alte Nachrichten bzw. Ereignisse weg, muss man die Vektoren der neuen Nachrichten nicht mehr mit diesen vergleichen. Man reduziert somit die Elemente im Vektorraum und somit die notwendigen Vergleiche, um zu bestimmen, ob die neue Nachricht ein neues Ereignis ist oder einem bereits existierenden Ereignis hinzugefügt werden kann. Besonders bei der Online-Analyse von Nachrichten ist der Punkt Performance ein wichtiger Aspekt. Hier kann pro Sekunde eine große Anzahl von Nachrichten auf das System eintreffen, die schnellstmöglich analysiert und eingeordnet werden müssen.
Eine bereits erwähnte Möglichkeit ist der Einsatz von *Look-Back Windows* oder *Sliding-Windows* genannt. In Abbildung 5 ist solch ein Sliding-Window schematisch dargestellt.

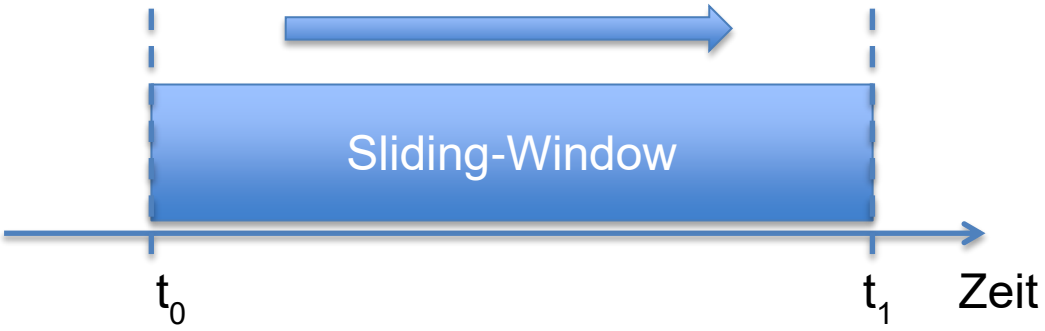

Abbildung 5: Sliding-Window

Dieses verschiebbare Fenster betrachtet nur die Nachrichten, die innerhalb eines bestimmten Zeitbereiches erstellt wurden. In den meisten Fällen reicht bei einer Online-Analyse das Fenster von dem gegenwärtigen Zeitpunkt t_1 bis ein paar Tage in die Vergangenheit zu t_0. Alles was vor der zeitlichen Grenze t_0 liegt, wird i.d.R. nicht mehr betrachtet. Dadurch schiebt sich das Fenster immer mit und es kommen neue Nachrichten herein und es fallen zur gleichen Zeit alte Nachrichten am Ende wieder heraus. Man kann auch, wie schon erwähnt, das verschiebbare Fenster dazu nutzen auch innerhalb des Fensters die Nachrichten oder Ereignis-Cluster unterschiedlich zu gewichten indem alte Ereignis-Cluster abschwächend gewichtet werden, so dass sie immer weniger Einfluss auf dem Vektorraum haben. Z.B. führt die Gewichtung dann dazu, dass neue Nachrichten nur zu einem alten Ereignis-Cluster noch hinzugeordnet werden können, wenn sie eine sehr hohe Ähnlichkeit haben. [27] Eine weitere Möglichkeit die Performance zu verbessern bietet sich, wenn man die Breite des verschiebbaren Fensters dynamisch verändert. [28] Bei großer Last auf einem Online-System, wird die Breite des Fensters verringert (das Fenster ragt nicht so weit in die Vergangenheit) was die Menge der Nachrichten verringert und somit die Vergleiche im Vektorraum reduziert. Natürlich ist damit auch mit einer potentiellen Verschlechterung der Ergebnisse zu rechnen, wenn die erkannten Ereignisse schneller entfernt werden.
Um die Vergleiche im Vektorraum noch weiter einzuschränken, ist es auch möglich die neu ankommende Nachricht nur mit den ersten Nachrichten eines Ereignisses zu vergleichen und nicht mit jeder Nachricht des Ereignis-Clusters selbst. [28] Weiterhin können die Vergleiche beschleunigt werden, indem die Vektoren selbst kürzer sind. So

wird beschrieben, dass z.B. nur die ersten 10 Top-Terme einer Nachricht verglichen werden und der Rest ignoriert wird. Ein weiterer Schritt ist, dass zuvor überprüft wird, ob ein Vergleich überhaupt sinnvoll erscheint. Dies wird realisiert, indem man überprüft, ob die Terme überhaupt in dem Ereignis-Cluster vorkommen.
Neben den genannten Verfahren, um die Vergleiche zu beschleunigen, beschreibt [28] auch, dass man durch Parallelisierung und geeigneten Datenstrukturen die Algorithmen weiter beschleunigen kann. Doch wenn dies alles immer noch nicht ausreichend ist, müssen die neuen Nachrichten zuvor nach ihren Quellen gewichtet werden, um die Nachrichten-Anzahl temporär zu begrenzen. Sinnvoll ist es auch die Quellen dahingehend weiter zu analysieren, ob aus diesen Quellen z.B. viele *First Stories*, die also häufig die ersten Nachrichten von neu erkannten Ereignissen sind, herkommen. Wenn man solche Quellen hat, so ist es sinnvoll in Lastspitzen diese Quellen gegenüber anderen Quellen zu bevorzugen.
Doch nicht nur die Datenmenge für die Analyse kann zu groß sein, sondern auch die Ergebnismenge für den Nutzer. Wenn ein Nutzer die Ergebnisse der Ereignisdetektion präsentiert werden soll, so muss das Ergebnis so präsentiert werden, dass er schnell die wichtigsten Ereignisse erkennen kann. Z.B. sollten die erkannten Ereignisse nach ihrer Wichtigkeit oder einem anderen Schema geordnet werden, welches der Nutzer benötigt. Sollte es aber zu viele neue Ereignisse geben, so dass der Nutzer überfordert wird, so kann man auch hier die Ergebnisse auf die wichtigsten einschränken. [28]
Die erwähnten Änderungen zur Performancesteigerungen beeinflussen natürlich das Ergebnis der Ereignisdetektion. Hier ist abzuwägen zwischen Performance und Qualität der Endergebnisse. Um die Qualität zu beurteilen, kann man z.B. mit zuvor definierten Testdaten arbeiten. Dies kann ein Korpus von Nachrichten sein, der z.B. von Hand zuvor bearbeitet wurde und zu jeder Nachricht bekannt ist, über welches Ereignis berichtet wird. [25] Weiterhin existieren Benchmarks, wie z.B. TDT5 [28], welche einen definierten Korpus von Nachrichten enthalten. Dieser Korpus kann zu Testzwecke analysiert werden. Damit wird es möglich, bei verschiedenen Algorithmen die Qualität bzw. den Einfluss von Performance steigernden Maßnahmen zu beurteilen.

2.4 Ereignisdetektion in den Daten der sozialen Netzwerke

In den Anfangsjahren des TDT-Forschungsgebietes ging es, mangels einer großen Anzahl alternativer Texte, hauptsächlich um die Ereigniserkennung in Nachrichtentexten. Diese Nachrichtentexte waren z.B. Meldungen aus Zeitungen, Fernsehnachrichten oder anderen Nachrichtenquellen. Doch im Zuge des Web 2.0 [29] und der zunehmenden Bedeutung des „User Generated Contents“ existiert nun eine fast unerschöpfliche Quelle von unterschiedlichen Texttypen und Inhalten die analysiert werden können. In diesen Inhalten z.B. der sozialen Netzwerke stecken all die Informationen was gerade geschieht, da Milliarden von Menschen (allein Facebook hat > 1 Mrd. Nutzer) soziale Dienste zurzeit nutzen und beständig neue Inhalte posten. Diese neuen Quellen sind aktueller und potentiell ergiebiger als die zuvor betrachteten Nachrichtentexte, wenn es darum geht gerade geschehende Ereignisse zu detektieren. [30] Durch eine Ereignisdetektion lassen sich aus dieser großen Menge an Texten, die Ereignisse herausfinden, über die gerade berichtet wird. Da die Nutzer der sozialen Netzwerke wie z.B. Twitter über alle möglichen gerade stattfindende Ereignisse in Echtzeit berichten, kann man mit Hilfe der Ereignisdetektion potentiell schneller neue Ereignisse erkennen und darüber schneller berichten, als dies etablierte Nachrichtenmedien können. In den nachfolgenden Betrachtungen, geht es vor allem um das soziale Netzwerk Twitter. Twitter wurde zum einen gewählt, da dieses Netzwerk eine hohe Verbreitung besitzt und genutzt wird, um Inhalte für die Allgemeinheit

zugänglich zu machen. Zum anderen kann man auf diese Inhalte über eine bereitgestellte API relativ leicht zugreifen. Somit ist Twitter Gegenstand einer großen Anzahl wissenschaftlicher Untersuchungen im Bereich der Ereignisdetektion oder anderen wissenschaftlichen Bereichen geworden, in denen es darum geht, Informationen aus Texten zu extrahieren. So gibt es z.B. Arbeiten, die es ermöglichen Zeiten aus Tweets (z.B. „Wir treffen uns nächste Woche Donnerstag.“) zu extrahieren (HeidelTime [31] [32] [33]) und diese in standardisierten Formaten (TIMEX3-Format) zu hinterlegen, um damit weiterarbeiten zu können. Im Weiteren soll es aber um die Ereignisdetektion gehen. Diese Untersuchungen sollen im nachfolgenden genauer vorgestellt werden, um die angewendeten Algorithmen besser kennenzulernen.

2.4.1 Ziele

Bei den wissenschaftlichen Untersuchungen zur Ereignisdetektion, unter der Nutzung der Twitter-Daten, geht es meist nicht einfach nur darum Ereignisse zu erkennen und auf einer Webseite zu präsentieren, sondern die Arbeiten beschreiben vielfältigste Nutzungsszenarien, wovon hier einige vorgestellt werden sollen, damit die vielfältigen Möglichkeiten deutlich werden.

In [34] und [35] wird ein System vorgestellt, welches Terme (dort *Keywords* genannt) und ihre Nutzungshäufigkeit untersucht. Somit kann erkannt werden, ob bestimmte Terme zu einem Zeitpunkt häufiger genutzt werden. Diese speziellen Terme (dort *Hot Keywords* genannt) werden dann dem Nutzer des Systems präsentiert und man kann so ablesen über was zu einem bestimmten Zeitpunkt geschrieben wurde. Als Datenquelle dienen hier neben Twitter auch die Daten der Blogossphäre (Gesamtheit von Blogs, die unter einander auch vernetzt sein können). Über die Angaben zum Autor des Blogs wird versucht zu orten, wo der Blog-Post geschrieben wurde, was aber nur dann korrekt gelingt, wenn der Blog die entsprechenden Metadaten auch besitzt. Die ermittelten Daten werden kommerziell verwendet. So können z.B. andere Firmen diesen Dienst nutzen, um zu analysieren ob z.B. eine Werbekampagne ein messbares Echo in den untersuchten Quellen ergeben hat. Ebenso ist es möglich zu erkennen, ob über ein bestimmtes Produkt oder Service der jeweiligen Firma vermehrt geredet wird. Dargestellt werden die Ergebnisse auf einer Webseite. Die Häufigkeit der Terme wird in Diagrammen dargestellt wo Bursts (gebündeltes Auftreten eines Ereignisses) erkennbar werden. Von dort aus können mit zusätzlichen Termen weitere Untersuchungen stattfinden und man kann sich andere Terme anzeigen lassen, die ebenfalls zu einer bestimmten Zeit eine hohe Häufigkeit hatten. Wenn der Ort bekannt ist, lassen sich die Terme auch auf einer Karte darstellen. Auch die Anzeige der betreffenden Blogeinträge, in denen die Terme auftauchen, ist möglich. Somit wird klar, über was genau zu welchem bestimmten Zeitpunkt geschrieben wurde. Neben der Analyse über einen Term lassen sich auch alle *Hot Keywords* einer bestimmten Region von einem bestimmten Zeitpunkt (z.B. gestern) anzeigen. Zu guter Letzt lassen sich nutzerdefinierte Alarme einstellen, um informiert zu werden, wenn die Nutzung bestimmter Terme stark zunimmt.

In [36] wird ein System beschrieben welches Eilmeldungen (Breaking News) erkennen soll und auf einer Webseite darstellen soll. Dabei werden die erkannten Ereignisse auf einer Karte verzeichnet. Der Nutzer kann sich in der Karte frei bewegen und somit die aktuellen Ereignisse einer bestimmten Region sehen. Auch ein Benachrichtigungssystem ist hier enthalten, welches selbst wieder Tweets versendet wenn ein bestimmtes, zuvor definiertes, Ereignis (z.B. Politik, Sport, Krankheitsausbrüche, News über Produkte und deren Rückrufe usw.) geschieht.

In [37] wird die Ereignisdetektion dagegen nur genutzt um mehr Informationen über zuvor bekannte stattfindende Ereignisse zu sammeln. Fand z.B. ein Festival statt, so wird

für diesem Zeitraum gezielt nach Tweets gesucht. Somit möchte man mehr Informationen zu einem stattgefundenen Ereignis bekommen.
Ein semi-automatisches Analysesystem wird in [38] vorgestellt. Der Nutzer gibt hier z.B. auch die Anzahl der gewünschten Ereignisse vor, die man als Ergebnis möchte. Weiterhin werden die Zeitspanne und die Region ausgewählt, aus der man die Tweets analysieren möchte. Das System analysiert daraufhin mit den entsprechenden Filtern und ordnet die Tweets den Ereignissen zu. Der Nutzer hat durch die Angabe der Anzahl der gewünschten Ereignisse die Möglichkeit, die Granularität der erkannten Ereignisse zu steuern. Gibt er eine hohe Zahl an, so werden auch kleinere Ereignisse detektiert, die hingegen bei einer kleineren Anzahl von Ereignissen vielleicht zu einem Ereignis zusammengefasst worden wäre. Der wichtigste Teil des Systems ist hier die Oberfläche, worüber der Nutzer immer neue und gezieltere Anfragen stellen kann bzw. die Regionen eingrenzen kann, um die gewünschten Ergebnisse zu bekommen. D.h. dass das System hier nicht in Echtzeit arbeitet, sondern dazu da ist, Ereignisse in der Vergangenheit genauer, mit Hilfe der Tweets, zu analysieren und Ereignisverläufe und deren Regionen sichtbar zu machen.
In [39] nutzt man die Ereignisdetektion, um Festivals zu erkennen. Man erkennt, dass es in der Region eine erhöhte Aktivität gibt und analysiert zusätzlich die Bewegungen in der Region und gleicht dies parallel mit einem Festivalplan ab. Somit ist es möglich, bestimmte Tweets zu bestimmten Festivals/Großereignissen zuzuordnen. Auch in [40] analysiert man die Bewegungen und versucht mit statistischen Methoden die räumliche und zeitliche Verbreitung von Ereignissen zu untersuchen bzw. die Veränderung derer im Laufe der Zeit. So ist es auch möglich, die Wohnorte und Arbeitsplätze der Nutzer zu differenzieren und Freundeskreise zu erkennen. Auch hier ist eine Erkennung von großen Events möglich. Es wird dort auch gezeigt, dass man die Art des Events unterscheiden kann, entweder ein Event was lokal stattfindet oder ein Medienevent was zeitgleich überall im Land stattfindet z.B. eine neue Produkteinführung.
Einer der vielleicht beeindrucktesten Anwendungsfälle für eine Ereignisdetektion aus Twitter-Daten zeigt [41]. Dort versucht man Naturkatstrophen, wie Erdbeben oder Taifune für ein bestimmtes Gebiet, hier Japan, frühzeitig zu detektieren, um die Bevölkerung zu alarmieren. In diesem Fall ist das Ereignis zuvor bekannt (Erdbeben und Taifun) und man versucht herauszufinden, wann die Twitter-Nutzer aus der Region vermehrt darüber schreiben. Man hat Japan als Testregion gewählt, da es hier eine große Anzahl von Twitter-Nutzern gibt, also viel Material vorliegt und die Region wird in Regelmäßigkeit von Naturkatastrophen heimgesucht. Tatsächlich kann durch das System schneller ein Erdbeben detektiert werden und die Bevölkerung alarmiert werden als das offizielle Erdebenwarnsystem. Zudem kann die Wegstrecke eines Taifuns ebenfalls bestimmt werden. Angewandt wird dieses System als Informationssystem für Erdbeben und Taifune. Man kann sich als Nutzer bei dem System registrieren und wenn ein Erdbeben oder Taifun registriert wird, welches auf den eigenen Standort zusteuert, wird man vom System informiert, um sich auf die bevorstehende Naturkatastrophe vorzubereiten.
Das Thema Katastrophen haben auch andere wissenschaftliche Arbeiten, wie z.B. [42], wo es darum geht, aufkommende Term-Bursts zu erkennen, um im Katastrophenfall schneller an Informationen zu kommen bzw. neue Ereignisse frühzeitig zu detektieren. Auch eine Analyse im Nachgang einer Katastrophe, wie in [43], kann neue Erkenntnisse zum Ablauf der Katastrophe bringen. Dort geht es um einen Waldbrand und es wird mit Hilfe der Ereignisdetektion untersucht, wie sich die Nachricht genau ausgebreitet hat und welche Medien als erstes darüber berichtet haben. Da der Brand in einer abgelegenen Region stattfand, fand man die ersten Meldungen nicht über Twitter sondern sie kamen von den lokalen Medien. Erst dann berichteten erste Twitter-Nutzer

von dem Geschehen. Danach konnte beobachtet werden, wie die Nachricht per Re-Tweets (weiterleiten der Nachricht an Freunde) weiter über Twitter verbreitet wurde.
Weitere Arbeiten [44] fassen das Thema etwas weiter zu „Crime and Disaster related Events“ (CDE) und analysieren die räumliche und zeitliche Ausdehnung und deren Wichtigkeit von solchen Ereignissen mit Hilfe der Twitter-Daten. Hier sind die gewünschten Ereignisse, die man entdecken will, thematisch auf CDE eingeschränkt. Neben der Darstellung der aktuellen CDE-Ereignisse, geht es auch darum, CDE-Ereignisse der Vergangenheit zu analysieren. Z.B. kann man an das System Anfragen der Art stellen: „Zeige mir alle Autounfälle im Juli“. Neben der Darstellung der gefundenen Ereignisse auf einer Karte, wird auch der zeitliche Verlauf dargestellt, wie häufig dieses Ereignis an den bestimmten Zeitpunkten auftrat (Abbildung 6).

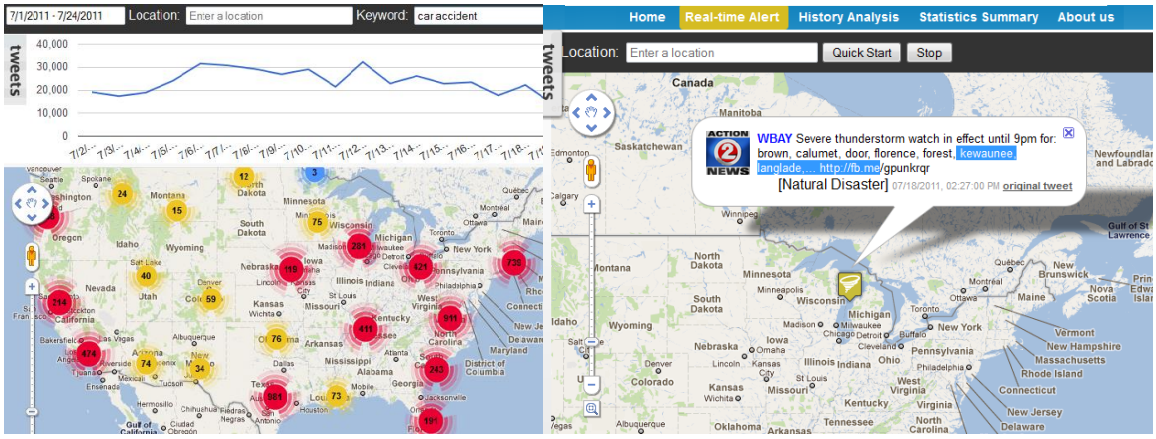

Abbildung 6: Muster der Autounfälle und die Entdeckung eines neuen CDE-Ereignisses aus [44]

Der Letzte Anwendungsfall, der hier kurz vorgestellt werden soll, ist die allgemeine frühzeitige Erkennung von Ereignissen aus den Twitter-Daten. Hier geht es darum allgemein erkannte Ereignisse zu detektieren, d.h. es können Katastrophen sein oder auch andere Ereignisse, die erkennbar werden. Dies findet man z.B. in [45], wo die erkannten Ereignisse auch auf einer Karte gezeigt werden und z.B. Erdbeben, Hurrikans und Aufstände erkannt werden können. In [46] wird gezeigt, dass man mit dieser Ereigniserkennung sogar schneller als die etablierten Medien (2h-3h) neue Ereignisse wie z.B. Erdbeben aus den Daten erkennen kann.
Dies sollte nur ein kurzer Überblick sein, welche verschiedenen Anwendungsfälle durch eine Ereignisdetektion mit Hilfe der Twitter-Daten möglich sind. Durch die einfache Möglichkeit per API an die Twitter-Daten zu kommen entstehen laufend immer neue wissenschaftliche Arbeiten mit neuen Anwendungsfällen oder auch neue kommerzielle Angebote die auf den Daten basieren.

2.4.2 Ereignisdetektionsalgorithmen

In den folgenden Abschnitten geht es um die eigentlichen Algorithmen der Ereignisdetektion, die für die Analyse der Twitter-Daten bisher eingesetzt werden. Dabei werden die Systeme betrachtet, deren Anwendungsgebiete im vorrangegangenen Kapitel vorgestellt wurden, aber auch noch nicht vorgestellte wissenschaftliche Arbeiten. Es wurde weiterhin versucht die Vorgehensweise grob zu ordnen in Cluster basierte Algorithmen, LDA basierte Algorithmen, Bursterkennung und als letztes in Algorithmen, die den Ort näher einbeziehen in die Analyse. Eine vollständige Trennung ist aber nicht möglich, da die Übergänge mitunter fließend sind. Grob lässt sich aber sagen, dass die Analysen immer aus zwei Phasen bestehen. Phase 1 bereitet die Daten auf und entfernt einerseits ungewollte Terme oder Tweets die als Spam oder auch als

Noise bezeichnet werden. Was oder wie viel so herausgefiltert wird, ist je nach Vorgehensweise unterschiedlich. Neben dem Entfernen von vermeintlich unwichtigen Bestandteilen, werden die restlichen Daten z.T. noch in die richtige Form gebracht. Dies kann z.B. ein Stemming sein, also eine Stammformreduktion, indem die verschiedenen morphologischen Varianten eines Terms auf einen Wortstamm zurückgeführt werden [47]. In der zweiten Phase findet dann die eigentliche Ereignisdetektion, mit den nun aufbereiteten Daten, statt. Mitunter werden die gefundenen Ereignisse danach ebenfalls noch einmal überarbeitet, wenn man z.B. nur bestimmte Ereignisse detektieren möchte.

2.4.3 Cluster-basierte Ereignisdetektionsalgorithmen

Das generelle Vorgehen der Ereignisdetektion, welches bereits in Kapitel 2.2 beschrieben wurde und deren Modifikationen aus Kapitel 2.3, lassen sich auch in den Arbeiten wiedererkennen und wurden an die Begebenheiten der zu analysierenden Daten angepasst und erweitert.

Als Beispiel, wie die Ereigniserkennung durchgeführt wird und welche Modifikationen vorgenommen wurden, soll folgendes Beispiel dienen. Das Ziel des Systems in [36] ist es, die neuesten Ereignisse zu detektieren und darzustellen. Dazu wird die Streaming-API von Twitter genutzt. Im ersten Schritt werden die ungewünschten Tweets, die als Noise bezeichnet werden, herausgefiltert. Man möchte nur Tweets haben, die Nachrichten beinhalten. Dazu hat man einen statischen Korpus von Tweets, welche Nachrichten enthalten. Dieser Korpus und zuvor bereits erkannte Tweets mit Nachrichten, werden dazu genutzt, um einen Bayes-Filter (der Bayes-Filter ist ein statistischer Filter, der nach dem englischen Mathematiker Thomas Bayes (ca. 1702-1761) benannt worden ist [48]) zu trainieren. Der so trainierte Filter, soll die einkommenden Tweets (über charakteristische Wörter) klassifizieren in News-Tweets oder Junk (Müll). Mit den klassifizierten News-Tweets wird wieder ein dynamischer Korpus erzeugt, der dann wieder als Trainingsgrundlage für den Bayes-Filter dient, um die zukünftigen Ergebnisse der Klassifizierung weiter zu verbessern.

Die erkannten News-Tweets gelangen dann zu der Online-Clusterung. Es wird, wie bereits in Kapitel 2.2 beschrieben wurde, ein Vektor (Feature Vector) gebildet, aus den zuvor mit TF-IDF gewichteten Termen. Dabei wird auch darauf geachtet, dass mehr als ein Term im Feature-Vektor enthalten sein muss, bevor er weiter verarbeitet wird. Des Weiteren werden Terme, die immer zusammen auftreten, wie z.B. „San Fransicso“ zu einem Term zusammengezogen, so dass der Begriff nicht mehrfach im Feature-Vektor auftaucht. Mit Hilfe des Feature-Vektors und den Ereignis-Cluster-Zentroiden wird der Cosinus-Abstand berechnet. Liegt der Abstand unter einer bestimmten Schwelle, so kann der Tweet zu diesem Ereignis-Cluster hinzugezählt werden. Wenn es zu keinen Ereignis-Cluster passt, so ist es ein neues Ereignis. Ist der Vektor einmal einem Ereignis zugeordnet, bleibt er für immer dort. Zusätzlich wird bei der Berechnung des Cosinus-Abstand auch das Alter des Ereignis-Clusters mit eingerechnet, nämlich über eine Gewichtung mit Hilfe einer Gauß-Funktion. D.h. der Abstand zu einem älteren Ereignis-Cluster muss schon sehr gering sein, damit der Tweet einem älteren Ereignis mit eingerechnet werden kann. Bei einem Alter von 3 Tagen wird hier der Ereignis-Cluster sogar komplett inaktiv geschaltet und kann somit aus dem Ereignis-Cluster-Raum @entfernt werden. Eine weitere Besonderheit der Ereignisdetektion, neben der Noise-Filterung mittels Bayes-Filter in [36], ist die unterschiedliche Betrachtung der Tweets. Bei der Analyse sind nicht alle Tweets gleichbedeutend sondern es wird nach Quelle unterschieden. Manche Twitter-Nutzer werden als „Seeder“ (ein Begriff aus dem File-Sharing-Bereich der für Nutzer steht die vorrangig Dateien zur Verfügung stellen anstatt zu laden) betrachtet. Nur Tweets von Seeders können Ereignis-Cluster bilden oder aber auch erst andere Ereignis-Cluster aktiv schalten, sollten neue Ereignis-Cluster von

anderen Nutzern (Nicht-Seeder) gebildet worden sein. Da die Seeder-Liste wird manuell gepflegt wird, ist die gesamte Ereignisdetektion in [36] stark von der Seeder-Liste abhängig. Die Seeder-Liste stellt somit eine Liste von Twitter-Nutzern dar, denen man vertraut bzw. die, in den Augen der Betreiber, eine gute und sichere Quelle für News-Tweets sind. Um die Seeder-Liste immer aktuell zu halten bzw. sie auch zu erweitern, sucht man neue potentielle Seeder bei den Twitter-Nutzern, denen die Seeder folgen. Ein weiterer manueller Eingriff in die Ereignisdetektion wird zum Zwecke der „Pflege“ des Vektorraumes durchgeführt. Man geht manuell die gebildeten aktiven Ereignis-Cluster durch und markiert ggf. auftretende Ereignis-Cluster-Kopien. Dies sind gleiche Ereignisse, die aber mehr als einen Ereignis-Cluster im Vektorraum haben. Die erkannte Kopie wird entweder inaktiv geschaltet oder aber einem anderen Ereignis-Cluster hinzugeordnet.
Um die erkannten Ereignisse einem Ort zuzuordnen, versucht man den Ort des Ereignisses zu finden. In [36] nutzte man dazu nicht die geographischen Daten, die ein Tweet enthält, da diese zu dieser Zeit noch nicht in der API existierten. Somit musste man sich auf die Ortsangabe des Twitter-Nutzers selbst verlassen. Bei der Registrierung eines Twitter-Nutzers kann dieser seinen Ort angeben. Es ist schnell ersichtlich, dass diese Angabe mitunter nichts mit einen Ereignis zu tun haben muss. Wenn sich z.B. der Nutzer auf Reisen befindet, so kann der hinterlegte Heimatort des Nutzers sehr verschieden zum tatsächlichen Aufenthaltsort und somit zum Sendeort des Tweets sein. Somit versucht man den Ort auch noch aus den Tweet-Inhalten herauszubekommen. Die ganzen vermutlichen Orte werden gewichtet und dieser Ort dann auf einer Karte mit angezeigt. Um noch mehr Tweets zu einem bestimmten Ereignis zu bekommen, werden weiterhin in [36] die Hashtags extrahiert und mit diesen versucht neue Tweets des gleichen Ereignisses zu finden.
Anhand dieser wissenschaftlichen Arbeit kann man erkennen, an welchen Stellen die Ereignisdetektion modifiziert werden kann. Es wurde bei der Filterung der einkommenden Tweets ein Bayes-Netzwerk eingesetzt, um nur bestimmte Arten von Tweets zu erhalten. Es gibt in diesem Beispiel viele Stellen wo manuell eingegriffen wird, sei es bei dem Trainieren des Bayes-Netzwerkes, indem man einen Korpus bereitstellt oder aber auch den nächsten Schritten, indem man den Vektorraum manuell pflegt, indem man doppelte Ereignis-Cluster entfernt. Auch der eigentliche Ereignis-Clusterschritt wurde angepasst, indem nicht alle Daten als gleichwertig angesehen wurden, sondern die Autoren der Tweets in zwei Gruppen eingeteilt wurden. Die Einteilung der Gruppen geschieht auch wieder manuell und nur Autoren aus der manuell ausgewählten Gruppe sind vertrauenswürdig und deren Tweets können neue Ereignis-Cluster erzeugen oder bestehende Ereignis-Cluster freischalten.
Der Einsatz von Klassifikatoren findet nicht nur bei der Filterung von Eingangsdaten Verwendung, sondern auch bei bzw. nach der Ereignis-Clusterbildung. So setzt [49] einen Klassifikator ein, um die gebildeten Ereignis-Cluster, also die erkannten Ereignisse, zu bewerten, ob es Real-World-Ereignisse sind oder nicht. Dazu werden zeitliche Eigenschaften, wie das zeitliche Auftreten von Tweets, die zu diesen Ereignis-Cluster gehören, ausgewertet. Es wird so überprüft, ob ein bestimmtes Muster bei dem Erscheinen der Tweets auftaucht, z.B. ein plötzlicher Anstieg von Tweets zu diesem Event. Weiterhin werden soziale Eigenschaften überprüft, ob z.B. die Tweets viele Re-Tweets erhalten oder nicht. Danach werden noch topologische und „Twitter-Centric“ Eigenschaften ausgewertet. Z.B. ob bestimmte Terme auftreten, die darauf hindeuten könnten, ob es ein Real-World-Event ist oder eher nicht. Alle diese Eigenschaften gehen in den Klassifikator mit ein und dienen als Grundlage zur Beurteilung, ob der Ereignis-Cluster ein Real-World-Event beschreibt oder nicht.

Die Eingangsdaten zur Ereignisdetektion bzw. die Ergebnisse können auch anders bearbeitet werden. In [50] werden z.B. Tweets über die Streaming API abgefragt, die bestimmte Terme enthalten, z.B. den Tag „#breakingNews“. Ziel ist hier, ein System zu bauen, was Breaking News, also Eilmeldungen, erkennen kann. Somit werden die Eingangsdaten stark eingeschränkt, auf wenige Terme, die man vorgibt. Es ist zudem fraglich, ob Nutzer, die einen gerade stattfindenden Ereignis beiwohnen, diesen Hashtag auch nutzen. Eher wahrscheinlich ist, dass dieser Tag von Nachrichtenagenturen bzw. Medien verwendet wird, um auf ein Ereignis hinzuweisen. Doch wenn man die Tweets erst von den etablierten Medien bekommt, dann kann man an sich nicht schneller reagieren, als die Medien selbst neue Ereignisse erkennen.
Bei der Berechnung des TF-IDF Wertes, werden in [50] zusätzlich noch Eigennamen stärker gewichtet, um diese Begriffe hervorzuheben. Dazu ist natürlich eine zusätzliche Datenbank von Nöten, um die Eigennamen im Text zu erkennen.
Die entstandenen Ereignis-Cluster werden zum Abschluss noch eingeordnet. Dazu wird eine Art „Verlässlichkeit“ und „Popularität“ errechnet, die sich aus der Summe der Followers bzw. der Anzahl der Re-Tweets berechnet. Zusätzlich werden noch aktuelle Ereignis-Cluster, also Ereignis-Cluster, bei denen als letztes ein neuer Tweet hinzugefügt worden ist, bevorzugt, um somit die Aktualität mit zu gewichten.
Eine weitere Art der Nachbearbeitung bzw. Bewertung der Ergebnisse nimmt [51] vor. Dort werden die gefundenen Events erst mit Hilfe der Wikipedia verifiziert. Und zwar nutzt man hier die Seitenaufrufe der Wikipedia, welche man ebenfalls abfragen kann. Somit will man besser Real-World-Ereignisse erkennen. Das Beobachten von neu angelegten Seiten, um ein neues Ereignis zu verifizieren, hat sich dagegen, laut [51], als kein guter Indikator bewährt.

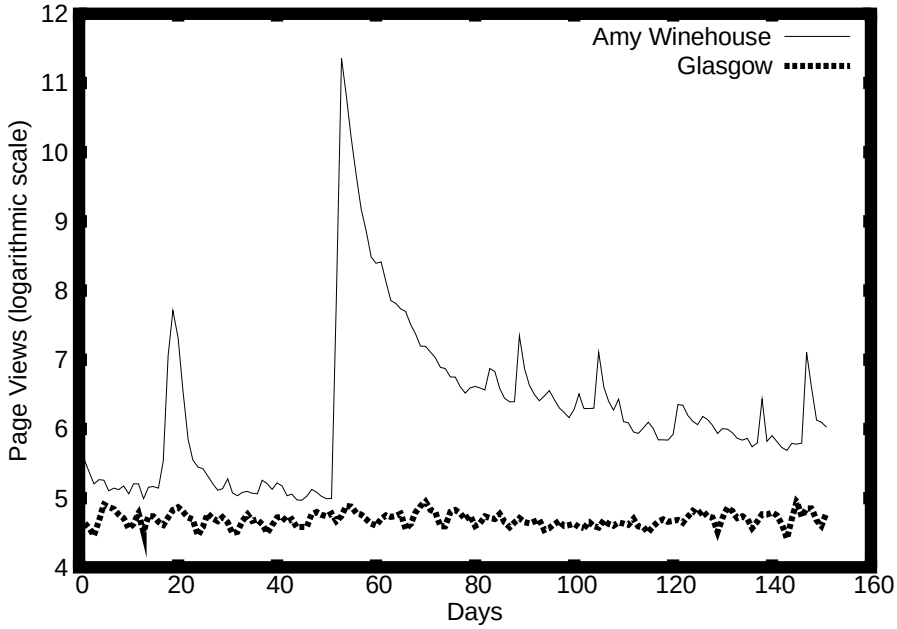

Abbildung 7: Stündliche durchschnittliche Anzahl von Besucher auf den Wikipediaseiten von Amy Winehouse und Glasgow in einer logarithmischen Tabelle zwischen den Tagen 1. 7.2011 und 31. 11.2011 [51]

In Abbildung 7 sind die stündlichen durchschnittlichen Besucherzahlen der englischen Wikipedia-Seiten von Amy Winehouse und der Stadt Glasgow im Zeitraum vom 01.07.2011 und 31.11.2011 in einer logarithmischen Tabelle eingetragen. Es sind in den Besucherzahlen der Amy Winehouse Seite, deutliche Sprünge zu erkennen, die durch Real-World-Ereignisse der Person selbst (Arrest und ihr Tod) verursacht wurden. Als Vergleich gegenüber gestellt sind die Seitenabrufe über die Stadt Glasgow. Zu dieser Zeit

fand in Glasgow kein ausgewöhnliches Ereignis statt. Doch dieses Vorgehen hat einen entscheidenden Nachteil, nämlich die Geschwindigkeit. Es wurde in [51] festgestellt, dass Wikipedia im Vergleich zu Twitter ein relativ langsames Medium ist. Die Seitenzugriffe steigen ungefähr erst 2h nach der Erkennung des Ereignisses in Twitter an. Im Durschnitt schauen die Nutzer also 2h nachdem ein Ereignis stattgefunden hat auf die betreffende Wikipedia-Seite, um sich zu informieren. Ein etwaiger Zeitvorsprung, durch die Detektion von neuen Ereignissen in Twitter, lässt sich so nicht ausnutzen.

2.4.3.1 Modifikationen der Cluster-basierten Ereignisdetektionsalgorithmus

Nicht nur in der Filterung und Bearbeitung der eintreffenden Daten oder der Weiterverarbeitung der erkannten Ereignisse gibt es Modifikationen, sondern auch am eigentlichen Erkennungsalgorithmus kann es diverse Erweiterungen geben. Diese Erweiterungen haben die Aufgabe, die Ergebnisse für den jeweiligen Anwendungsfall zu verbessern oder aber auch die Erkennung performanter ablaufen zu lassen.
In [52] und [53] wird mit Hilfe eines Ansatzes, namens KeyGraph versucht, Ereignisse zu detektieren. Diese gefundenen Ereignisse bzw. sogenannte Communities werden im Vektorraum durch ein künstliches Dokument (Key Document) repräsentiert. Dieses Key Document nimmt sozusagen die Rolle des Zentroiden der Ereignis-Cluster ein. Kommt ein neues Dokument hinzu, so wird die Ähnlichkeit (Cosinus-Abstand) zu diesem Key Document berechnet. Ist diese Ähnlichkeit über einer bestimmten Schwelle, so wird das neue Dokument dem Ereignis dazugerechnet.

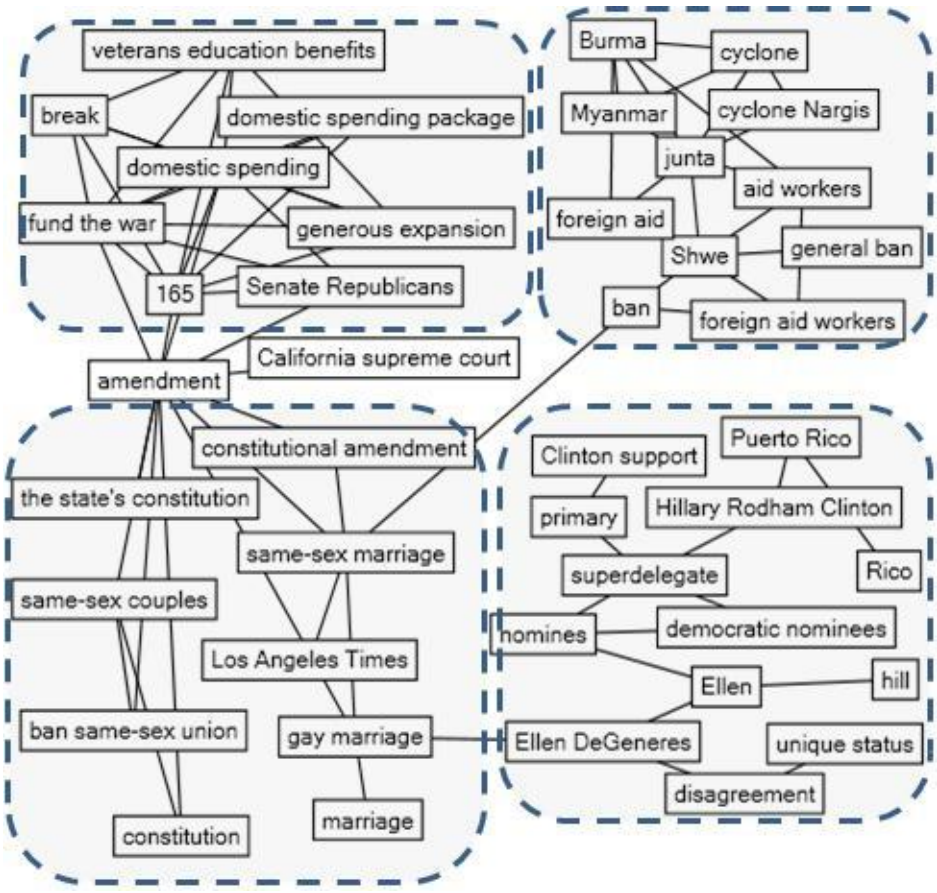

Abbildung 8: Beispiel KeyGraph mit eingezeichneten Communities und Termen [52]

In Abbildung 8 ist ein Beispiel für solch gefundene Communities (gestrichelte Bereiche) mit deren Termen und Verbindungen dargestellt. Im ersten Schritt werden aus den Tweets Terme extrahiert. Dies sind z.B. Eigennamen oder Nominalphrasen (ist ein

Phrase, deren Kern ein Substantiv ist z.B. „der alte Baum“ im Satz „Der alte Baum wurde vom Blitz getroffen“ [54]) die auf ihre Stammform reduziert sind und von Stoppwörtern befreit worden sind.
Danach werden die Termfrequenzen $TF_{i,j}$, die Dokumentfrequenz DF_i und die Inverse Dokumentfrequenz IDF_i errechnet. Terme mit einer DF_i, die kleiner als eine zuvor festgelegte Schwelle ist, werden wieder entfernt (d.h. Terme, die sehr selten in den Dokumenten/Tweets auftauchen, werden nicht berücksichtigt). Danach werden diese Terme miteinander verbunden, wenn sie zusammen in ein Dokument auftauchen. Dabei wird eine Schwelle angesetzt, so dass die Verbindung nur gesetzt wird, wenn die Wahrscheinlichkeit, dass der Term k_i zusammen mit dem Term k_j in einem Dokument auftritt größer als eine Schwelle ist. Ebenso umgekehrt, d.h. $p(k_i|k_j)$ und $p(k_j|k_i)$ müssen größer als eine definierte Schwelle sein.
Hat man diese Schritte durchgeführt, so hat man ein Netzwerk von Termen, die miteinander verbunden sind. Abgrenzende Communities bzw. Events lassen sich aber noch nicht richtig voneinander trennen, da es noch Verbindungen zwischen den Communities gibt. Diese Verbindungen werden im nächsten Schritt entfernt, indem der Betweenness- Zentralitätswert errechnet wird. Dazu wird zwischen zwei Termen der kürzeste Pfad berechnet. Ist eine Kante ein Teil dieses kürzesten Pfades, so wird dessen Betweenness-Zentralitätswert um eins erhöht. Kanten, die somit z.B. an der Grenze einer Community liegen und zwei Communities verbinden, haben somit eine hohe Betweenness-Zentralität (Intermediationszentralität). Diese Kanten werden entfernt und das Ganze wird erneut berechnet, solange bis es keine hohen Betweenness-Zentralitätswerte gibt. Liegt ein Term in zwei Communities gleichzeitig, so wird er in [52] verdoppelt, so dass er anschließend in beiden Communities liegt.
Nach diesen Vorgehen haben sich die Communities, also die erkannten Ereignisse, getrennt und werden, wie oben beschrieben, als künstliches Dokument in den Vektorraum integriert, damit die neu ankommenden Dokumente/Tweets mit diesen verglichen werden können.
Somit ist die eigentliche Ereigniserkennung (First Story Detection) durch den KeyGraph-Ansatz entschieden anders abgelaufen. Die restlichen Schritte sind danach wieder vergleichbar mit dem generellen Vorgehen der Ereignis-Cluster-basierten Ereignisdetektion.
Ein weiterer Grund für einen veränderten Ereigniserkennungsalgorithmus ist die Performance. Eine Performanceoptimierung der Erkennung wird in [55] durchgeführt. Um die Performance zu optimieren, wurden in den zuvor besprochenen Beispielen ebenfalls schon die Ereignis-Cluster im Vektorraum in Abhängigkeit mit der Zeit gewichtet. So fielen alte Ereignis-Cluster aus dem Vektorraum heraus, so dass einkommende Tweets nur noch mit aktuellen Ereignis-Clustern verglichen werden mussten. Wenn die Menge der einströmenden zu untersuchenden Tweets aber zu groß ist, so müssen weitere Optimierungen durchgeführt werden, um die eintreffenden Daten zu bearbeiten. In [55] wird dies mit Hilfe von sogenannten Buckets realisiert. Die Grundidee besteht darin, die Vergleiche im Vektorraum zu reduzieren. Ein neuer Tweet und dessen extrahierter Feature-Vektor soll so nicht mit allen Ereignis-Clustern oder allen Feature-Vektoren im Vektorraum verglichen werden müssen, um den nächsten Nachbarn zu ermitteln. Die Vergleiche sollen nur mit einer Teilmenge von Feature-Vektoren durchgeführt werden. Dazu werden die Feature-Vektoren in Buckets einsortiert. Dabei finden sich die Feature-Vektoren, die nah beieinander liegen, immer im gleichen Bucket wieder. Somit findet der eigentliche Vergleich der Feature-Vektoren nur mit den Vektoren statt, die im gleichen Bucket liegen, wie der neue Feature-Vektor des neu eingetroffenen Tweets. In [55] hat man dazu ein Hashing-Verfahren gewählt, um

diese Buckets zu realisieren. Der Abstand der Vektoren wird mit dem Cosinus-Abstand beschrieben. Die Wahrscheinlichkeit, dass zwei Vektoren kollidieren und somit im gleichen Bucket liegen ist proportional zum Cosinus-Abstand der beiden Vektoren im n-dimensionalen Vektorraum. Damit beträgt die Wahrscheinlichkeit, dass es eine Kollision bei diesem Hashverfahren gibt:

$$P_{coll} = 1 - \frac{\theta(x,y)}{\pi}$$

Formel 5: Wahrscheinlichkeit der Kollision von Vektoren im Vektorraum [55]

Wobei $\theta(x,y)$ der Winkel zwischen den zwei Vektoren ist. Somit ist es eben sehr wahrscheinlich, dass zwei Feature-Vektoren, die nah beieinander liegen, in denselben Bucket einsortiert werden. Die restlichen Vergleiche finden dann nur mit den Elementen in diesen Bucket statt und nicht mehr mit allen Vektoren im Vektorraum.
Bei dem Fall, dass durch diesen Algorithmus kein nächster Nachbar gefunden wird, überprüft man noch in [55] den Abstand zu den am häufigsten genutzten Vektoren im Vektorraum. Dies wird deshalb gemacht, da der Abstand zu dem neuen Vektor größer sein kann und somit durch den Bucket-Algorithmus nicht im selben Bucket gelandet ist.
Eine weitere Optimierung wird anhand der Bucket-Größe vorgenommen. Die Buckets können nur eine bestimmte Menge an Tweets aufnehmen. Ist das Bucket voll und soll ein neuer Tweet in den Bucket kommen, so wird das älteste Element aus dem Bucket entfernt. Somit wird die Menge der Vergleiche weiter eingeschränkt und konstant gehalten.

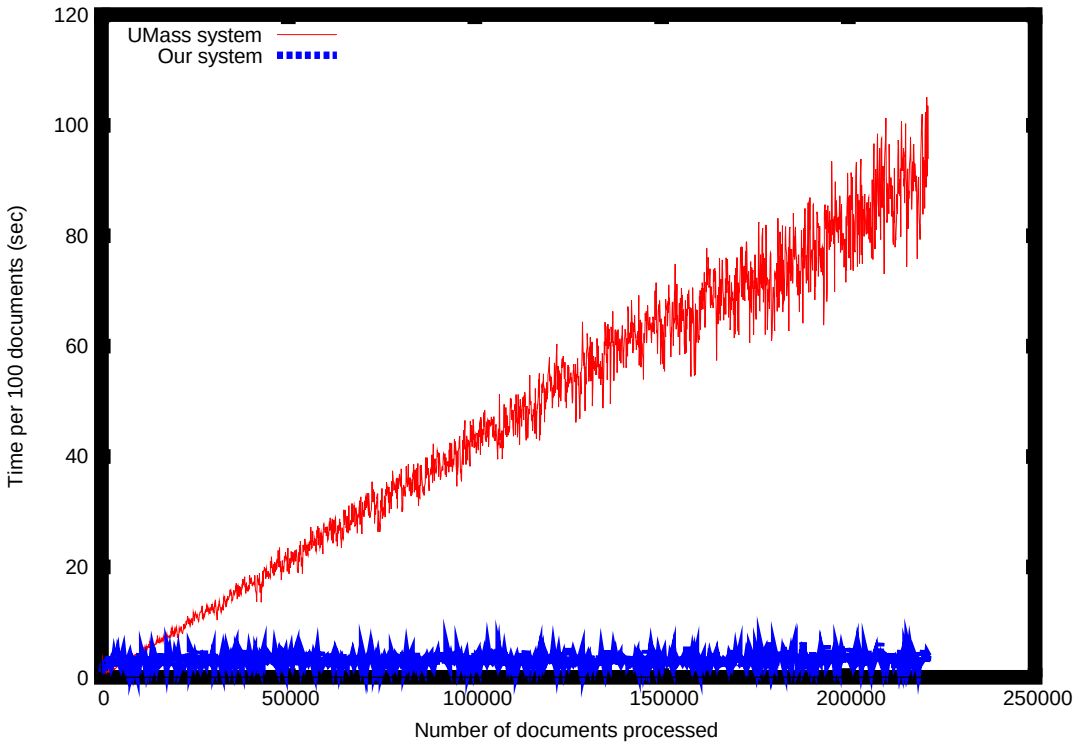

Abbildung 9: Ausführungszeit zwischen den performanceoptimierten System in [55] und einem herkömmlichen System, welches jedes Dokument mit allen Dokumenten im Vektorraum vergleicht

In Abbildung 9 erkennt man wie die Performanceoptimierungen gewirkt haben im Gegensatz zu einem herkömmlichen System [55]. Während die Rechenzeit bei einem herkömmlichen System mit der Menge des Inputs linear steigt, bleibt die Rechenzeit des

Algorithmus in [55] relativ konstant, da die Anzahl der Vergleiche im Vektorraum durch den Algorithmus gedeckelt ist.
Eine weitere Möglichkeit der Modifikation des Erkennungsalgorithmus stellt [56] vor. Dort geht es auch darum, die Menge der Ereignis-Cluster im Vektorraum zu begrenzen und alte Ereignis-Cluster zu entfernen, die nicht mehr benötigt werden. Es werden aber nicht starr Ereignis-Cluster die ein bestimmtes Alter haben entfernt, sondern die Herauslösung erfolgt dynamisch, je nachdem, ob dieses Ereignis noch aktiv ist oder nicht.
Im vorigen Beispiel wurden Buckets genutzt, um die Menge der Vergleiche zu begrenzen. Als Nachteil erweist sich aber hier, dass Tweets zu dem gleichen Ereignis in unterschiedliche Buckets landen und somit nicht erkannt werden können, dass sie ein und dasselbe Event beschreiben. Auch die Verwendung von starren Zeiten ist problematisch, um alte Ereignis-Cluster zu entfernen. Es gibt Ereignisse, die über einen längeren Zeitraum gehen können. D.h. es gibt Ereignis-Cluster, die zwar alt sind, aber wo immer noch Tweets zu diesem Ereignis neu hereinkommen (z.B. eine langwierige Gerichtsverhandlung). Oder aber der gegenteilige Fall, wenn ein Ereignis sehr kurz ist (z.B. ein Verkehrsunfall) und der Ereignis-Cluster noch im Vektorraum drin ist, wenn ein ähnliches Ereignis (z.B. ein neuer Verkehrsunfall) geschieht. Dann kann es sein, dass die zwei unterschiedlichen Ereignisse miteinander vermischt werden. Die Arbeit in [56] will dieses Problem lösen und führt dazu ein „self-adaptive life cycle“ ein. Das Ereignis wird hier mit der Geburt, dem Aufwachsen, dem Altern und dem Tod eines Lebewesen verglichen. Genutzt wird eine endogene Fitness. D.h. die Energie des Ereignisses steigt an, wenn das Ereignis populär ist und sinkt automatisch im Laufe der Zeit ab, wenn das Ereignis langsam an Relevanz verliert. Ist die Energie im Laufe der Zeit erloschen, d.h. das Ereignis ist tot, kann das Ereignis entfernt werden. Die Tweets selbst stellen in diesem Modell das „Futter“ für das Ereignis bzw. des Lebewesens dar und erhöhen die Energie des Ereignisses. Je besser der Tweet zum Ereignis passt, also je kleiner der Abstand zwischen den Ereigniszentroiden und des Feature-Vektors des Tweets ist, desto energiereicher ist er für das Ereignis. Der Energiegehalt des Ereignisses wird weiterhin noch auf einen Wertebereich zwischen 0 und 1 gemappt mit Hilfe einer Sigmoid-Funktion um den Energiegehalt eines neuen Tweets zu deckeln. Dies ist notwendig, damit ein Ereignis nicht zu viel Energie bekommt. Dies kann der Fall sein, wenn sehr viele Tweets, die sehr gut zum Ereignis passen, das System erreichen. Erst durch das Mapping auf den Bereich zwischen 0 und 1 wird der maximale Energiegehalt des Ereignisses selbst noch gedeckelt. Durch Tests in [56] konnte gezeigt werden, dass mit diesem Vorgehen sehr gut lang andauernde und kurzzeitige Ereignisse erkannt werden konnten.

2.4.4 LDA-basierte Ereignisdetektionsalgorithmen

Eine weitere Möglichkeit Ereignisse zu detektieren, besteht darin eine Latent Dirichlet Allocation (LDA) Analyse durchzuführen. Die LDA ist ein generatives Wahrscheinlichkeitsmodell, welches von David Blei, Andrew Ng und Michael I. Jordan im Jahr 2002 vorgestellt wurde [57]. LDA wird für Text- und Bildkorpora verwendet. Der Grundgedanke ist, dass jedes Dokument in einem Korpus ein oder mehrere Themen beschreibt. Dabei kann ein Term in solch einem Dokument ebenfalls zu einem oder mehreren Themen gehören. Ein Thema bzw. in unseren Fall Ereignis, wird durch eine Anzahl von Termen (Bag of Words) und deren jeweilige Wahrscheinlichkeit ihres Auftretens beschrieben [58]. Eine Besonderheit bei LDA ist, dass die Anzahl der Themen/Ereignisse vorher definiert wird und somit fest steht. Die LDA-Analyse liefert dann, für die zuvor festgelegte Anzahl von Ereignissen, die einzelnen Terme zu den Ereignissen. In Abbildung 10 ist solch eine Analyse beispielhaft dargestellt.

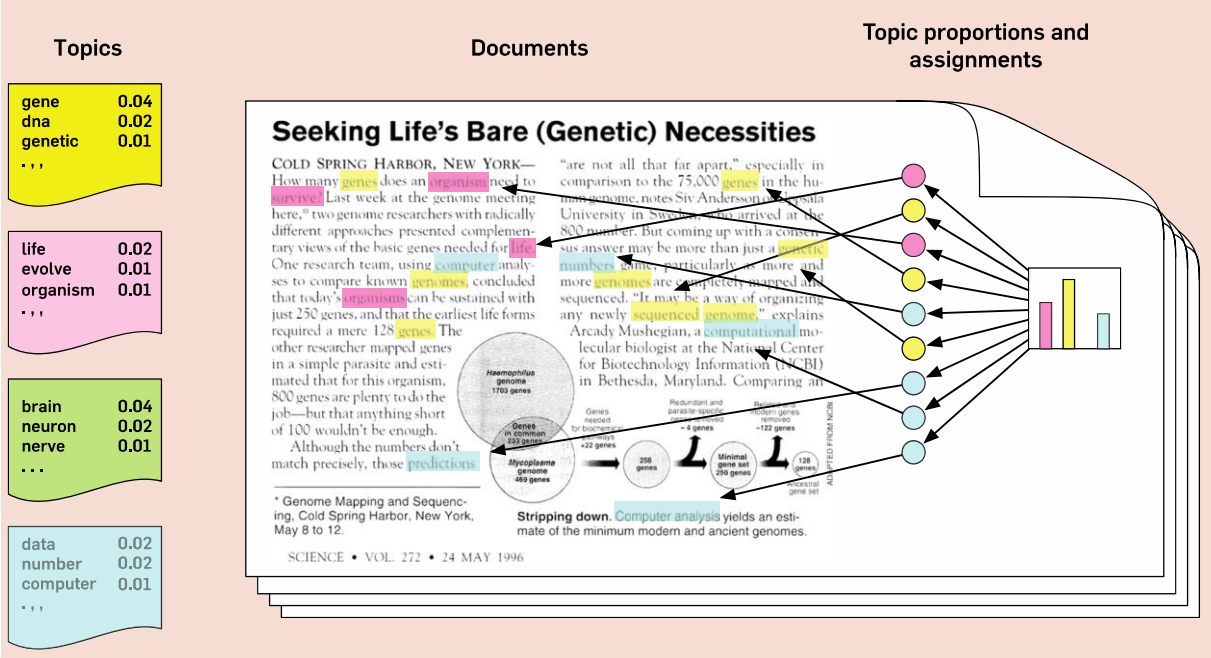

Abbildung 10: Beispiel wie aus einem Dokument, Terme extrahiert werden und einer zuvor bestimmten Anzahl von Ereignissen, zugeordnet wird [58]

Solch eine LDA-Analyse zur Ereignisdetektion wird in [38] durchgeführt. Dort wird ein semi-automatisches Analysesystem beschrieben, welches einen Analyst unterstützt, Ereignisse einer bestimmten Region und eines bestimmten Zeitabschnittes zu analysieren. Dabei gibt der Analyst auch an, wie viele Ereignisse detektiert werden sollen. Möchte er eine bestimmte Region oder einen Zeitabschnitt genauer analysieren, so kann er die Anzahl der gewünschten Ereignisse erhöhen und das System kann die einzelnen Terme in noch genauere Ereignisse zerlegen. Es kommt somit auf den Analysten an, was die Analyse am Ende genau als Ergebnis produziert. Mit Wahl der entsprechenden Parameter kann der Analyst die Analyse in die gewünschte Richtung lenken. Durch die große Bedeutung des Analysten, beschreibt [38] auch vorrangig die Interaktion mit dem System bzw. wie die Ergebnisse visualisiert werden bzw. wie daraus neue Anfragen an das System gestellt werden können. Der Analyseprozess ist in Abbildung 11 beispielhaft dargestellt.
Hat der Analyst den Zeitrahmen, die Region und die gewünschte Anzahl von Ereignissen angegeben, so wird auf den Tweets eine LDA-Analyse durchgeführt. Zuvor wird noch ein Stemming der Terme vorgenommen, welches eine Vorbearbeitung der Inputdaten darstellt, die man aus den vorherigen Ereignis-Cluster-basierten Ereignisdetektions-algorithmen schon kennt.
Nach der LDA-Analyse sind die Terme den jeweiligen Ereignissen zugeordnet. Entweder können die gefundenen Ereignisse dem Analyst angezeigt werden oder es kann optional eine weitere Analyse stattfinden, indem man die Außergewöhnlichkeit der Ereignisse berechnet mit Hilfe von Tweets, die von einer Woche vor dem zu analysierenden Zeitraum stammen. Dies erfolgt optional durch eine „Seasonal Trend Decomposition based on Loess“ (STL) wo versucht wird, wiederkehrende Ereignisse herauszufiltern.

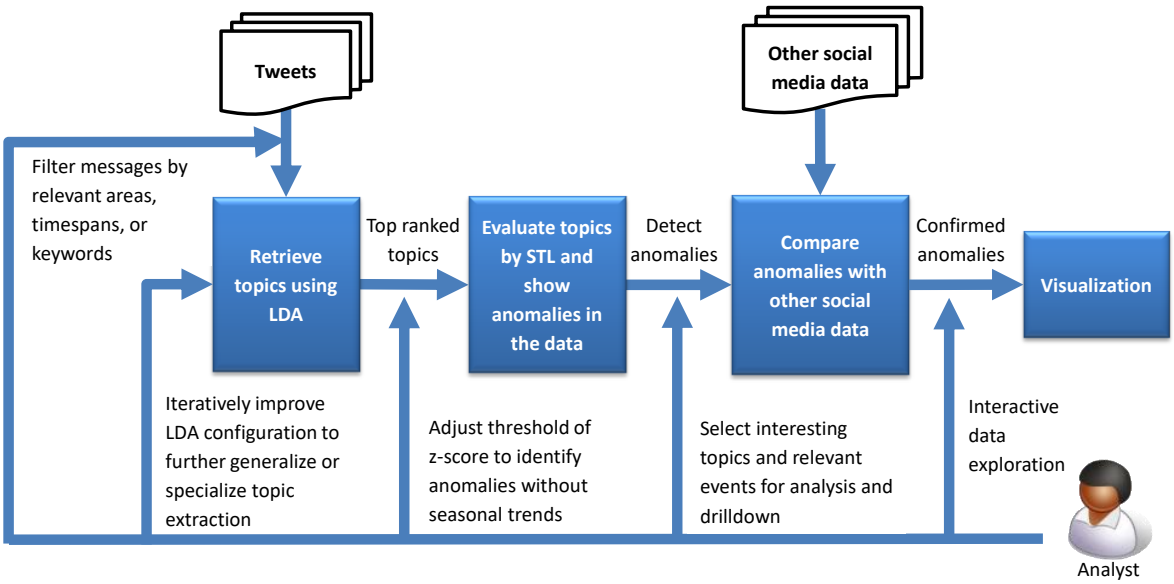

Abbildung 11: Beschriebener Analyseprozess in [38]

Es wird versucht nur unerwartete Ereignisse zu detektieren, die es zuvor noch nicht gab. Dazu werden die Ergebnisse noch auf Peaks untersucht, da dies darauf hindeuten kann, dass hier ein ungewöhnliches Ereignis im Gange ist. Man möchte also nur die Anomalien haben, die dann mit anderen Social Media Diensten validiert werden. Wenn es auch in den anderen Social Media Diensten eine Entsprechung gefunden wird, dann wird das Ereignis dem Analyst angezeigt. Hier wird ersichtlich, dass dieses Analysesystem nicht dazu gemacht ist, um Ereignisse in Echtzeit zu detektieren, sondern für eine spätere Analyse von Ereignissen bzw. genauere Analyse von Regionen. Bei der Analyse in [38] wurde erkannt, dass bei einer sehr genauen Analyse, d.h. eine LDA-Analyse mit einer hohen Anzahl von vorgegebenen Ereignissen, sehr viele kleine Ereignisse detektiert werden konnte, die man in den anderen Social Media Diensten nicht gefunden hatte. Auch bei abrupten Ereignissen wie z.B. eine Schießerei oder ein Erdbeben, konnte dieses Ereignis in Twitter detektiert werden, aber nicht z.B. bei YouTube oder Flickr. Die Autoren der Arbeit kommen zu dem Schluss, dass die Nutzer bei solchen Ereignissen viel schneller ein Tweet geschrieben haben als ein Video zu erstellen und dieses zu Posten. Bei zuvor bekannten Ereignissen wie z.B. der Occupy Wallstreet Bewegung findet man dagegen diese Ereignisse auch bei anderen Social Diensten. Aber auch hier ist zumindest eine Verzögerung zu erkennen. Twitter ist auch hier die schnellere und aktuellere Informationsquelle gewesen.
Auch in anderen Arbeiten wird die LDA-Analyse eingesetzt, um Ereignisse in den Twitter-Daten zu erkennen, wie z.B. in [59]. Doch kann die Analyse nicht als Echtzeitanalyse fungieren, da immer die Anzahl der Ereignisse vorher festgesetzt werden muss, die die Analyse danach findet. Somit fungiert dieses Verfahren eher dazu das man eine semi-automatisch Analyse danach durchführen kann, um die Ereignisse in den Twitter-Daten zu untersuchen.

2.4.5 Burst-basierte Ereignisdetektionsalgorithmen

Eine andere Herangehensweise ein Ereignis zu erkennen ist es, die Häufigkeit von Termen zu beobachten und das plötzliche gehäufte Auftreten bestimmter Terme zu erkennen, dass wiederum auf ein gerade stattfindendes Ereignis schließen lässt.
In [60] heißt es dazu: „*The underlying assumption is that some related words would show an increase in the usage when an event is happening.*“.

Die Analyse von Bursts kann einerseits als alleinige Ereigniserkennung fungieren oder aber als ein weiterer Baustein zur Ereigniserkennung. So kann z.B. eine Cluster-basierte Ereigniserkennung und eine Burst-basierte Analyse Hand in Hand gehen um die Ergebnisse zu verbessern.
Dass die Burstanalyse nicht erst mit Twitter-Daten unternommen wurde, zeigt [35]. Dort wird die Blogosssphäre zusammen mit den Tweets analysiert, die im Jahre 2007 erst neu aufkamen. Das Ziel ist dort zu erkennen, wann bestimmte Terme gehäuft in den Blogposts oder Tweets auftreten und welche Terme in diesem Zusammenhang gleichzeitig ebenfalls in der Nutzung mit ansteigen. Dieser Anstieg kann auf ein gerade laufendes Ereignis hindeuten. Man möchte so ermitteln, was gerade passiert und wann das Ereignis stattgefunden hat und wo es passiert. Bei der Frage wo, ist es schwierig zu ermitteln, da man hier auf die Metadaten der damaligen Blogs angewiesen war und es lokalisierte Tweets noch nicht gab. So wird angenommen, dass der Ort des Nutzers den er z.B. in einem Account angegeben hat, auch der Ort ist, wo das Ereignis geschieht. Bzw. wird versucht aus mehreren Quellen auf den Ort zu schließen. Durch Recherchemöglichkeit auf der Oberfläche der Analysesoftware, kann man nach weiteren Termen suchen und sich die betreffenden Blogposts anzeigen lassen. Auch die Häufung der entsprechenden Terme lässt sich in Diagrammen darstellen. So soll sichtbar werden, welche Terme im Zusammenhang stehen und die betreffenden Quellen präsentiert werden.
Um die Termfrequenz zu bewerten, ob sie erhöht ist gegenüber der Norm, benötigt man eine Vergleichsgrundlage. Dies geschieht mittels eines Korpus. In [61] wird dazu der Edinburgh Twitter Corpus genutzt, welcher 97 Millionen Tweets enthält und als ein Beispielkorpus für viele andere wissenschaftliche Arbeiten genutzt wird [62]. Der Korpus stellt sozusagen die Nulllinie dar und die neuen zu analysierenden Tweets werden damit verglichen, ob im Vergleich mit dem Korpus bestimmte Terme häufiger auftreten. Diese Art der Analyse wird auch Differenzanalyse genannt.
In [61] wird mit Hilfe des Korpus die normalisierte Termfrequenz errechnet. Mit Hilfe des Korpus wird die Termfrequenz in das Verhältnis zum Korpus gesetzt. Neben der normalisierten Termfrequenz, wird auch die eigentliche Frequenz der Terme, die TF-IDF und ein Entrophiewert errechnet. Mit zuvor definierten Schwellen bei den jeweiligen Werten wird überprüft, ob die errechneten Werte diese Schwellen überschreiten. Wenn ja, dann wird der Term markiert und kommt in die Ergebnismenge. Analysiert wurden Uni-Gramme und Bi-Gramme. Ziel war es, die Trending Topics, die Twitter selbst auf seiner Homepage präsentiert, so gut wie es geht nachzubauen d.h. also neue Ereignisse zu entdecken. Eine Filterung der Eingangsdaten, wie man sie von den vorherigen Methoden kennt, findet man hier auch. Es wird versucht Spam frühzeitig zu erkennen und herauszufiltern. Besonders möchte man nur englische Tweets analysieren und entfernt anderssprachige Tweets zuvor bzw. filtert Links und ähnliches heraus.
In einer weiteren Arbeit legt man den Fokus mehr auf die Echtzeiterkennung von Ereignissen. Einer der ersten Arbeiten, die die Burstanalyse auf den Twitter-Daten durchführte, um eine Echtzeitereignisdetektion durchzuführen, ist [63]. Es wird versucht Bursts zu erkennen, indem man die Terme pro Minute zählt. Die erkannten Burst-Terme werden mit Hilfe einer Kookkurrenz-Analyse (Analyse, ob bestimmte Terme aufgrund ihres gemeinsamen Auftretens in Beziehung stehen [64]) gruppiert. Die Terme die miteinander gehäuft auftreten gehören nun einer Gruppe an. Über die Burst-Terme und mit Hilfe der Kookkurrenz-Analyse werden danach noch weitere Terme gesucht, die mit den erkannten Burst-Termen häufig in Verbindung vorkommen. Ist dies geschehen, so wird versucht zu den jeweiligen erkannten Ereignissen (Burst-Terme und durch die Kookkurrenz-Analyse gefundenen Terme) mehr Informationen zu beschaffen, indem man z.B. Newslinks in den jeweiligen Tweets dem Nutzer mit präsentiert.

Durch die Tauglichkeit der Burstanalyse als eine Möglichkeit die Ereignisse in Echtzeit zu erkennen, gibt es auch Arbeiten, die diese Erkennung einsetzen, um damit Katastrophen zu detektieren, wie z.B. in [42]. Zusätzlich werden die Terme dort gewichtet, um z.B. wiederkehrende Ereignisse zu unterdrücken.
Es zeigt sich, dass es auch hier wieder alle Modifikationen der Analysemöglichkeiten gibt, wie man sie auch in den vorrangehenden Abschnitten bei der Cluster-oder LDA-basierten Analyse gesehen hat. In der Datenaufbereitung gibt es wieder die Variationen was und wie man filtert z.B. in [60], wo selten auftretende Terme zuvor schon herausgefiltert werden. Bei der Analyse selbst betrachtet man entweder Unigramme, also die Terme und ihr Auftreten einzeln, oder aber auch das Auftreten von bestimmten Termkombinationen (Bi-Gramme). Auch die Betrachtung der Ereignisse als Lebewesen und deren Energiegehalt als Maß für die Aktualität des Ereignisses, findet man in [65] wieder.
Auch bei der Aufbereitung der Ereignisse findet man die bekannten Modifikationen wie z.B. die Verbindung der erkannten Ereignisse mit Real-Life-Ereignissen [60] bzw. der Verknüpfung der Ereignisse mit anderen Quellen oder die Gruppierung von Termen [65] und die Suche nach weiteren Terme per Kookkurrenz-Analyse, um das Ereignis näher zu beschreiben.

2.4.6 Zusammenfassung der Modifikationen der Detektionsalgorithmen

In den vorhergegangenen Kapiteln wurde deutlich, dass die unterschiedlichen Ereignisdetektoren im Kern oft ähnlich sind. Dennoch unterscheiden sie sich im Detail doch, da die Ereignisdetektion viele Möglichkeiten zur Modifikation bietet (Abbildung 12).
Das fängt bei der Filterung und Aufbereitung der Eingangsdaten an und endet bei der Nachbearbeitung der Ergebnisse. Selbst der eigentliche Kern, die Ereignisdetektion, kann in vielen Varianten durchgeführt werden. Neben der groben Unterteilung der Detektion in Cluster-, LDA- und Burst-basierte Verfahren, gibt es innerhalb der Varianten weitere Modifikationen. Durch die hohe Variabilität der Methoden ist eine saubere Trennung der Verfahren auch kaum möglich und so kann es sein, dass man auch verschiedene Verfahren zusammen in einem solchen System wiederfindet. Das Ziel der Modifikationen des Ereignisdetektionsschritts ist zum einem die Verbesserung der Detektion, um die gewünschten Ereignisse überhaupt zu detektieren, zum anderen ist es die Performancegetriebene Optimierung. Gerade bei einer gewünschten Echtzeiterkennung von Ereignissen, müssen mitunter große Datenmengen verarbeitet werden. Um dies zu erreichen, sollte der Erkennungsschritt so performant wie möglich und so ressourcenschonend ablaufen wie möglich. So wird z.B. in den Cluster-basierten Verfahren großen Wert darauf gelegt, den Vektorraum so klein wie möglich zu halten, um die nötigen Vergleiche der Vektoren zu minimieren. Besonders im ersten Teil der Analyse, in der die Eingangsdaten gefiltert und aufbereitet werden, gibt es ein großes Spektrum an Modifikationen. Das Ziel ist hier, die Eingangsdaten so aufzubereiten, dass die Ereigniserkennung im nächsten Schritt optimal durchgeführt werden kann. Besonders legt man Wert darauf, nur solche Tweets dem System zu präsentieren, die auch potentiell ein Ereignis beschreiben.
Der Rest wird oft als Rauschen (Noise) oder noch drastischer als Müll bezeichnet und wird so gut wie es geht aussortiert. Einen guten Überblick, über die zur Verfügung stehenden Methoden dieser Filterung oder aber auch Normalisierung der Eingangsdaten (z.B. per Wortstammreduktion (Stemming), Entfernung von Schreibfehlern, Entfernung von nicht alphanumerischen Zeichen, Entfernung von zu vielen Satzzeichen „!!!“ usw.) liefert auch [66].

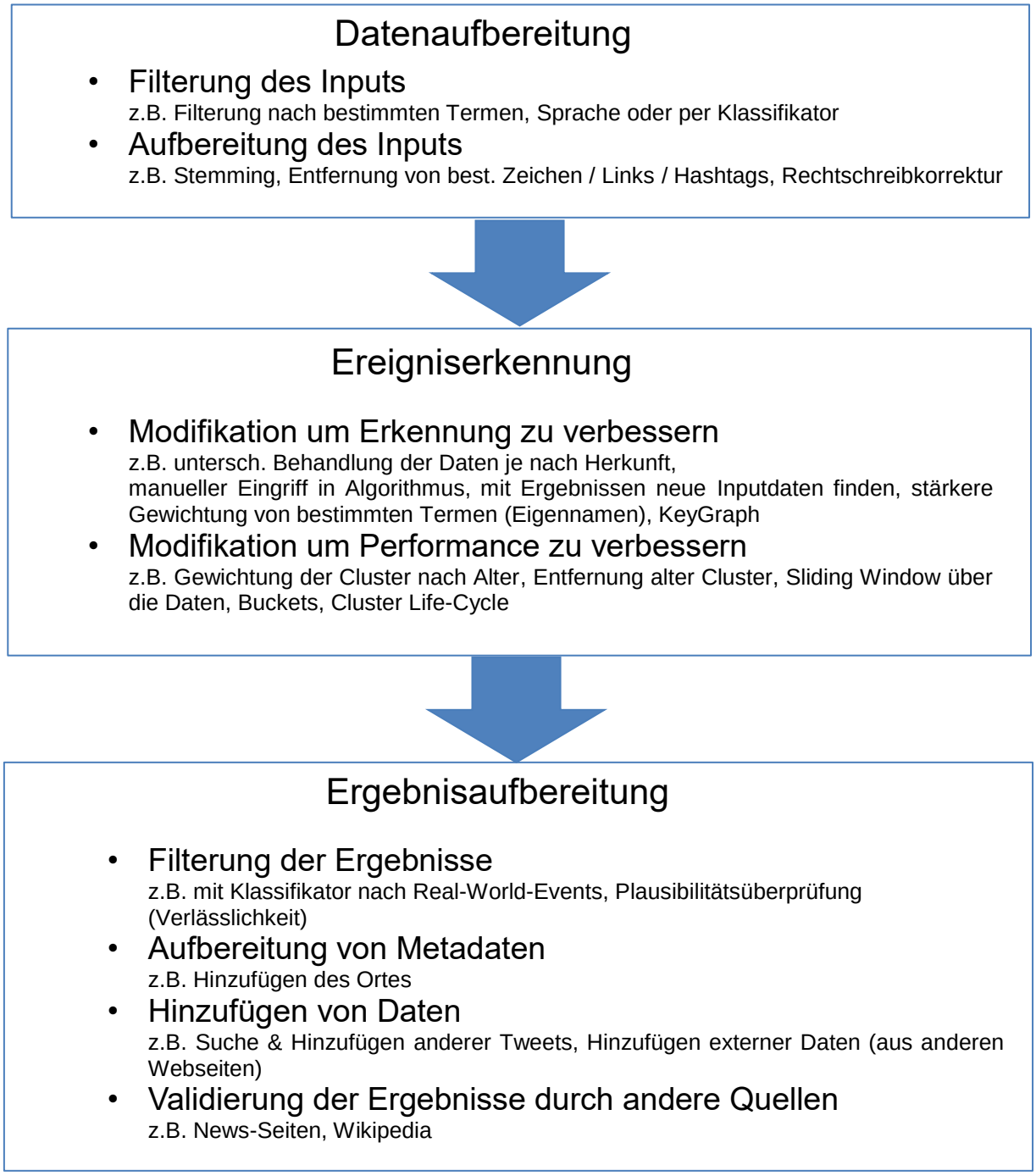

Abbildung 12: Auswahl von Modifikationsmöglichkeiten in den einzelnen Verarbeitungsschritten der Ereignisdetektion

Eine weitere drastische Filterung der Eingangsdaten ist es, wenn z.B. nur nach einer bestimmten Liste von Termen Ausschau gehalten wird. Hier werden nur die Tweets zur Analyse weiter gegeben bzw. nur abgerufen, wenn sie einen bestimmten Term beinhalten. Um die Tweets noch weiter zwischen nützlichen Tweets und Noise zu unterscheiden, können in diesen Schritt auch Klassifikatoren zum Einsatz kommen, die die Tweets bewerten und weiter aussortieren.
Auch die Ergebnisse der Detektion können auf unterschiedliche Arten weiter verarbeitet werden. Zum einen können sie noch mit weiteren Metadaten angereichert werden, wie z.B. der vermeintliche Ort des Ereignisses. Zum anderen werden die Ergebnisse auf Plausibilität überprüft, indem ein Gegencheck mit anderen Nachrichtenquellen oder Abrufstatistiken der entsprechenden Wikipedia-Seiten erfolgt. Auch die Filterung der

Ergebnisse mit Hilfe eines Klassifikators, um zu entscheiden, ob sich das detektierte Ereignis um ein „Real-World-Ereignis“ handelt, kann stattfinden. Um die Analyse zu verbessern, wird mit Hilfe der Ergebnisse nach weiteren Tweets gesucht, die das Ereignis näher beschreiben. Je nachdem wie die Analyse dem Nutzer präsentiert werden soll, werden Beispiel-Tweets gesucht oder andere Texte zusammengestellt, die das Ereignis beschreiben.

2.4.7 Ereignisdetektion mit Ortsbetrachtung

Bei den bisher betrachteten Systemen zur Ereignisdetektion, spielte der Ort kaum eine Rolle. Jedenfalls floss er nicht in die zentrale Analyse der Detektion ein, sondern wurde, wenn überhaupt, nur am Ende dazu genutzt, den gefundenen Ereignissen noch zusätzlich einen Ort zu benennen. Es wird dazu z.B. überprüft, ob die Tweets zu dem gefundenen Ereignis, zufällig alle aus einer Region stammen oder aber, wenn an den Tweets keine Ortsangaben enthalten sind, ob durch den Inhalt der Tweets auf einen Ort geschlossen werden kann. Dies ist z.B. der Fall in [44], wo nur CDE-Tweets (Crime and Disaster related Events) untersucht werden, indem nach bestimmten Termen Ausschau gehalten wird. Mit einem Klassifikator wird hier versucht nur CDE-Tweets aus dem Strom herauszufiltern. Hat man die Ereignisse erkannt, so wird versucht für dieses Ereignis einen Ort zu bestimmen, indem überprüft wird, ob der Tweet eine Ortsangabe in den Metadaten enthält, der Ort durch den Inhalt des Tweets hergeleitet werden kann oder der hinterlegte Ort des Autors des Tweets hier weiterhelfen kann. Der Ort wird dann final dazu genutzt, das entdeckte Ereignis, also hier das CDE, auf einer Karte zu markieren und dem Nutzer eine weitere Recherchemöglichkeit zu geben, indem darüber alle CDE Ereignisse für einen bestimmten Ort abgefragt werden können, um die räumliche und zeitliche Ausdehnung des Ereignisses darzustellen.
Doch mit Hilfe des Ortes lassen sich auch Ereignisse detektieren. Auch hier gibt es verschiedene Herangehensweisen. In [67] untersucht man Tweets, die in zeitlicher und räumlicher Nähe entstanden sind. Über diese Tweets wird eine Kookkurrenz-Analyse durchgeführt, um Key-Terme zu finden. Wenn bei den unterschiedlichen Nutzern dieselben Key-Terme gefunden werden, so deutet das laut [67] auf ein neues Ereignis hin (Aufbau des Systems siehe Abbildung 13). Da für die Arbeiten die Anzahl der Tweets mit Geoinformationen zu gering war, versuchte man auch Tweets einem Ort zuzuordnen, die keine Geoinformationen enthielten. Dazu wurden Tweets genutzt von Diensten, die melden, dass man sich an einen Ort eingecheckt hat (z.B. Foursquare [68]). Diese Tweets enthalten Geoinformationen und auch den Ortsnamen. Wenn einige Nutzer den Ort auch so benennen, dann kommt der Ortsname in eine Datenbank. Wenn nun ein Tweet auftaucht mit dem Inhalt „Es ist hier schön in X“ und X ist ein bekannter Ortsname, so kann man davon ausgehen das der Nutzer sich gerade dort befindet, sofern es kein Re-Tweet war.

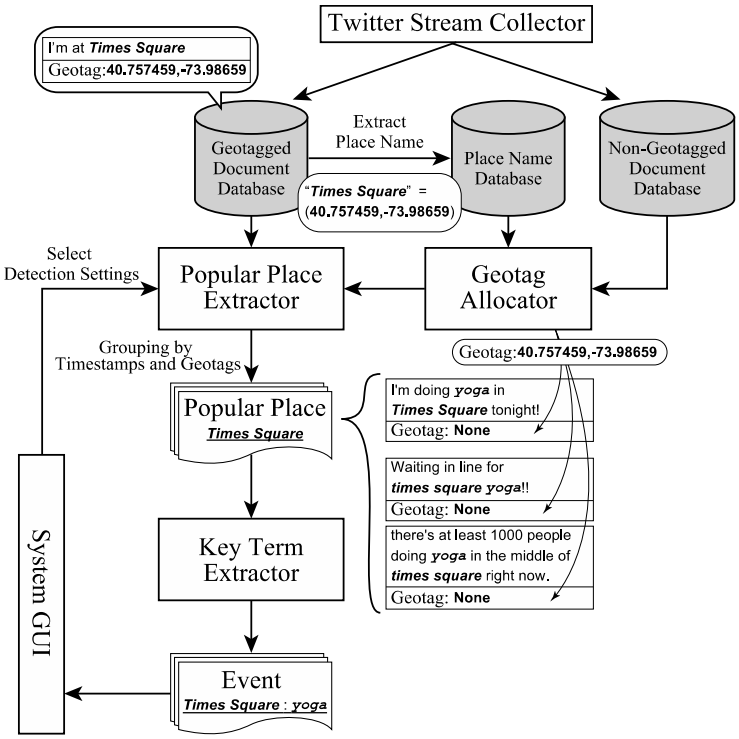

Abbildung 13: Aufbau des Systems in [67]

Auch die Orte an sich können genau beobachtet werden und somit ein erhöhtes Tweet-Aufkommen in einen Bereich entdeckt werden. So wird in [69] und [39] analysiert, wie viel Nutzer in einen bestimmten Gebiet tweeten und wie viel Nutzer zu bestimmten Gebieten ab- oder zuwandern (Abbildung 14). Das Ziel ist so, große Ereignisse wie Festivals zu erkennen, zu denen eine hohe Anzahl von Besucher hingehen und natürlich auch dort tweeten. Diese Untersuchungen wurden in Japan gemacht, wo es sehr viele Twitter-Nutzer gibt und somit potentiell sehr viele Tweets mit Geoinformationen. Detailliert beobachtet werden die Anzahl der Tweets, die Anzahl der Nutzer und die Bewegung der Nutzer zwischen den einzelnen Regionen. Wenn mindestens zwei dieser Werte signifikant höher als erwartet sind, geht man von einen Ereignis aus. Um den täglichen Ablauf zu berücksichtigen (Fahrten zu und von der Arbeit z.B.) wird der Tag gesplittet in jeweils 6h und diese dann getrennt untersucht. Da zu diesem Zeitpunkt der Arbeit die Twitter-Streaming-API noch nicht benutzbar war, hat man die Daten über die „normale“ Twitter API abgerufen. Hier besteht aber eine Einschränkung, dass man maximal 1500 Ergebnisse als Antwort bekommt. Um doch alle Tweets zu bekommen und somit eine Aussage über eine Zu- oder Abnahme der Tweets in einer bestimmten Region zu machen, teilte man das Gebiet in Voronoi-Regionen, so dass in jeder neuen Region die Tweetanzahl kleiner als 1500 ist und man alle Tweets aus dieser Region per Twitter-API abrufen kann.

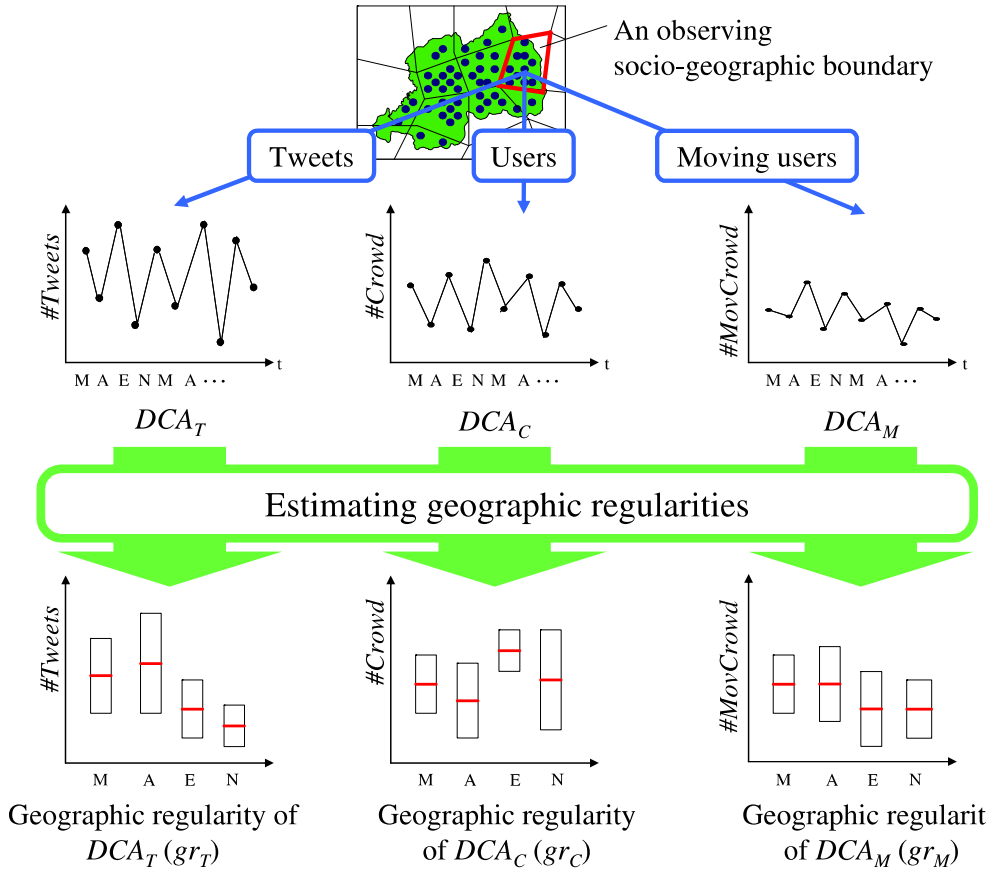

Abbildung 14: Beobachtung der Parameter in jeder Region und Überprüfung auf signifikante Veränderungen in den Regionen [39]

Eine Voronoi-Region ist dabei eine Teilfläche, die bei einer Zerlegung des Raumes durch ein Voronoi-Diagramm (oder auch Dirichlet-Zerlegung genannt) entsteht. Jede Voronoi-Region besitzt ein Zentrum. Alle Punkte in so einer Region haben die Eigenschaft, dass sie zu dessen Zentrum am nahesten liegen [70].

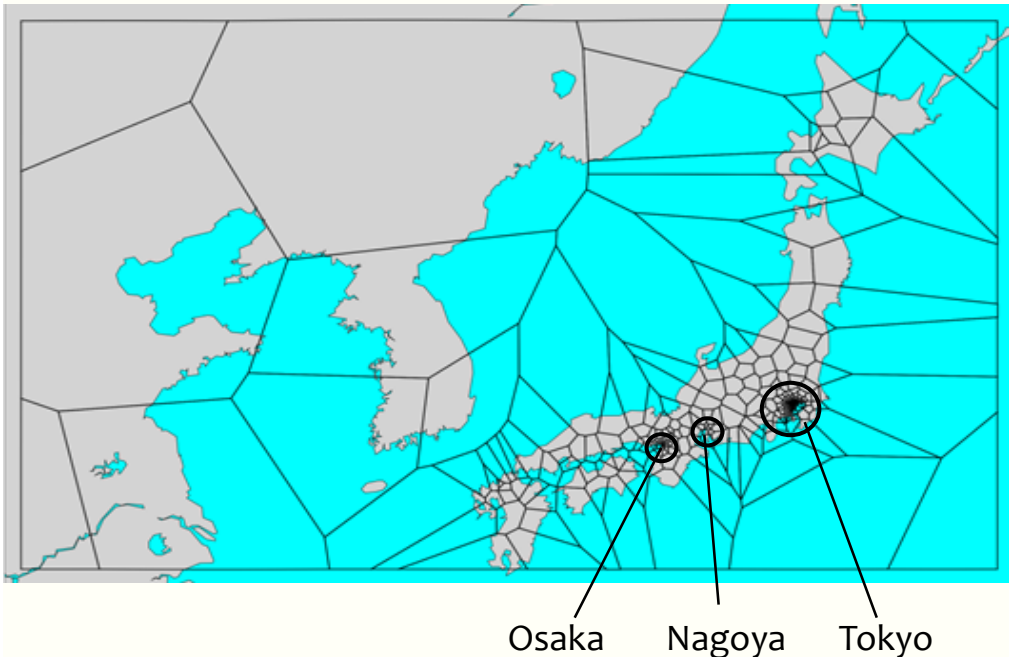

Abbildung 15: Zerteilung der Regionen in Voronoi-Regionen, um alle Tweets in diesen Gebieten über einen Twitter-API-Aufruf abzurufen [69]

In Abbildung 15 ist solch eine Zerlegung dargestellt. Einen ähnlichen Ansatz hat auch [40]. Hier nutzt man aber zum Abruf der Tweets die Streaming API und man ist so nicht mehr auf die Aufteilung des Gebietes in Voronoi-Regionen angewiesen. Als Forschungsgebiet hat man sich hier Irland ausgesucht. Es wird davon ausgegangen, dass Ereignisse abnormale Werte in der Termkonzentration sowie in den räumlichen und zeitlichen Verteilungen verursachen. Ein gehäuftes Auftreten in einen bestimmten Gebiet von bestimmten Termen oder eine vermehrte Bewegung bzw. einer höheren Anzahl von Tweets von unterschiedlichen Nutzern in diesem Gebiet, könnten auf ein Ereignis wie z.B. einen Festival hindeuten. Wo mehr Menschen sind, sind potentiell auch mehr Twitter-Nutzer, die auch mehr schreiben (auf das Gebiet hochgerechnet). In [40] konnte auch beobachtet werden, dass Menschen die gerade reisen, auch mehr tweeten. Weiterhin ließ sich auch beobachten, dass die Personen, mit denen direkt über Twitter kommuniziert wird, sich meist in der Nähe befinden, d.h. einmal in der Nähe des Arbeitsplatzes (Kollegen) oder aber die Freunde am Wohnort. Für die Ereignisdetektion wurde in [40] die LDA-Analyse eingesetzt. Die Annahme in dieser Arbeit war, dass große Ereignisse (z.B. Festivals zu sehen als Beispiel in Abbildung 16) automatisch ein detektierbares Ereignis in den Twitter-Daten erzeugt, die durch ihre Analyse sichtbar wird. Dies kommt daher, weil die Nutzer bei diesem Ereignis über dieselben Dinge schreiben und somit über dasselbe Ereignis. Durch die LDA-Analyse wird diese spezifische Termverteilung als ein Ereignis detektiert. Bei der Anwendung ihrer Methoden auf die Tweets von Irland, konnten sie so ein großes Musikfestival detektieren und dessen räumliche Ausbreitung.

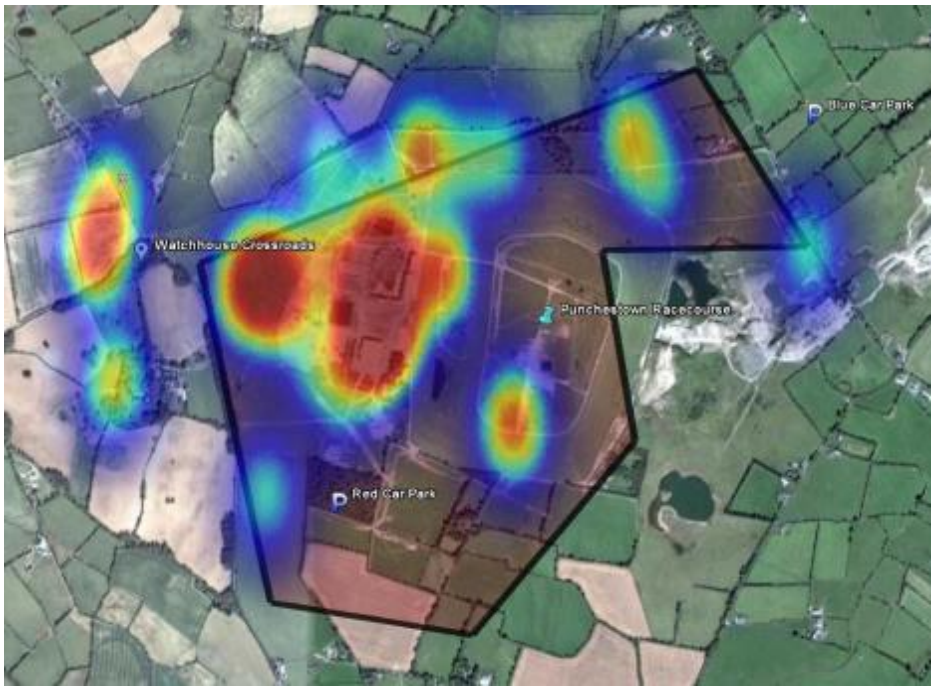

Abbildung 16: Übersicht über das Oxegen Festival Gebiet in Irland mit den erkannten Hotspots. Das rötliche Gebiet waren die offiziellen Grenzen des Festivals. [40]

Im Gegensatz zu einem Ereignis welches nicht an einem speziellen Ort gebunden ist z.B. die Premiere eines neuen Kinofilmes, erzeugte dieses Ereignis keinen signifikanten Hotspot.
Auch bei weiteren Arbeiten zur Ereigniserkennung mit Einbeziehung des Ortes, wie z.B. [46], findet man das bekannte Vorgehen. Hierzu gehören z.B. die Aufbereitung der Daten (z.B. Filterung der Eingangs-Tweets nach vordefinierten Termen, Entfernung von nicht

englischen Tweets, Filterung von Sonderzeichen und das Durchführen einer Wortstammreduktion (Stemming)) und die Anwendung der Bursterkennung. Kommen die Tweets plötzlich gehäuft in einer Region vor, so ist von einen neuen Ereignis auszugehen. Man konzentriert sich in [46] auf die Detektion von Naturkatastrophen wie z.B. Erdbeben und Tornados und auf von Menschen gemachte Ereignisse/Katastrophen wie z.B. Aufstände. Das Ziel ist, die Ereignisse so schnell wie möglich zu detektieren und so schneller als andere Medien zu sein. Man nutzt dazu die Streaming-API von Twitter und filtert alle nicht englischen Tweets heraus, filtert Tweets aus einer bestimmten Region und schränkt die Tweets zusätzlich auf bestimmte Terme ein, die in den Tweets beinhaltet sein müssen. Es konnte gezeigt werden, dass das System die Ereignisse ca. 2-3h schneller detektieren kann, als sie über die Internet Newsseiten publiziert worden wären.
Ein ähnliches Verfahren, welches aber nicht auf die Echtzeitanalyse setzt, ist [71]. Auch hier werden die Tweets über die Streaming API eingesammelt und die Tweets und die Nutzer gezählt, um plötzliche Anstiege in der Nutzung zu erkennen. Die Erkennung dieses Anstieges der Tweets bzw. Nutzer, wird mittels eines neuronalen Netzes durchgeführt. Die Besonderheit hier ist, dass Bins für die Orte erstellt werden, also ähnlich den schon vorgestellten Buckets, um die Laufzeit der Analyse zu optimieren. So bestimmt man den Ort per Koordinate und weißt sie einen Ort bzw. Bin zu. Weiterhin werden Zeitfenster betrachtet z.B. 1min, 10min, 1h und 6h. In diesen Zeitfenstern analysiert man die Anzahl der Tweets und Nutzer. Das System ist nicht für die Echtzeitanalyse gemacht, da durch die Architektur des Systems bedingt, die Analyse immer in 30min Schritten stattfindet. Erst dann werden die neuesten Tweets weitergeleitet bzw. es werden die entsprechenden Zeitreihen gebildet. Ein Anstieg wird erst als Ereignis erkannt, wenn der Anstieg in mind. 2 Zeitfenstern erkennbar ist. Je größer die Zeitfenster sind (1h oder 6h), umso besser sind die Ergebnisse. Bei größeren Zeitfenstern wird zwar nicht auf kleine Ereignisse reagiert, dafür wird aber auch die Erkennung. Aus den Tweets, die als Ausreiser erkannt werden, werden die Terme extrahiert, die das Ereignis beschreiben könnten. Zum Schluss werden die erkannten Ereignisse auf Newsseiten gegengecheckt, um zu erfahren was es für Ereignisse sind.

Zu guter Letzt sollen hier noch zwei weitere Arbeiten vorgestellt werden, um zu zeigen, welche Anwendungsgebiete mit Hilfe der Ereigniserkennung und den mit Geokoordinaten angereicherten Twitter-Daten noch möglich sind. Die Gemeinsamkeit der beiden Arbeiten ist, dass das Ereignis hier schon bekannt ist und man dieses bekannte Ereignis mit Hilfe der Twitter-Daten versucht zu analysieren. In [43] ist das Ereignis ein Waldbrand in Südfrankreich. In Abbildung 17 ist dessen Chronologie dargestellt. Man möchte die Geschehnisse, anhand der Daten, noch einmal Revue passieren lassen und den zeitlichen Ablauf und die Orte der Tweets darstellen. Anhand vorgegebener Terme werden die Tweets aus dem ausgewählten Zeitraum des Waldbrandes abgerufen. Da zu dieser Zeit die Nutzung der Geodatenfelder in den Tweets noch nicht so weit verbreitet war, wurden nur relativ wenige Tweets mit angehängten Koordinaten gefunden.

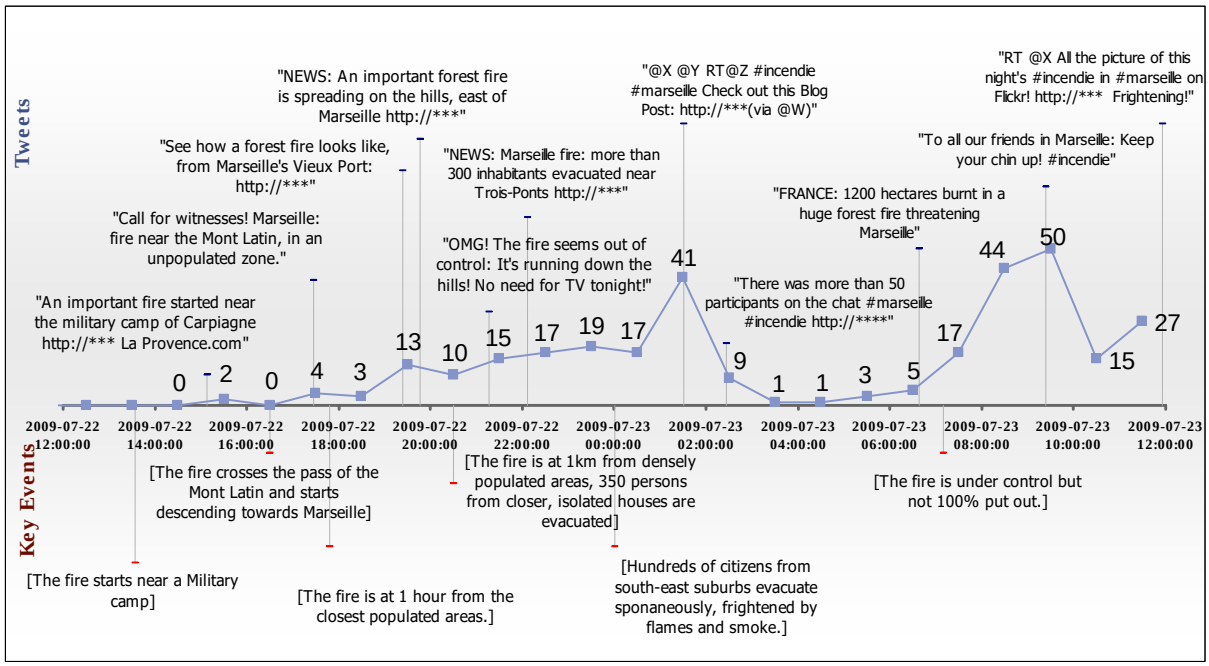

Abbildung 17: Chronologie des betrachteten Waldbrandes in Südfrankreich. Anzahl der ausgewählten Tweets und Inhaltsbeispiele. [43]

Da der Brand in einer abgelegenen Region begann, konnte gezeigt werden, dass hier erst regionale Medien von dem Ereignis erfahren haben. Erst eine Stunde später wurde über das Ereignis in Twitter berichtet, gefolgt von Augenzeugen-Tweets, die über das Ereignis direkt berichteten. Auch wurde sichtbar, dass die Meldung in dem Verlauf des Ereignisses, oft geretweetet wurde von Media-Aggregatoren (Webseiten die Nachrichten aus verschiedenen Quellen sammeln und präsentieren). Auch viele Tweets von den Medien selbst und mit enthaltenen Links zu den Medienseiten wurden registriert.
Die andere Arbeit, die in gewisser Weise auch die Ereignisdetektion nutzt, ist [41]. Hier wird die Schnelligkeit des Mediums Twitter genutzt, um ein ganz bestimmten Ereignistyp, nämlich Erdbeben, so schnell wie möglich zu detektieren.

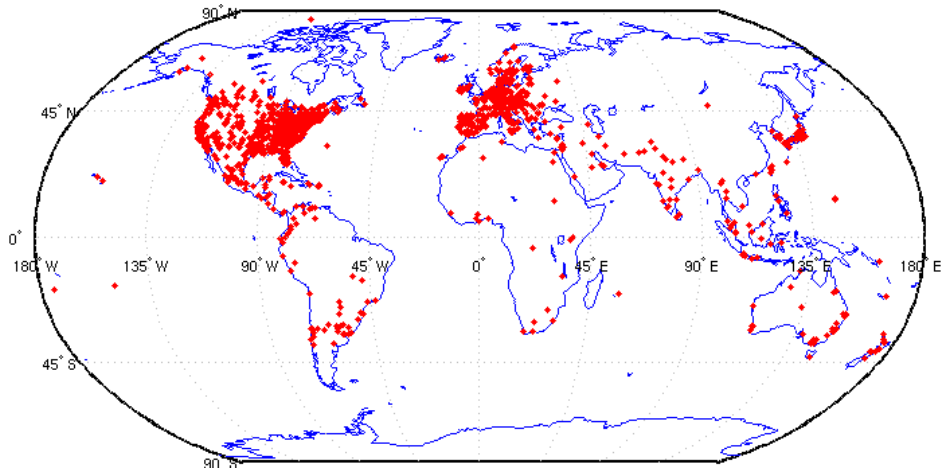

Abbildung 18: Twitternutzerkarte [41]

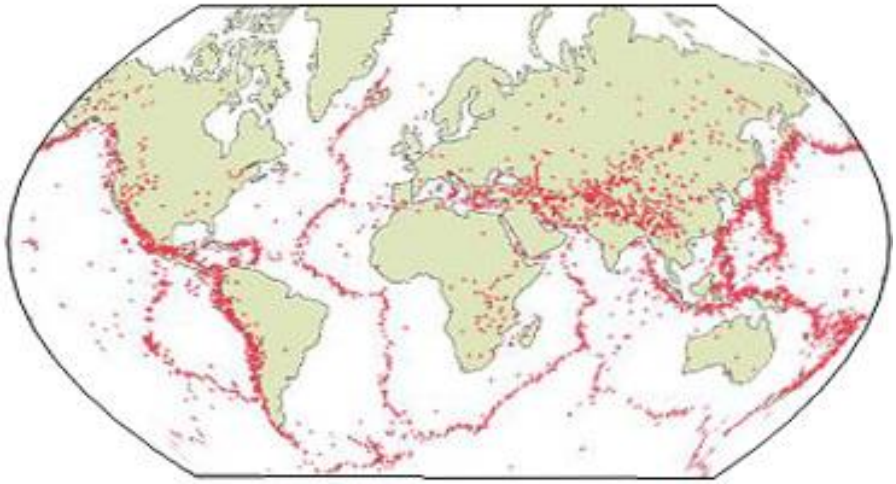

Abbildung 19: Erdbebenkarte [41]

Als Region für diese Analyse hat man Japan ausgewählt. Zum einen gibt es dort sehr viele Twitter-Nutzer und daher auch viele Tweets mit Geoinformationen und zum anderem ist es eine Region auf der Erde wo sehr viele Erdbeben auftreten (Abbildung 18 und Abbildung 19). Die Twitter-Nutzer in der gewählten Region werden als stark verrauschte Sensoren angesehen. D.h. wenn ein Erdbeben im Gange ist, kann es sein, dass der Twitter-Nutzer es twittert oder auch nicht bzw. kann es sein, dass der Nutzer über das Thema Erdbeben tweetet auch wenn gerade keins ist. Die Tweets werden nach bestimmten Termen gefiltert z.B. „shake“ und „earthquake“. In einen weiteren Schritt werden die Tweets, welche diese gesuchten Terme enthalten, noch klassifiziert, um zu beurteilen, ob damit gerade ein Erdbeben gemeint ist oder der Nutzer über andere Themen schreibt z.B. „Someone is shaking hands with my boss“ [41]. Auch der Anstieg der Tweets mit den betreffenden Termen ist wieder signifikant, wenn wirklich gerade ein Erdbeben stattfindet. Hier wird ein exponentieller Anstieg beobachtet. Für den Ort des Tweets wurden die Geodaten aus den Tweets ausgelesen oder zur Not der hinterlegte Ort des Nutzers genommen. In diesem Fall wurden diese Daten nicht für eine spätere Berechnung genutzt. Ziel war es, bei der Arbeit auch den Ort des Erdbebens zu detektieren. Mit Hilfe der Menge an Tweets und eines Partikelfiltersystems (Sequenzielle Monte-Carlo-Methode) ist es möglich, allein über die Sendepositionen der Tweets, den Ort des Erdbebens zu errechnen. Auch die Detektion und die Lokalisierung von Taifunen wurden durchgeführt und man konnte die ungefähre Wegstrecke des Taifuns über das System registrieren. Für die Positionsbestimmung stellte es ein Problem dar, wenn der Taifun noch über dem offenen Meer war. Da keine Tweets von Meerseite existierten, konnte die Position des Taifuns hier nur ungenauer ermittelt werden.
Eine der größten Erfolge der Arbeit ist es aber, dass die Erkennung des Erdbebens schneller von statten geht als die offizielle Erdbebenwarnung des Landes. Somit wurde ein Service implementiert, welcher die Nutzer in dem Umkreis des detektierten Erdbebens benachrichtigt, so dass die Menschen sich auf die Erschütterungen vorbereiten können. Die Abbildung 20 und Abbildung 21 zeigt auf einer Karte die Ergebnisse der Analyse.

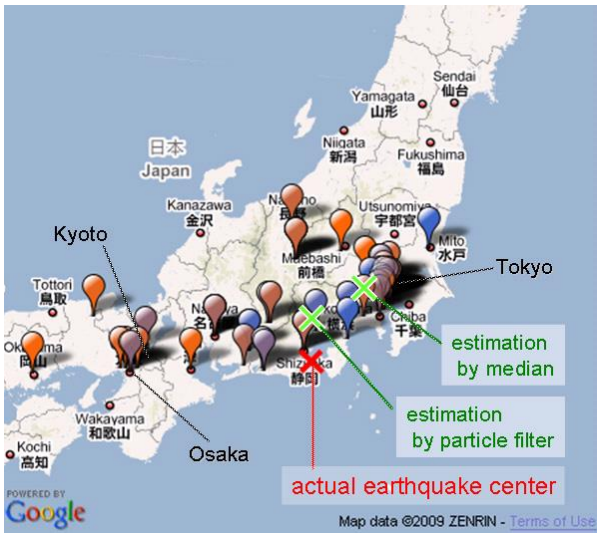

Abbildung 20: Lokalisierung von Erdbeben anhand von Tweets über Erdbeben und deren Position [41]

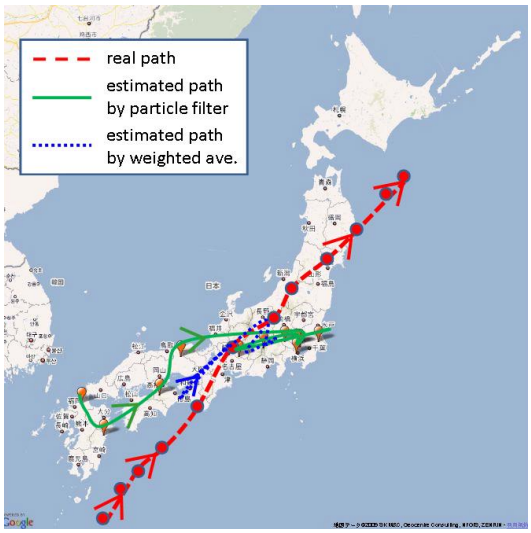

Abbildung 21: Berechnung des Weges eines Typhons anhand von Tweet-Daten [41]

2.5 Verbesserungsmöglichkeiten für die Ereignisdetektion

Noch nie war es so einfach an aktuelle (sogar in Echtzeit) und öffentliche Nachrichten von einer großen Anzahl von Nutzer zu kommen. Durch die sozialen Netzwerke mit deren APIs und den vielen Nutzern, die andere Menschen an ihrem Leben teilhaben lassen, indem sie Nachrichten posten über das was sie tun oder was in ihrer Umgebung geschieht, ist dies nun möglich. Diese Nachrichten beinhalten zudem eine Menge an zusätzlichen Metadaten, wie z.B. den Ort der Nachricht. So ist es nicht verwunderlich, dass diese Nachrichten für kommerzielle und für wissenschaftliche Zwecke, sowie für weitere Analysen und Auswertungen genutzt werden. Durch die jederzeit zur Verfügung stehende große Anzahl von aktuellen Nachrichten ist es zu einer großen Anzahl von neuen wissenschaftlichen Arbeiten, wie auf dem untersuchten Gebiet der Ereigniserkennung, gekommen. Es ist aber auch zu erkennen, dass es einen großen Spielraum für weitere Verbesserungen gibt. So wird z.B. die Tatsache, dass man die Tweets per Streaming-API in Echtzeit bekommen kann, selten in einen Vorteil für die Ereigniserkennung umgewandelt. Nur einige [63], [41], [36], [42], [46] der betrachteten Arbeiten versuchen die Ereignisse so schnell wie möglich zu erkennen, um schneller als die etablierten Medien Ereignisse zu erkennen und ggf. zu publizieren. Oft wird schlicht der zeitliche Vorsprung verspielt, indem die eigenen Ergebnisse der Ereigniserkennung erst wieder durch andere Medien verifiziert werden. Wenn man aber erst eine Bestätigung des Ereignisses über andere Quellen/Medien benötigt, so bringt die Ereigniserkennung selbst nichts und man kann gleich die anderen Quellen nutzen. Welchen immensen Vorteil die Schnelligkeit des Mikroblogging Medium und die darauf aufbauende Ereigniserkennung bieten kann, hat [41] eindrucksvoll gezeigt, indem Erdbeben so schnell detektiert werden konnten, dass man die betroffenen Menschen früher warnen kann, als es bisherige Warnsysteme getan haben. Doch nicht nur die Schnelligkeit von Twitter wird nicht voll ausgenutzt, sondern auch die Einbeziehung der Metadaten, vor allem die Geokoordinate an den Tweets wird meist nur am Rande genutzt. Die wenigsten Tweets haben zwar eine exakte Geokoordinate, doch durch die große Anzahl der versendeten Tweets, steigt auch die Anzahl der Tweets mit Geokoordinaten. Bei den Analysen ist oft zu erkennen, dass der Ort nicht mit in die Ereigniserkennung direkt mit einfließt. Meist wird am Ende nur versucht, die erkannten Ereignisse einem Ort zuzuweisen, den man in einem der Tweets findet. Außer bei den zuletzt vorgestellten Arbeiten [39], [40], [41], [43], [44], [46], [67], [69], [70], [71] in 2.4.7 spielt der Ort kaum eine Rolle.

Ein weiterer großer Spielraum für Verbesserungen ist der Umgang mit den eintreffenden Daten. Es findet mitunter eine große Aufbereitung und Filterung der Daten statt, bevor sie überhaupt zum eigentlichen Ereigniserkennungsschritt kommen. Zum großen Teil wird sogar nur nach bestimmten Termen gefiltert, so dass nur Tweets die diesen Term beinhalten überhaupt abgerufen werden. Weiterhin werden durch die Wortstammreduktion und andere Verarbeitungsschritte die Tweets nur von einer ganz bestimmten Sprache beachtet. Auch der Inhalt der Tweets wird überarbeitet. So werden in diesem Schritt oft schon alle nicht alphanumerischen Zeichen entfernt oder alle Twitter speziellen Eigenschaften wie z.B. Hashtags entfernt. Gerade die Hashtags selbst werden vom Autor der Nachrichten dazu genutzt, um Tweets einen oder mehreren Themen zuzuordnen. Genau solch wertvolle Inhalte, die auf neue Ereignisse hindeuten können, werden so im ersten Schritt der Analyse rausgefiltert weil sie als Noise angesehen werden. Auch eine Klassifizierung der Tweets mit Hilfe eines Klassifikators (z.B. Bayesfilter) kann die Tweet-Menge weiter einschränken. Es geht bei all der Filterung darum, vermeintliches Rauschen oder auch Müll bzw. Spam genannt zu entfernen. Diese Schritte führen aber dazu, dass die Systeme der Ereigniserkennung

dadurch nicht universell einsetzbar sind, da sie meist auf eine Sprache (z.B. durch Wortstammreduktionslisten usw., vorgefertigte Filterlisten) oder auf ein konkret zu suchendes Ereignis eingeschränkt wurden. Eine so radikale Filterung und Bearbeitung der Eingangsdaten sowie eine Beurteilung ob dieser oder jener Tweet oder Bestandteile eines Tweets Müll/Rauschen sind, ist jedoch kritisch zu hinterfragen.
Wünschenswert wäre hier ein Ereigniserkennungssystem, welches nicht auf solch eine Filterung oder Vorverarbeitung der Eingangsdaten angewiesen wäre und eine Echtzeit-Ereigniserkennung auf den Echtzeitdaten eines oder mehreren sozialen Netzwerke ausführen könnte. Es sollte jedes Ereignis erkennen können und müsste zugleich sprachunabhängig sein. Die detektierten Ergebnisse solch eines Systems sollten die aktuellen Ereignisse sein, die gerade stattfinden, egal in welcher Sprache sie beschrieben sind. Eine zusätzliche Validierung der Ergebnisse mit anderen Quellen darf hier nicht stattfinden, um den erwähnten Zeitvorteil nicht zu verspielen. Auch eine weitere Klassifizierung der Ergebnisse in Real-World Ereignisse und eines Ereignisses, welches dem Ausgangspunkt im Netz hat, soll nicht stattfinden. Es ist zumal fraglich, wo man genau bei den Ereignissen die Grenze zieht. Ist eine Produktankündigung über Twitter nun ein Real-World Ereignis oder nicht, bzw. wenn nicht, warum wird es als Ereignis zweiter Klasse angesehen nur weil die etablierten Medien nicht darüber berichten? Mit solch einem universalen Ereignisdetektionssystems, welches den Ort der Tweets direkt zur Ereignisdetektion mit nutzt, und nicht in einen angelagerten Schritt zum Schluss, ließe sich erkennen, wo die Ereignisse gerade stattfinden bzw. könnte es möglich sein, durch geeignete Parameter, das Ereignisdetektionssystem auf die eigenen Bedürfnisse einzustellen, z.B. wenn man aus einem bestimmten Gebiet auch regional kleinere Ereignisse detektieren möchte und ähnliches. Durch den Wegfall komplexer Filter und Klassifizierungsmaßnahmen bei der Aufbereitung der Daten, wäre auch ein performanteres Ereignisdetektionssystem vorstellbar, um die Ereignisse von größeren Regionen zu detektieren.

2.6 Stimmungsdetektion in sozialen Netzwerken

Die neuen entwickelten Algorithmen und Methoden zur Ereignisdetektion, die im nächsten Kapitel vorgestellt werden, lassen sich in einer abgewandelten Form auch für andere Probleme einsetzen. Als Beispiel wurden die Algorithmen für eine ortsabhängige Stimmungsdetektion eingesetzt. Diese Stimmungsanalyse soll als ein Beispiel dienen, um das weitere Potential der Algorithmen auch für andere Themengebiete aufzuzeigen.
Das Thema der Stimmungsdetektion ist wiederum ein eigenes Fachgebiet und wird unter anderem auch auf den nutzererzeugten Daten der sozialen Netzwerke eingesetzt. In diesem Kapitel soll nicht das ganze Gebiet der Stimmungsanalyse im Detail vorgestellt werden, sondern es sollen nur einige ausgewählte Arbeiten näher betrachtet werden, die sich speziell mit der Stimmungsanalyse in Tweets befassen, um einen Einblick in diesen Bereich zu bekommen.

2.6.1 Ziele

Mit Hilfe der Tweets ist es nun erstmals möglich die Stimmung zu allen möglichen Themen in Echtzeit zu ermitteln, da man Zugriff auf die Echtzeitnachrichten dieses sozialen Netzwerkes hat [72]. Eingesetzt wird die Stimmungsanalyse z.B. zur Analyse von Kundenmeinungen zu einer Firma oder einen bestimmten Produkt. Es gibt unzählige Portale wo über alle möglichen Produkte gesprochen wird. Zuvor war es nur über solche Portale möglich an Texte zu kommen, die Meinungen zu Produkten oder Filme beinhalten. Doch diese Texte waren nicht immer aktuell. Auch in Twitter schildern die Nutzer ihre Erfahrungen oder Probleme mit bestimmten Firmen oder Produkte. Laut

[72] haben 19% der Tweets Aussagen über Produkte oder Firmen zum Inhalt. Für eine Firma ist es interessant zu analysieren wie die Stimmung dieser Kundenmeinungen ist bzw. ob eine plötzliche Änderung dieser Stimmung auftritt. Dies könnte auf ein Problem mit einem Produkt oder Service der Firma hindeuten oder aber es ist der Erfolg eines Produktes abzulesen, wenn die Meinungen zu einem Produkt besonders gut sind. Eine automatische Analyse hat den Vorteil, dass große Datenmengen aus verschiedenen Quellen ständig analysiert werden können und somit frühzeitig etwaige Probleme erkannt werden können. Auch für die Erkennung von beleidigenden Nachrichten und zur Analyse von Markttrends [73] wird die Stimmungsanalyse genutzt.
Durch eine Stimmungsanalyse lässt sich auch eine Art Stimmungskarte erstellen, wo die aktuelle Stimmung einer Region aus den Tweets (oder anderen Quellen) der Region errechnet wird. Besonders die Beobachtung von Änderungen dieser Stimmung und dem Ort der Änderung können interessante Informationsquellen darstellen.
Selbst die Analyse eines Ereignisses kann neue Ergebnisse bringen. So wurde in [74] eine Präsidentschaftsdebatte analysiert, um zu sehen wie die Kandidaten bei den jeweiligen Themen, aus Sicht der Zuschauer, abgeschnitten haben. Man analysierte dazu die Tweets, die die Nutzer während der Debatte mit einem speziellen Hashtag geschrieben haben, um zu zeigen, dass es um die Debatte geht.
In [72] konnte ebenfalls gezeigt werden, dass die Stimmung in Twitter mit den Aktienpreisen und mit Real-World Ereignissen korreliert. D.h. eine negative Stimmung konnte man mit niedrigeren Aktienkursen an den Börsen wiedersehen. Es ist das erste Mal, dass solch eine Echtzeitquelle zur Verfügung steht. Durch eine ständige Analyse ließe sich auch erkennen, durch welches Ereignis eine Stimmungsänderung ausgelöst wurde.
Mit diesen Beispielen sollte aufgezeigt werden, dass eine Stimmungsanalyse auf den Tweets eine breite Palette von Anwendungsfällen bieten kann.

2.6.2 Stimmungsdetektion auf Tweets

Zur Bewertung der Tweets, ob sie positiv oder eher negativ sind, gibt es eine Vielzahl von unterschiedlichen Verfahren, die auch miteinander kombiniert werden. Die Analyse von Tweets birgt besondere Schwierigkeiten, da die Textmenge sehr gering ist. Zuvor bestehende Analysen analysierten mehr Text, um daraus die Stimmung zu bestimmen. Die trivialste Methode, um eine Stimmung zu detektieren, ist, wenn man Personen fragt, bestimmte Tweets manuell zu bewerten. Dies wird z.B. gemacht, um einen spezifischen Korpus zu erhalten, so dass man eine Grundlage hat, was negative oder positive Tweets sind. In [74] nimmt man diese manuelle Bewertung, um die Tweets einer Präsidentschaftsdebatte zu bewerten. Man nutzt dazu den Service von Amazon Mechanical Turk, um die Tweets der Präsidentschaftsdebatte zu bewerten.
Amazon Mechanical Turk ist eine Online-Plattform, wo Menschen kleine Aufgaben übernehmen können und dafür entlohnt werden. Diese Aufgaben können z.B. sein: Korrektur von Rechtschreibfehlern, Suchaufträge im Internet, Erkennung von Objekten auf Bildern [75]. Es sind somit alles Aufgaben, die momentan noch besser von Menschen gelöst werden können als von Maschinen.
Man wollte damit zeigen, dass man durch die Analyse von Tweets dieses Ereignis besser analysieren kann und die Bewertung der Kandidaten durch die Zuschauer so untersuchen kann. Da die Analyse über den Amazon Mechanical Turk lief, konnte dies natürlich nicht in Echtzeit erfolgen und stellt eher ein Proof of Concept dar.
In [76] werden weitere Methoden erklärt zur Bestimmung der Stimmung in einen Tweet. Die Systeme benötigen in der Regel eine vorbewertete Liste von Termen (z.B. ANEW – Affective Norms of English Words [77]) oder vorklassifizierte Texte. Manche Blogsoftware bietet auch Metadatenfelder an, so dass der Autor die Stimmung zu seinem

Blogeintrag mit angeben kann. Diese können auch als Trainingsdaten für diverse Systeme verwendet werden. Man nutzt Twitter-Tags und Emoticons [78] (ein Emoticon ist eine ASCII-Zeichenfolge, die eine Stimmung ausdrücken soll wie z.B. das Smiley :-)) zur Beurteilung der Tweets und setzt keine vorbewerteten Tweets ein. Genutzt wird eine Liste von 50 Twitter-Tags und 14 Emoticons. Daraus wird ein Klassifikator trainiert. Sehr kurze Tweets (< 5 Terme) werden nicht berücksichtigt sowie alle nicht englischen Tweets. Andere Bestandteile wie URLs werden aus den Tweets entfernt bzw. durch einen Tag ersetzt. Durch eine Jury (mit Hilfe des Amazon Mechanical Turk) werden die Tweet-Tags wie z.B. „#sucks“ bewertet. Beinhaltet der Tweet mehr als ein Tag, so wird er ignoriert.
Es wurde festgestellt, dass eine Klassifizierung einfacher ist, wenn ein Emoticon enthalten ist als ein Hashtag. Zwei gegensätzliche Hashtags können nämlich oft in einem Tweet zusammen auftauchen z.B. „happy days of training going to end in a few days #sad #happy“ [76]. Neben den erwähnten Tags und Emoticons werden noch andere Eigenschaften wie Zeichensetzung, N-Gramme, häufig auftretende Terme usw. genutzt. Diese Features werden wieder zu einem Feature-Vektor zusammengegesetzt und anhand dieses Vektors wird der Tweet wieder bewertet. Dies ist in gewisser Weise wieder vergleichbar mit der Cluster-basierten Ereigniserkennung.
Die Nutzung von Emoticons zur Bewertung der Texte hat auch einen weiteren Vorteil. Man kann so die Texte themen- und zeitunabhängig nutzen, um damit einen Klassifikator zu trainieren. In [73] wird gezeigt, dass die Trainingsdaten der Klassifikatoren einen entscheidenden Einfluss auf die Qualität der Klassifizierung der Tweets und somit auf die Stimmungserkennung haben. Nur wenn der Klassifikator mit Texten zu dem gleichen Thema trainiert wurde, kann er auch später gute Ergebnisse liefern. Auch die Zeit spielt eine Rolle, da sich die Texte im Laufe der Zeit ändern können. Somit sollte der Trainingsdatensatz einerseits aktuell sein und zum anderem das gleiche Thema bzw. aus der gleichen Domäne sein wie die zukünftigen klassifizierten Texte es sein werden.
Um einen Tweet zu bewerten, sagt man in der Regel nicht einfach nur, dass er positiv oder eher negativ ist, sondern man setzt POMS (Profile of Mood States) ein, um die Stimmung genau zu beschreiben [77]. POMS sind 65 Adjektive mit dem ein Text bewertet wird. Zu jedem Adjektiv muss ermittelt werden, ob dieses Adjektiv den Text beschreibt. Zum Schluss werden die Ergebnisse auf einen 6-dimensionalen Stimmungsvektor (Tension (Spannung), Depression (Depression), Anger (Ärger), Vigour (Kraft/Energie), Fatigue (Müdigkeit), und Confusion (Verwirrung)) transferiert und man hat somit wieder einen Feature-Vektor, den man z.B. mit anderen Feature-Vektoren vergleichen kann. Von POMS gibt es auch reduzierte Versionen mit weniger Adjektiven aber auch erweiterte Version wie sie in [77] eingesetzt wurden wo 793 Terme eingesetzt wurden. Die 793 Terme, wo die Stimmung bekannt ist, sind unter anderem Synonyme für die standardisierten 65 Adjektive, die sonst eingesetzt werden. Bevor dort ein Text bewertet wird, werden alle nicht alphanumerischen Zeichen entfernt, alles in Kleinbuchstaben gewandelt, Stoppwörter entfernt und eine Wortstammreduktion durchgeführt. In einem weiteren Schritt werden nur Tweets analysiert, die ausdrücklich ihre Stimmung ausdrücken indem sie Terme enthalten wie z.B. „feel“, „I’m“, „Im“, „am“, „being“ usw. Tweets mit URLs werden ebenfalls nicht beachtet. Von den übrig gebliebenen Tweets wird die Stimmung anhand den 793 Terme ermittelt und somit ein Vektor für diesen Tweet generiert (es wird überprüft, ob einer der 793 Terme in dem Tweet enthalten ist). Im nächsten Schritt errechnet man so die Stimmung des ganzen Tages aus und kann dann die Stimmung der Tage in einen größeren Zeitraum miteinander vergleichen. Wenn Ereignisse wie die Präsidentschaftswahl und

Thanksgiving in den untersuchten Zeitraum enthalten sind, kann man die Stimmungsänderungen bei diesen Ereignissen ablesen.

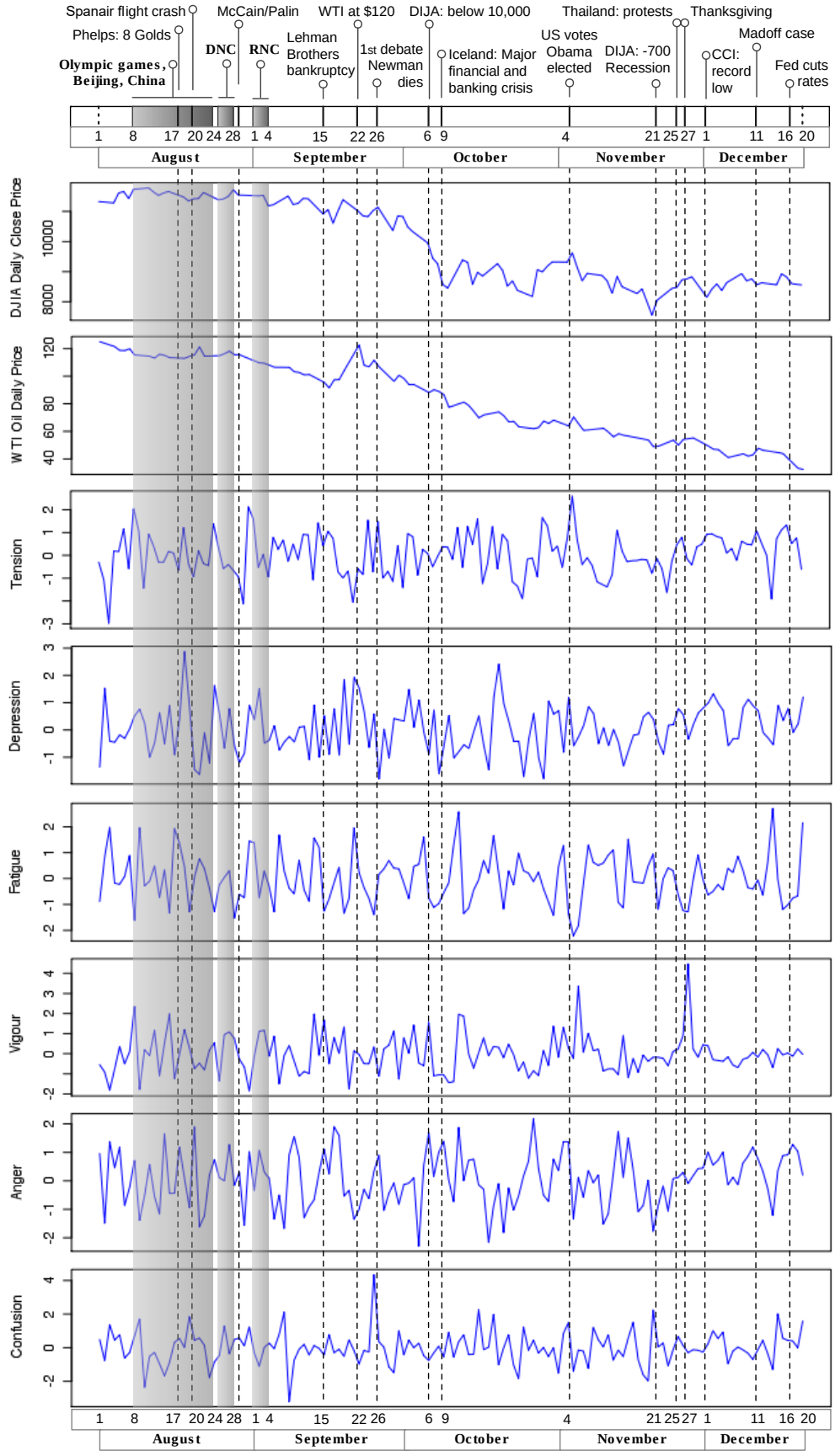

Abbildung 22: POMS-Verläufe vom 1. August bis 20. Dezember 2009. Markiert sind die Hauptereignisse in diesen Zeitraum [77]

In der Analyse wurde erkennbar, dass die Ereignisse wie z.B. Thanksgiving, Wahl des Präsidenten, Tod von bekannten Persönlichkeiten, Naturkatastrophen wie Erdbeben aber auch die Aktienkurse und der Ölpreis sich in der Stimmung niederschlägt. In den Tweets kann man diese Stimmungsänderungen detektieren.
Zusammenfassend kann man sagen, dass die Arbeitsschritte zur Stimmungsdetektion vergleichbar mit der Ereignisdetektion sind. Am Anfang werden die Tweets wieder gefiltert und aufbereitet. Je nach Analyse-Variante geschieht dies mehr oder weniger. In [79] werden z.B. die URLs entfernt, Rechtschreibfehler korrigiert, in [80] werden Sonderzeichen entfernt, eine Wortstammreduktion durchgeführt, sehr kurze Terme (<= 2 Zeichen) entfernt, Zeichenwiederholungen (z.B. „cooooooool“) korrigiert [81], Stoppwörter entfernt usw. Bevor die eigentliche Analyse stattfindet, sind die Eingangsdaten also mehr oder weniger gefiltert und normalisiert. Der eigentliche Erkennungsschritt stellt sich vielfältig dar. Es gibt verschiedenste Methoden. Wie z.B der Einsatz von Klassifikatoren die zuvor trainiert an vorbewerteten Tweets wurden, Bewertung von Tweets über vordefinierte Terme und ihren Stimmungswert oder Beurteilung über Emoticons oder Hashtags. Meist kombinieren die Systeme mehrere dieser aufgezählten Möglichkeiten. Auch das gewünschte Ergebnis ist unterschiedlich. In [77] ist das Ergebnis der 6-dimensionale Stimmungsvektor (POMS) welcher in Abbildung 22 zu sehen ist. Doch es gibt auch Arbeiten, bei denen die Tweets in zwei Klassen (positiv/negativ) oder in drei Klassen (positiv/neutral/negativ) einsortiert wurden. Eine weitere Unterscheidung der Analysen ist, wie weit die Analyse der Texte getrieben wurde. In [79] wurden Adjektive mit einem Korpus bewertet, da die Bewertung von Adjektiven themenabhängig ist. Weiterhin werden die Emoticons differenzierter betrachtet anstatt nur positiv oder negativ. Z.B. wird :D positiver bewertet als :-). Adverbien sind dagegen nicht themenabhängig, also wird in [79] eine vordefinierte Liste genommen, um die Terme zu bewerten. Besonders die Beachtung von Adverbien ist wichtig, da sie die Bedeutung eines Satzes umkehren können z.B.: „This is not a good book“ [79]. Um die gesamte Stimmung des Tweets zu bewerten, werden die einzelnen Teilergebnisse zusammengerechnet. In [81] nutzt man eine vordefinierte Liste mit Termen und ihren Stimmungswert. Wenn der entsprechende Term nicht zu finden ist, wird ein Synonym für den Term gesucht und mit Hilfe dieses Synonyms wird wieder die Liste abgefragt, ob der Synonymeintrag enthalten ist. Somit konnte in [81] zu 81,1% der Terme einen entsprechenden Stimmungswert gefunden werden. Zum Schluss werden die Terme des Tweets in einen Baum abgebildet und der Stimmungswert des ganzen Tweets wird dann errechnet. Auch die genutzten Satzzeichen werden z.B. in [80] beachtet und gehen in das Endergebnis mit ein.

2.7 Twitter

Da bei den praktischen Arbeiten, in den nächsten Kapiteln, die Analysen auf den Daten von Twitter durchgeführt werden, soll dieses Netzwerk nun näher vorgestellt werden. Twitter ist ein soziales Netzwerk, genauer gesagt ein Echtzeit-Mikroblogging-Dienst. Die Firma Twitter, die den Dienst betreibt, wurde im März 2006 [82] gegründet. Ähnlich wie bei Blogs, können Nutzer hier Beiträge verfassen, die von anderen Nutzern gelesen werden können. Das Besondere bei Twitter ist, dass die Länge der Beiträge, genannt Tweets (von englisch „to tweet“ (zwitschern)), auf 140 Zeichen beschränkt ist. Der Tweet kann auch Verweise auf andere Webressourcen wie Websites-, Bilder- oder Videolinks enthalten. Auch optionale Daten wie die aktuelle Position des Nutzers während des Schreibens des Tweets, kann mit an den Tweet angehangen werden. Twitter hatte 2013 [83] 200 Millionen aktive Nutzer und es werden täglich 400 Millionen Tweets versandt. Die Nutzer von Twitter können anderen Twitter Nutzer folgen, d.h. sie können die Nachrichten des Nutzers abonnieren und werden somit in Echtzeit informiert, wenn der gefolgte Nutzer einen neuen Tweet geschrieben hat. Um Tweets von Nutzern zu lesen, ist es aber nicht notwendig, dass man bei Twitter angemeldet ist.
In der sogenannten Timeline eines angemeldeten Nutzers, sieht er alle aktuellen Tweets in chronologischer Reihenfolge von Nutzern denen er folgt. Er kann einzelne Tweets retweeten, d.h. an seine eigenen Follower weiterleiten, um z.B. wichtige Meldungen schnell weiter zu verbreiten. Alternative kann er auf einen Tweet antworten oder ihn auch markieren („faven“). Damit kann er auszudrücken, dass er diesen Tweet sehr interessant oder gut fand (vgl. mit „liken“ bei Facebook) oder er nutzt die dies einfach nur, um den Tweet zu markieren.
Genutzt wird Twitter von den Nutzern auf verschiedene Weise, z.B. um Informationen zu teilen oder nur um Informationen zu suchen oder um mit Freunden in Kontakt zu stehen [84]. Dabei wird auch deutlich, dass nur 10% der Nutzer 90% der Informationen posten. D.h. die meisten Nutzer nutzen Twitter eher passiv. [85] Geschäftlich wird Twitter überwiegend genutzt, um mit den Kunden schnell in Kontakt zu treten, zu Marketingzwecken oder zur Beobachtung und Analyse der Konkurrenz. [85]

2.7.1 Hashtags

Da pro Tweet nur 140 Zeichen zur Verfügung stehen, werden häufig Abkürzungen und Hashtags genutzt (Wörter oder Zeichenketten mit vorangestelltem Doppelkreuz z.B. #Dissertation), die den Tweet einem bestimmten Thema oder einem bestimmten Kontext zuordnen bzw. den Tweet verschlagworten [86]. Weitere Nutzungsmöglichkeiten sind, dass über Hashtags auch Stimmungen ausgedrückt werden können oder sie werden für Werbezwecke genutzt, indem man den Nutzern schon Hashtags empfiehlt (z.B. bei Veranstaltungen), die sie nutzen sollen, um über die Veranstaltung oder über das Produkt zu schreiben. Mit Hashtags ist es bei Twitter möglich, gezielt nach Tweets zu suchen bzw. bestimmten Hashtags zu folgen. Folgt man bestimmten Hashtags, so bekommt man automatisch in Echtzeit alle Tweets, die dieses Hashtags beinhalten. Dies wird z.B. bei sogenannten Twitterwalls [87] verwendet, die zur Anzeige von Tweets mit bestimmten Hashtags auf einer großen Anzeigetafel bzw. Leinwand z.B. bei Diskussionsrunden eingesetzt werden. Dies ermöglicht es, direkt Feedback vom Publikum in die Diskussion einfließen zu lassen.

2.7.2 Schnittstellen

Die Tweets der Nutzer oder die Hashtags, denen man folgt, können entweder direkt auf den Seiten von Twitter gelesen werden oder mit einer der zahlreichen Programme, die

es für sehr viele Plattformen gibt. Überall dort ist es auch möglich, selbst neue Tweets zu veröffentlichen, Tweets zu retweeten, zu beantworten, neue Nutzer zu abonnieren usw. In einigen Betriebssystemen wie z.B. ab Windows 8 [88], ab Windows Phone 8 [89], ab Mac OS X 10.8 [90] und iOS ab Version 5 [91] ist die Möglichkeit, Tweets zu verfassen sogar schon eingebaut.
Auch die Nutzung des Dienstes über SMS, ist in den USA, Kanada und Indien möglich [92]. Ebenso ist der Abruf der Tweets per RSS möglich, um die Tweets per RSS-Reader zu lesen oder anderweitig weiter zu bearbeiten. [93]
Einer der mutmaßlichen Gründe des Erfolges von Twitter ist sicherlich die Offenheit des Dienstes. Durch die kostenlos bereitgestellte API (REST) des Dienstes [94], ist es möglich, mit der eigenen Software mit Twitter zu interagieren, was von unzähligen anderen Diensten und Apps auch verwendet wird. Es können darüber Tweets gepostet werden aber auch Tweets abgerufen werden, beispielsweise die abonnierten Tweets aus der eigenen Timeline oder man kann nach bestimmten Tweets suchen. Bei der Suche nach Tweets kann man nicht nur nach Wörtern in Tweets oder nach Nutzer suchen, sondern auch ein Gebiet angeben, aus der die gewünschten Tweets gepostet wurden.
Neben diesem API-Teil gibt es als Besonderheit auch eine Streaming-API [95]. Bei der Streaming-API wird eine dauerhafte HTTP-Verbindung mit Twitter hergestellt, worüber man in Echtzeit aktuelle Tweets bekommt. Entweder die aktuellen Tweets eines oder mehrerer Twitter-Nutzer oder alle öffentlichen Tweets, den sogenannten Public-Stream. Den Zugriff auf den gesamten Stream, den sogenannten „Firehose“ (Feuerwehrschlauch), ist beschränkt und i.d.R. nur bestimmten Firmen gestattet, die Vereinbarungen mit Twitter haben wie z.B. Microsoft, die die Echtzeitdaten nutzen, um die Suchergebnisse ihrer Suchmaschinen mit Echtzeit-Nachrichten anzureichern [96]. Auch weitere Firmen haben Zugriff auf diese Daten. Diese Firmen sammeln diese Daten von den verschiedenen sozialen Netzwerken, um diese aufzubereiten und die Ergebnisse der Aufbereitung weiter zu vermarkten. Bei dieser Aufbereitung geht es z.B. um die Analyse für andere Firmen, wie über diese Firmen oder deren Produkte in diesen sozialen Netzwerken berichtet wird. Oder es geht um die Analyse von Konkurrenten, wie diese sich in den sozialen Netzwerken verhalten und wie die Meinung zu den eigenen Produkten in diesen Netzwerken ist. Somit möchte man etwaige Probleme frühzeitig erkennen.
Wo man dagegen uneingeschränkten Zugang hat, ist der sogenannte „Gardenhose“ (Gartenschlauch) was 1% der öffentlichen Tweets, also des Firehose entspricht. Eine weitere Möglichkeit besteht darin, dass man über die Streaming-API einen gefilterten Echtzeitdatenstrom mit Tweets bekommen kann. Auch hier gelten dieselben Filter wie bei der REST-API. Man kann nach Stichwörtern filtern, um alle Tweets in Echtzeit zu bekommen, in denen das Stichwort enthalten ist. Aber es besteht auch die Möglichkeit nach Regionen zu filtern. Mit diesem Filter ist es möglich, in Echtzeit Tweets von bestimmten Regionen zu bekommen. Eine Region wird hier als eine Box beschrieben. Die Ecken, dieser gedachten Box, werden jeweils mit Breiten- und Längengradangaben beschrieben. Die Region ist dann die Fläche die diese Box aufspannt. Alle diese Tweets, die man über die Streaming-API mit diesem Regionen-Filter bekommt, sind Tweets, die Positionsdaten beinhalten. Entweder enthalten diese Tweets die exakte Position in Breiten- und Längengrade oder sie geben eine Fläche vor, wo der Nutzer sich gerade befindet. Somit kann der Nutzer selbst steuern wie genau er seine Position mitteilen möchte. Diese Möglichkeit Tweets in Echtzeit aus dem Gardenhose von bestimmten Regionen zu bekommen, wurde vom Autor im Rahmen dieser Arbeit intensiv genutzt.

2.7.3 Infrastruktur

Um die sehr großen Datenmengen, die bei Twitter anfallen, zu verarbeiten, braucht es eine geeignete Infrastruktur und Software. Twitter setzt hier auf Open-Source-Software und hat hier auch selbst diverse selbstentwickelte Software (Storm, Bootstrap, Finagle usw.) als Open-Source veröffentlicht [97]. Die u.a. eingesetzte Software Storm von Nathan Marz, dient zur verteilten Berechnung von Echtzeitdatenströmen in Clustern. Vergleichbar ist Storm mit Apache Hadoop [98], welche ebenfalls große Datenmengen als Batch-Jobs verarbeitet und auf den MapReduce-Algorithmus (ein Programmiermodell um große Datenmengen auf einem Computercluster zu verarbeiten) [75] [99] von Google basiert. Storm verarbeitet dagegen Datenströme in Echtzeit (Stream Processing) und kommt somit theoretisch nie zu einem Ende, außer man beendet die Software manuell.
Storm ist eine fehlertolerante und skalierbare Software. In Abbildung 23 ist zu erkennen, dass es zwei Arten von Knoten, den Master-Knoten (Nimbus) und die Worker-Knoten (Supervisor) gibt [100].

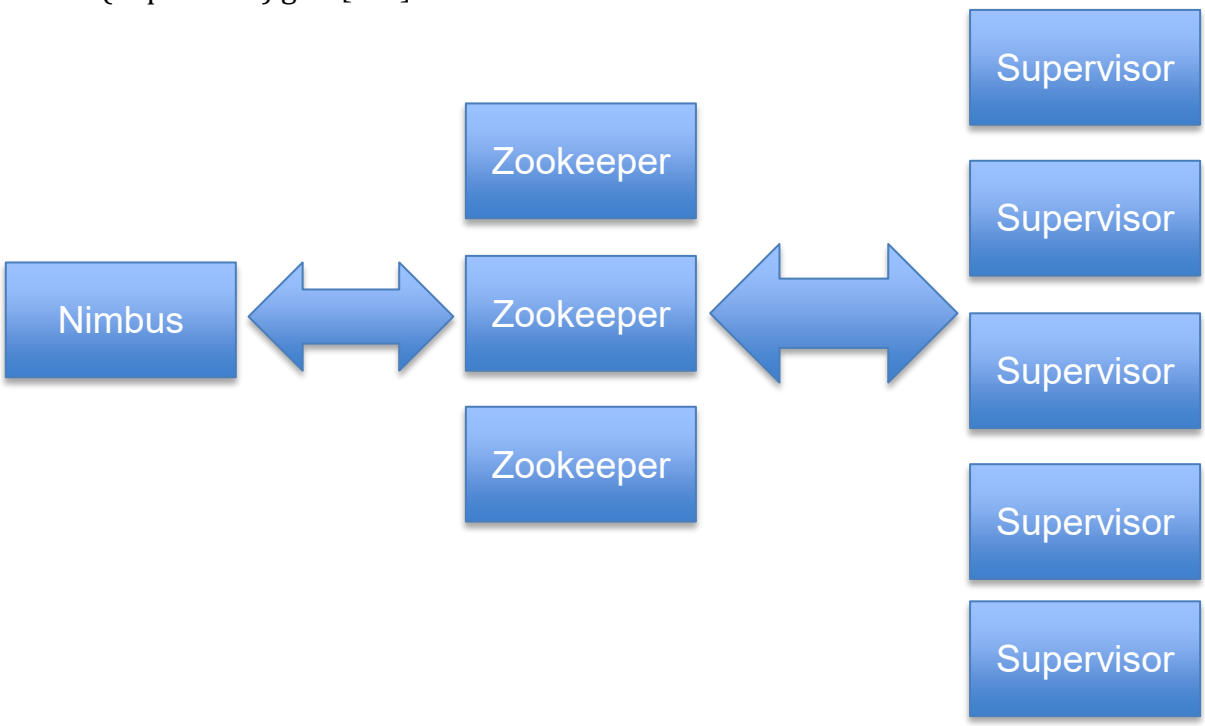

Abbildung 23: Zusammenarbeit der Knoten in einem Storm Netzwerk, nach [100]

Nimbus empfängt die Aufgaben, verteilt diese auf die Supervisor-Knoten und überwacht die Supervisor-Knoten. Die Verbindung zwischen Nimbus und den Supervisor-Knoten findet über einen Zookeeper-Cluster statt, der auch die Zustände der Knoten abspeichert. Der Nimbus-Knoten und die Supervisor-Knoten selbst sind zustandslos und können so im Fehlerfall einfach neu gestartet werden bzw. die Aufgabe des ausgefallenen Knoten kann schnell eine andere Maschine übernehmen. [100] Storm verarbeitet Streams. Ein Stream ist hier ein kontinuierlicher Datenstrom aus Tupeln, die durch Storm verarbeitet/umgewandelt werden. Das Endergebnis ist wieder ein Ergebnis-Stream von Tupeln. Zur Verarbeitung unterscheidet Storm intern zwischen Spouts und Bolts. Spouts sind die Datenquellen, die den Stream zu den Bolts senden, die den Stream verändern/verarbeiten und das Zwischenergebnis an andere Bolts weiterschicken können (siehe Abbildung 24).

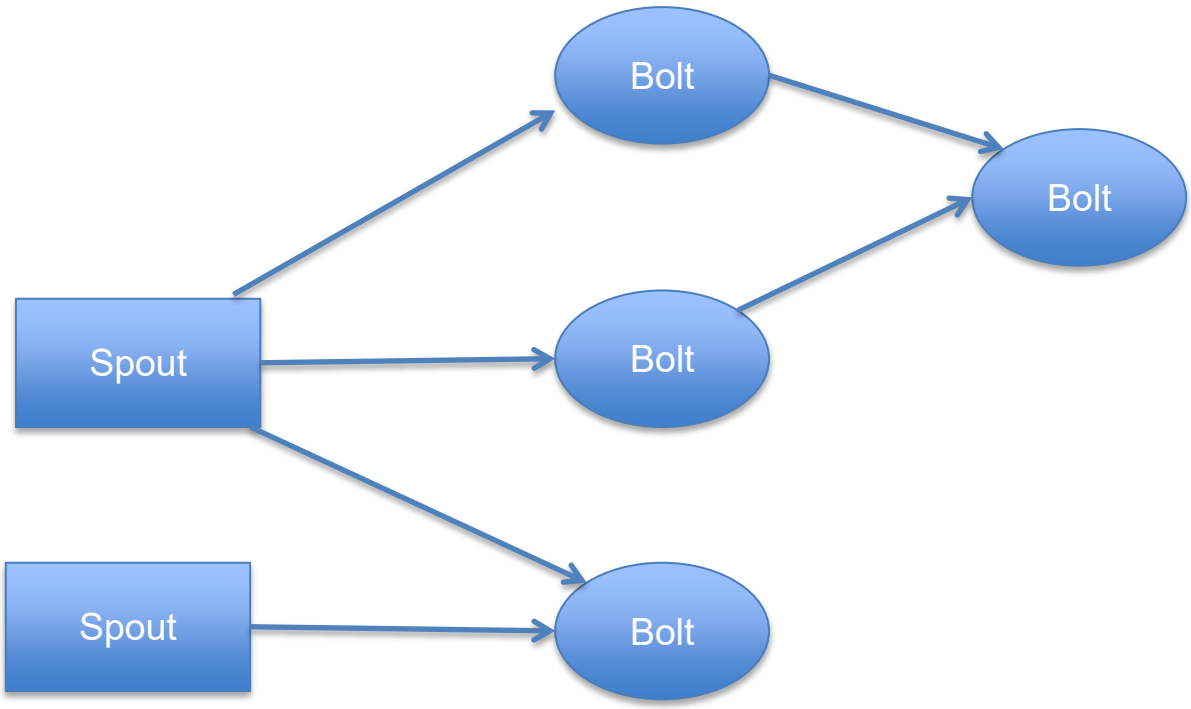

Abbildung 24: Vernetzung (Topologie) von Spouts und Bolts in Storm, nach [100]

In den Spouts und Bolts läuft der anwendungsspezifische Code, wie die Streams zu verarbeiten sind. Auch die Vernetzung der Spouts und Bolts ist abhängig von der jeweiligen Aufgabenstellung und wird als Topologie bezeichnet. Die Topologie beschreibt, wie der Stream bearbeitet wird und welche Spouts und Bolts beteiligt sind und wohin die Zwischenergebnisse gesendet werden. Dabei können die Spouts und Bolts einzelne Threads auf derselben Maschine sein, aber auch in einen Cluster laufen, um mit einer hohen Last umzugehen. Storm garantiert ebenfalls, dass jede Nachricht, die von Spouts oder Bolts versendet werden, auch bearbeitet wird. D.h. auch wenn ein Teil der Topologie offline geht, würde Storm die Nachrichten an neue Maschinen, die die Aufgabe der offline gegangenen Teile der Topologie übernehmen, zur Bearbeitung weiterreichen. [100]

2.7.4 Weitere Stream-Processing-Software

Neben Storm gibt es auch weitere Software, die ein Stream-Processing durchführen kann. Zu nennen wäre da z.B. Apache S4 [101] oder Spark Streaming [102] welches, laut Aussage der Entwickler [103], mehr Daten pro Sekunde verarbeiten kann als Apache S4 und Storm. Spark Streaming zerteilt den Datenstrom dazu in kleine Teilaufgaben und versucht diese mit Batch-Jobs so schnell wie möglich abzuarbeiten:
„The key idea is to treat streaming as a series of short batch jobs, and bring down the latency of these jobs as much as possible. This brings many of the benefits of batch processing models to stream processing, including clear consistency semantics and a new parallel recovery technique that we believe is the first truly cost-efficient recovery technique for stream processing in large clusters.“ [104]
Spark führt zudem ein In-Memory-Data-Mining durch, d.h. es behält die Daten im RAM, um den IO-Flaschenhals „Festplatte“ oder andere Massenspeicher zu umgehen. Daher ist es selbst in den Batch-Jobs schneller als die vergleichbare Software z.B. Hadoop. [102]

3 Konzeption einer ortsabhängigen Analyse in Echtzeitdaten sozialer Netzwerke

In diesem Kapitel geht es um die Konzeption von Algorithmen für die Ereignisdetektion in den Daten von sozialen Netzwerken, im speziellem in den Daten des sozialen Netzwerkes Twitter. Um zu zeigen, dass die entwickelten Konzepte auch für andere Problemfelder eingesetzt werden können, wird im Anschluss daran gezeigt, wie man die entwickelten Algorithmen für die Stimmungsanalyse einsetzen kann.
In Kapitel 2.5 wurde aufgezeigt, dass die vorgestellten Ereignisdetektionsalgorithmen in Kapitel 2 potentiell noch Raum für Verbesserungen haben. Vor allem die Tatsache, dass es sich bei den Daten über die Streaming-API von Twitter um Echtzeitdaten handelt, wird selten in einen Vorteil umgewandelt. So ist es mit diesen Echtzeitdaten potentiell möglich, Ereignisse schneller zu detektieren und zu publizieren, als es etablierte Nachrichtenquellen (Medien) könnten. Welche eindrucksvollen Ergebnisse man bekommt, wenn man diesen Zeitvorteil nutzt, zeigt [41] und [46], welches schneller über gerade stattfindende Erdbeben warnen kann, als es die bestehenden Warnsysteme können. Im Gegensatz dazu wurde in vielen Arbeiten das Ergebnis, also die erkannten Ereignisse, erst durch andere Quellen verifiziert, wodurch man diesen Zeitvorteil verspielte.
Wie wertvoll dieser Zeitvorsprung sein kann, zeigt das Unternehmen Dataminr, welches den Tweet-Datenstrom in Echtzeit ebenfalls mit proprietärer Technik analysiert, um daraus Ereignisse oder Stimmungen, welche Relevanz für Aktienkurse haben könnten, zu extrahieren. So konnte Dataminr seinen Kunden einen 3-minütigen Vorsprung verschaffen, um die abstürzende Blackberry-Aktie im November 2013 abzustoßen. [105] Einen 20-minütigen Vorsprung hatte man bei der Nachricht, dass Osama Bin Laden gefasst wurde, gegenüber den etablierten Medien [105]. Besonders für Fondsmanager und Aktientrader sind diese Informationen, besonders wenn noch nicht etablierte Medien darüber berichtet haben, sehr viel wert.
Auch die mitunter starke Filterung der Eingangsdaten, in den Arbeiten von Kapitel 2 und deren Aufbereitung führt dazu, dass nur ein kleiner Teil der Daten in den eigentlichen Analyseschritt analysiert wird. So konzentrierte man sich in den Arbeiten meist auf eine bestimmte Sprache (bedingt durch Bearbeitungsschritte wie z.B. die Wortstammreduktion) oder man filterte die Tweets so sehr, dass nur Tweets durchgelassen wurden, die bestimmte Terme enthalten. Vor allem die Beachtung des Ortes, wo der Tweet geschrieben wurde, ging nur unzureichend in die meisten Analysen ein. Der Ort der erkannten Ereignisse wurde normalerweise im Anschluss hinzugefügt und spielte in der eigentlichen Phase der Ereignisdetektion fast immer keine Rolle.
Im nächsten Unterkapitel werden noch einmal die Ziele der Entwicklung zusammengefasst. Danach erfolgt in Kapitel 3.2 ein kurzer Blick in die Daten sowie eine Visualisierung derselben. Im daran anschließenden Abschnitt 3.3 geht es dann um die Konzeption der neuen Ereignisdetektionsalgorithmen.
Schließlich wird in Abschnitt 3.4 gezeigt. dass diese Algorithmen auch für andere Aufgabenstellungen genutzt werden können. Dazu wurde eine Möglichkeit erarbeitet, wie Stimmungen detektiert werden können.

3.1 Ziele

Ziel der Arbeiten ist es, Ereignisdetektionsalgorithmen zu entwickeln, die Ereignisse schnell detektieren können, um den Zeitvorsprung gegenüber anderen Quellen nicht zu verlieren. Die Algorithmen müssen ebenso effizient sein, damit sie mit der hohen Datenmenge umgehen können, die einströmt (Echtzeit-Tweets aus bestimmten Regionen). Der Ort der Tweets soll ein wichtiger Bestandteil der Analyse sein und somit soll das Ergebnis der Analyse das erkannte Ereignis und dessen Ort sein. Durch bestimmte Parameter soll es möglich sein, die Ergebnisse zu beeinflussen. D.h. ob man auch kleinere Ereignisse detektieren möchte oder nur große. So kann es sinnvoll sein, sich in einer bestimmten Region auch kleinere Ereignisse anzeigen zu lassen, weil man z.B. dort lebt und informiert sein möchte, was vor Ort vorgeht.
Vor allem die Aufbereitung der Daten sollte einfacher werden als in vergleichbaren Arbeiten. Es sollte nicht nötig sein, die Tweets aufwendig aufzubereiten oder nur Tweets einer bestimmten Sprache zu analysieren. Die Algorithmen sollten mit jeglichen Tweets klarkommen, so dass durch eine etwaige Vorverarbeitung keine potentiell wertvollen Daten weggefiltert werden können.
Die nachfolgenden Kapitel stellen neue Algorithmen vor, welche aus einem Echtzeitdatenstrom von georeferenzierten eintreffenden Kurznachrichten eines sozialen Netzwerkes, neue Ereignisse und deren Ort extrahieren können. Dabei erfolgt eine sprachunabhängige auf Termhäufigkeitsänderungen basierte Echtzeit-Ereignisdetektion, die ohne Vorverarbeitung bzw. Filterung der chronologisch eintreffenden Eingangsdaten auskommt. Um die Analyse durchzuführen, werden ortsabhängige adaptive Korpora (Analyse- / Referenzkorpus) erzeugt, um eine Differenzanalyse durchzuführen. Durch die Ortsabhängigkeit der adaptiven Korpora führt die Ereignisdetektion zu besseren Ergebnissen.
Durch Modifikationen der ortsabhängigen Korpora, kann Einfluss auf die Ergebnismenge der Ereignisdetektion genommen werden, um unterschiedliche Einsatzszenarien zu realisieren bzw. die Analyse auf Orte mit unterschiedlichem Kurzmitteilungsaufkommen pro Fläche und Zeit anzupassen. Es kann zum einem z.B. die Zusammensetzung der Korpora aus georeferenzierten Kurzmitteilungen unterschiedlicher geographischer Entfernungen zum Analyseort verändert werden oder aber die Detektionsschwellen bei der Differenzanalyse können angepasst werden.
Durch Separierung und Aufbereitung der Ergebnismenge der Ereignisdetektion können die gefunden Ergebnissen einzelnen abgrenzbaren Ereignissen zugeordnet werden. Um die Ergebnisse besser zu beschreiben, werden die Ereignisse mit zusätzlichen extrahierten Mediendaten angereichert und entsprechend präsentiert.
Für die Analysen selbst wurde weiterhin eine verteilte skalierbare Systemarchitektur entworfen mit der unabhängige gekapselte Echtzeit-Analysen über mehrere Rechner hinweg möglich sind und deren Ergebnisse sofort als Website aufbereitet werden können.

3.2 Ein Blick in die Daten

Bevor es um die Konzeption der Ereignisdetektionsalgorithmen geht, lohnt sich ein erster Blick in die Daten, um ein paar Eigenschaften zu erkennen. Ein einfacher Blick in die eintreffenden Tweets, die man von der Streaming-API Schnittstelle empfängt, bringt für das erste nur wenig neue Erkenntnisse. Man erkennt durch alleiniges manuelles durchschauen von Tweets, dass die Inhalte sehr unterschiedlich sind. So gibt es z.B. allgemeine Tweets von Nutzern, die schreiben, was sie gerade tun oder was sie

beschäftigt. Tweets zwischen Nutzern aber auch von Systemen erzeugte Tweets sind erkennbar. Systeme, die Tweets automatisch erzeugen können z.B. automatische Wetterstationen sein, Plattformen die Meldungen (z.B. Pressetexte, veröffentlichte Mitfahrgelegenheiten, vermietbare Wohnungen) posten, Radiostationen, die den aktuell gespielten Titel posten usw. Diese grobe Einteilung der Nachrichten erhebt keinen Anspruch auf Vollständigkeit und dient eher als grober Überblick, welche Arten von Tweets man ad hoc, beim manuellen Durchsehen, erkennen kann.
Visualisiert man einen Teil der Tweets, so lassen sich mehr Eigenschaften ablesen. Das Ziel der nachfolgenden Visualisierung sollte es sein, die Tweets auf einer interaktiven Karte zu platzieren so, dass man die Tweets erforschen kann. Auf die Implementierung dieser Visualisierung wird das Kapitel 4 genauer eingehen. In den nachfolgenden Abbildungen wurden nur Tweets visualisiert die genaue Positionsangaben besaßen. In der späteren Analyse fließen aber auch Tweets mit ein die auch etwas ungenauere Positionsangaben besitzen.
Die Tweets wurden in der interaktiven Karte von Google Earth [106] visualisiert. Dabei wurde versucht, so viel wie möglich an Informationen in diese Visualisierung zu integrieren. Als erstes wurden die einzelnen Tweets als Kugel dargestellt. Die Farbe der Kugel kodiert die Menge an Followers, die der Nutzer hat. Nutzer mit wenigen Followers (ca. 10) haben eine hellrote Kugel, die mit mehr Followers immer röter wird. Bei komplett roten Kugeln haben die Nutzer mehr als 1000 Followers. Der Ort der Tweets auf der Karte ist natürlich der Ort, der an dem Tweet hinterlegt ist. Also der Ort, von wo aus der Tweet gesendet wurde. Klickt man eine Kugel an, so sieht man in einem Popup-Fenster den Text des Tweets sowie den Nutzernamen. Die Tweets wurden etwas über dem Boden der Karte platziert, damit alle Tweets bei der Darstellung von 3D-Gebäuden sichtbar bleiben. Je neuer der Tweet ist, umso höher ist er wiederum angeordnet. D.h. ist ein Tweet über den anderen angeordnet bedeutet dies, dass der obere Tweet später geschrieben wurde. Somit ist es möglich in der interaktiven Karte in der 3D-Ansicht das zeitabhängige Tweet-Aufkommen an dem jeweiligen Ort zu visualisieren. Mit Hilfe einer Animation können die Tweets in ihrer chronologischen Reihenfolge ein- bzw. ausgeblendet werden, was zeitlich veränderte räumliche Tweet-Aktivitäten besser sichtbar macht.

Abbildung 25: Visualisierte Tweets mit Geoinformationen von New York innerhalb einer Stunde (18 Uhr-19 Uhr vom 29.01.2014)

In Abbildung 25 sieht man das Tweet-Aufkommen des New Yorker Stadtteils Manhattan und den angrenzenden Stadtteilen von einer Stunde. Sehr gut ist hier die sehr hohe Aktivität im Stadtteil Manhattan erkennbar. In den angrenzenden Stadtteilen, ist im Vergleich viel weniger Aktivität zu beobachten. Dies bedeutet, dass das Tweet-Aufkommen je nach Region sehr stark variieren kann und dass der Übergang von einer Region zur nächsten Region ziemlich abrupt stattfinden kann.

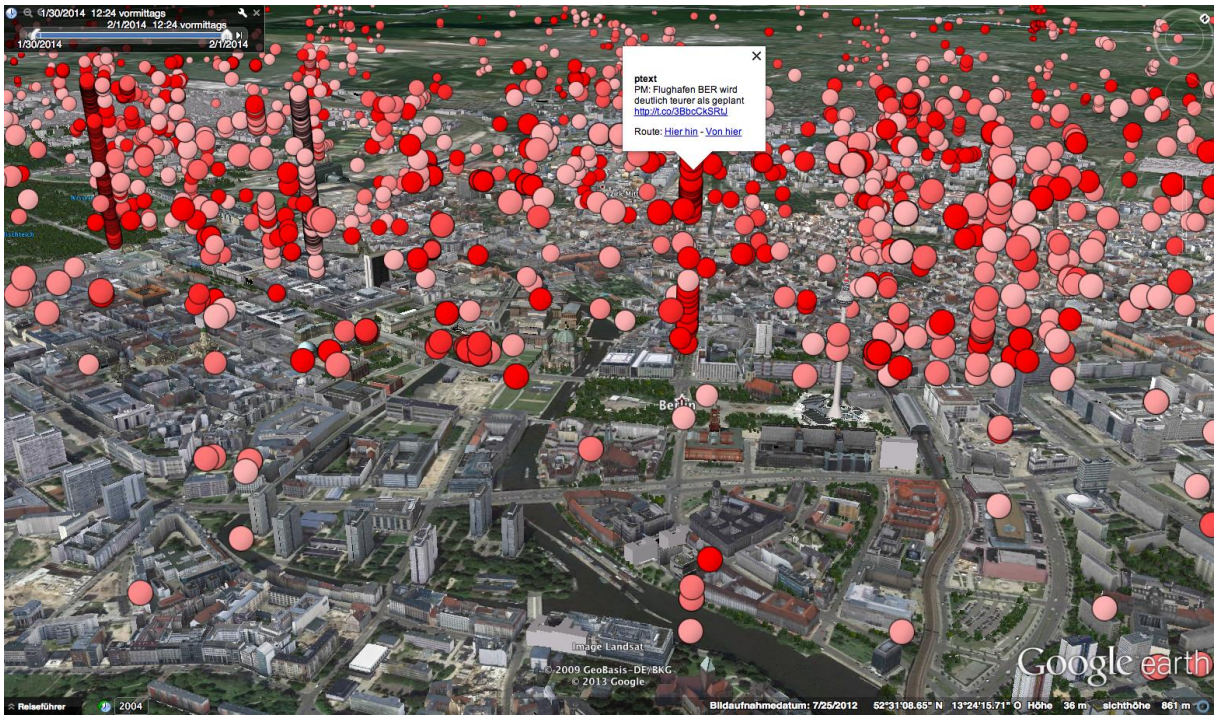

Abbildung 26: Visualisierte Tweets mit Geoinformationen mit Blick auf Berlin-Mitte (30.01.2014-01.02.2014)

In der Abbildung 26 sieht man das Tweet-Aufkommen des Stadtteils Berlin-Mitte von einem Tag. Deutlich ist zu erkennen, dass das Tweet-Aufkommen hier viel geringer ist als in Manhattan (Abbildung 25 zeigte nur das Tweet-Aufkommen von einer Stunde). Auch hier ist zu erkennen, dass an den angrenzenden Regionen das Tweet-Aufkommen merklich abnimmt, hier zu sehen an der nördlichen Stadtgrenze von Berlin am oberen Teil des Bildes. Weiterhin ist zu erkennen, dass es viele Tweets gibt, die scheinbar von derselben Position gesendet worden sind, zu erkennen an den senkrechten Säulen von Kugeln. Lässt man sich solche Tweets anzeigen, erkennt man das hier meist Systeme bestimmte Nachrichten senden und immer die gleiche Position verwenden in den Tweets. Im Beispiel zu sehen ist ein Tweet, welcher einen Pressetext beinhaltet.
Im letztem Beispiel in Abbildung 27 ist der Blick auf die Karte nun senkrecht von oben eingestellt so, dass man den Ort der Tweets genauer sehen kann. Zu sehen ist Deutschland mit den angrenzenden Nachbarstaaten mit den Tweet-Aufkommen von einer Stunde. Die regionalen Unterschiede in der Anzahl von Tweets mit Positionsangaben sind hier deutlich zu erkennen. Eine hohe Aktivität in den Ballungsgebieten wie z.B. Berlin, Hamburg, München und dem Ruhrgebiet, steht einer z.T. sehr niedrigen Aktivität in den restlichen Flächen von Deutschlands gegenüber. Allein im Bundesland Thüringen, wurden in der beobachteten Zeit, nur 4 Tweets mit Positionsangaben registriert. Ebenfalls erkennbar ist, dass der Unterschied des Tweet-Aufkommens nicht nur zwischen Regionen wie Stadtteilen oder zwischen Stadt und Umland besteht, sondern es gibt auch große Unterschiede zwischen den Ländern. Das Tweet-Aufkommen in den Beneluxstaaten, allen voran den Niederlanden, ist um einiges

höher. Die Niederlanden gehört zu den 10 aktivsten Twitter-Nationen [105]. Allein im Juni 2013 wurden in den Niederlanden allein 360 Millionen Tweets versandt [105]. Auch dort ist ein Unterschied in der Nutzung zwischen Stadt und Land erkennbar doch das gesamte Volumen an Tweets ist dort höher als in Deutschland. [107]

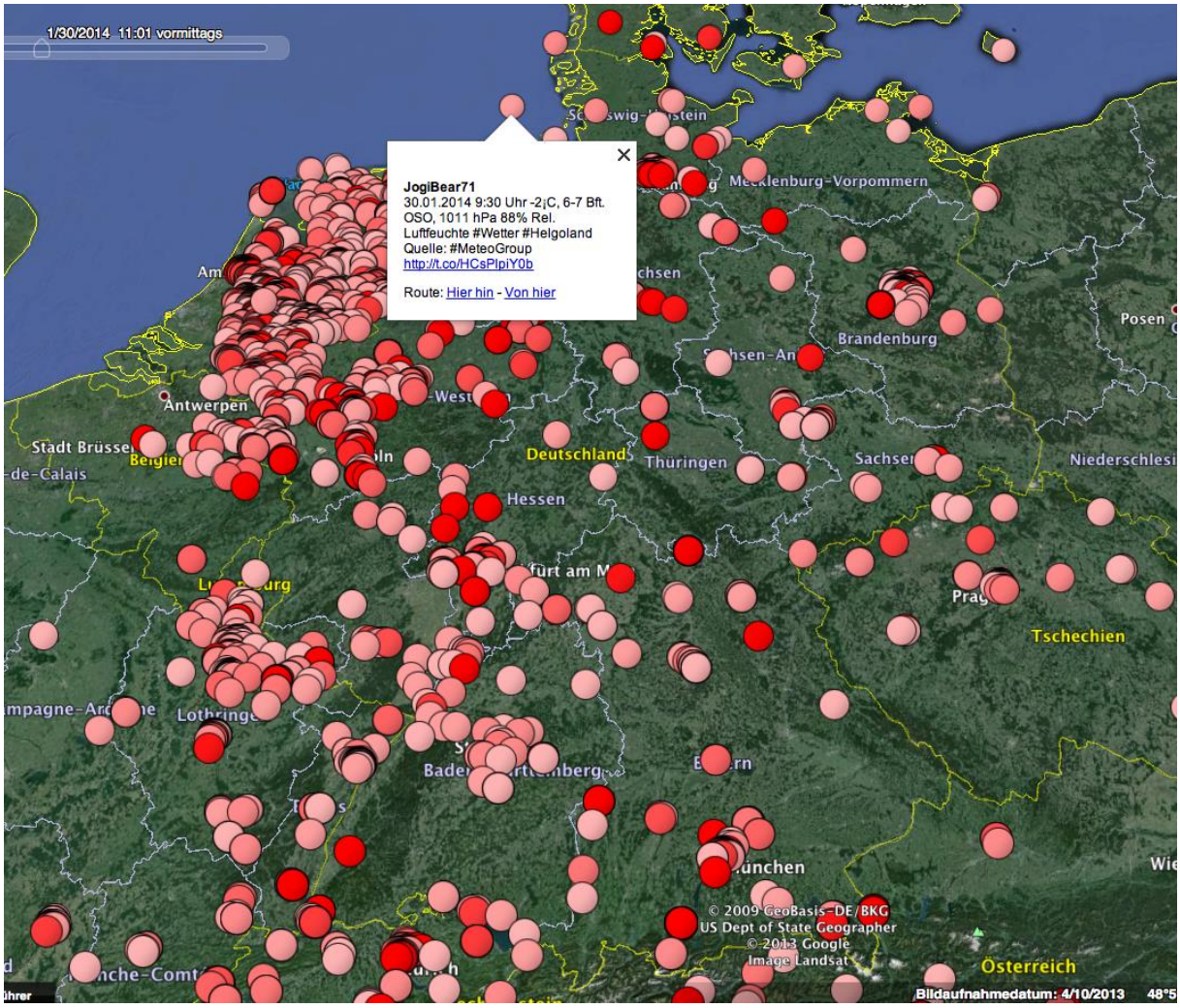

Abbildung 27: Visualisierte Tweets mit Geoinformationen von Deutschland und den angrenzenden Ländern (10 Uhr-11 Uhr 30.01.2014)

Das Tweet-Aufkommen an sich ist für eine spätere Analyse (Ereignisdetektion oder eine andere Analyse) insoweit von Bedeutung, dass mehr zur Verfügung stehende Daten in einer Region auch eine entsprechend bessere Analyse ermöglichen.
Eine Möglichkeit die Anzahl der zu analysierenden Tweets zu erhöhen, ist die Position anderweitig festzustellen. So gibt es Paper wie z.B. [108], welche die Position aus den Inhalt der Tweets rückschließen. Weiterhin besteht die Möglichkeit auch Tweets zur Analyse mit hinzuzufügen, die zwar Positionsdaten beinhalten aber nicht die exakte Position sondern nur einen ungefähren Ort. In diesen Metadaten findet man mehrere Koordinaten die eine Fläche beschreiben, aus der der Tweet gesendet worden ist. Es ist in den letzten Jahren aus den eigenen gesammelten Daten erkennbar, das solche Tweets mit ungenauen Ortsangaben signifikant mehr geworden sind. Momentan (Stand September 2015) stellen sie die Mehrzahl der georeferenzierten Tweets dar, die über die Streaming API empfangen werden können. Diese signifikante Erhöhung von Tweets deutet darauf hin, dass die Apps, mit denen die Tweets gesendet werden, nicht mehr standardmäßig die genauen Koordinaten verschicken. Dies ist z.B. sinnvoll, wenn man

nur mitteilen möchte, dass man sich z.B. in einer bestimmten Stadt befindet aber aus Gründen der Privatsphäre nicht den exakten Ort mitteilen möchte.
Da die Anzahl der Tweets mit exakter Koordinate so stark abgenommen haben, wurde das System erweitert, so dass auch diese ungenauen Orte mit ausgewertet werden können. Da es aber verschiedene Stufen der Ungenauigkeit gibt (die per Koordinaten beschriebene Box ist unterschiedlich groß) muss zuvor überprüft werden, ob die Ungenauigkeit nicht zu groß ist und somit die Koordinate nicht doch zu vage ist. Ist die Seitenlänge der aufgespannten Koordinatenbox größer als ein Breitengrad, so besteht schon eine Ungenauigkeit von mehr als 111km. Damit wäre die Ortsangabe nicht zu gebrauchen da sie zu ungenau ist. Beträgt die Ungenauigkeit dagegen nur wenige Kilometer, so kann man das Zentrum dieser Koordinatenbox errechnen und diese Koordinate zur Berechnung mit heranziehen.

3.3 Konzeption von Ereignisdetektionsalgorithmen

Die Grundlage der nun vorgestellten Ereigniserkennungsalgorithmen ist die Burst-basierte Ereigniserkennung, die in Kapitel 2.4.5 beschrieben wurde. Die Annahme war dort, dass bei einem Ereignis die Häufigkeit von bestimmten Termen im Nachrichtenstrom ansteigt. Beobachtet man also die eintreffenden Tweets und analysiert die Häufigkeit der einzelnen Terme, so kann man plötzliche Anstiege in der Häufigkeit der Nutzung von bestimmten Termen erkennen, die auf ein neues Ereignis hindeuten könnten. Die betrachteten Systeme in Kapitel 2.4.5 haben gezeigt, dass dieser Ansatz sich gut eignet, um Ereignisse zu detektieren. Der Vorteil z.B. gegenüber einer LDA-Analyse ist, dass die Anzahl der Ereignisse nicht vorher festgesetzt werden muss. In Abbildung 28 ist der Ablauf der Analyse stark vereinfacht in Pseudocode dargestellt. Die einzelnen Schritte werden detailliert erläutert.

```
create Referenzkorpus;                                              (a)
create Analysekorpus;                                               (b)
foreach ( Tweet in Analysekorpus ) {
        foreach ( Term in Tweet ) {
                errechne H_Referenzkorpus(Term_Ort);                (c)
                errechne H_Analysekorpus(Term_Ort);                 (d)
                Hochrechnung von H_Analysekorpus auf Referenzzeitraum;
                if(H_Analysekorpus - H_Referenzkorpus>Schwelle) {   (e)
                        Term_Ort in Ergebnismenge;
                }
        }
}
Gruppiere alle gefundenen Term_Ort zu Ereignissen;                  (f)
Reichere Ereignisse mit Original Tweets an welche                   (g)
Term_Ort aus der Ergebnismenge beinhalten;
```

Abbildung 28: Vereinfachter Ablauf der Analyse in Pseudocode

Was man für eine Burstanalyse benötigt ist eine Referenz, mit der man die Häufigkeit des Auftretens der Terme vergleichen kann. Für diese Zwecke nimmt man eine Menge

an Texten die als Referenz dient. Diese Menge an Beispieltexten nennt man Referenzkorpus. Aus diesem Referenzkorpus wird für jeden Term errechnet wie oft er auftritt. (Abbildung 28 (a)) Von den neu eintreffenden Tweets muss nun auch die Häufigkeit der Terme bestimmt werden und auf dem Referenzkorpus bezogen verglichen werden. Die Menge der zu analysierenden Tweets, die dem Referenzkorpus gegenüber gestellt werden, wird im weiteren Verlauf als Analysekorpus bezeichnet.
Diese Art der Analyse, der Vergleich eines Analysekorpus mit einem Referenzkorpus, nennt man Differenzanalyse.
Um etwas gegen einen Referenzkorpus vergleichen zu können, benötigt man eine Vergleichsmenge. Hier sind das die aktuellen Tweets z.B. der letzten Stunde einer bestimmten Region die man untersuchen möchte.
Diese Tweets sind Teil des Analysekorpus und sind aktuelle Tweets, z.B. der letzten Stunde (Abbildung 28 (b), Abbildung 29). Wie weit man hier den Bereich in die Vergangenheit zieht, kommt auf die Menge der Tweets an, die man aus einer Region erfasst. Liefert die Region eine große Anzahl von Tweets pro Minute, so kann man den zeitlichen Bereich für den Analysekorpus einschränken. Liefert eine Region dagegen sehr wenige Tweets, so muss man länger sammeln, damit man genügend Tweets hat, um daraus eine Aussage über die aktuelle Häufigkeit von Termen in den Tweets zu treffen.
Die Auswahl des Referenzkorpus ist dagegen schwieriger. Hier gibt es zum einen vorgefertigte Korpora, die man nutzen könnte, wie z.B. der Deutsche Referenzkorpus, welcher seit 1964 existiert und von dem Institut der deutschen Sprache in Mannheim gepflegt wird und die verschiedensten Texte aus Magazinen, Zeitungen und Bücher enthält [109] oder der Edinburg Twitter Korpus [62] der selbst aus 97 Millionen Tweets besteht und als Grundlage vieler wissenschaftlicher Arbeiten dient. Doch sind diese Korpora alle starr und zum Teil nicht einsetzbar, da sie in der Regel immer für eine bestimmte Sprache sind. Der Deutsche Referenzkorpus beinhaltet nur Texte deutscher Sprache und der Edinburg Korpus nur englische Tweets. Hier wäre also eine Filterung der einkommenden Tweets zwangsweise nötig, wenn man die Korpora einsetzen möchte, da die gesammelten Tweets aus einer Region aus einem Sprachmix bestehen bzw. auch Tweets von unterschiedlichen Regionen der Erde gesammelt werden. Doch eine Filterung der Eingangsdaten nach Sprache wiederspricht den gesetzten Zielen die Eingangsdaten ohne Filterung oder Nachbearbeitung zu analysieren.
Das Problem des fehlenden Referenzkorpus lässt sich aber in diesem Fall elegant lösen indem man eine große Menge an zeitlich zuvor gesammelten Tweets als Referenzkorpus nimmt. Damit löst sich auch das Problem der unterschiedlichen Sprachen auf, da die gleiche Sprachmischung auch im Referenzkorpus zu finden ist. Diese spezielle neue Art des Referenzkorpus, der auch noch weitere Vorteile bietet, wird ab jetzt adaptiver Referenzkorpus genannt.

3.3.1 Adaptiver Referenzkorpus

Ein normaler Referenzkorpus (wie in Kapitel 2.4.5 eingeführt wurde) dient bei einer Differenzanalyse als starre Referenz, um Abweichungen im Analysekorpus zu erkennen. Bei der aktuellen Problemstellung wäre diese Starrheit des Korpus aber hinderlich, was folgendes Gedankenexperiment zeigen soll.
Gegeben sei ein Referenzkorpus, der aus einer großen Anzahl von Tweets erzeugt wurde (z.B. der Edinburgh Korpus). Dieser Referenzkorpus wird genutzt, um die Häufigkeit der Terme zu bestimmen, um diese mit der Häufigkeit im Analysekorpus zu vergleichen. Nun beginnt ein großes Ereignis stattzufinden z.B. eine Fußballweltmeisterschaft (WM) und das Vorkommen von Termen in Bezug zu dieser WM steigt. Durch den Vergleich zum Referenzkorpus wird das gehäufte Vorkommen auch erkannt und man hat ein neues Ereignis entdeckt. Doch auch ein paar Tage später wird immer noch das Ereignis

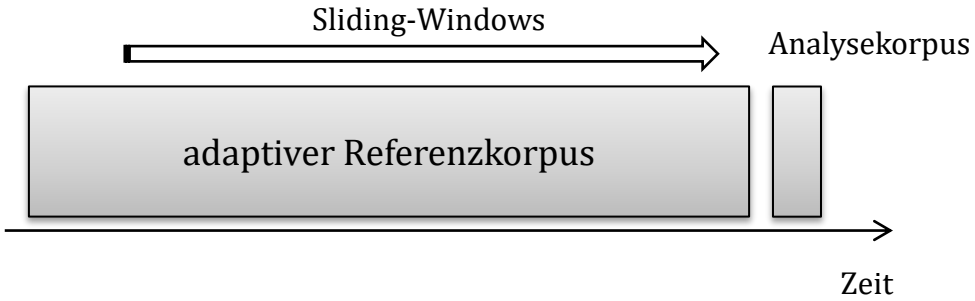

Abbildung 29: Adaptiver Referenzkorpus und Analysekorpus dargestellt auf einem Zeitstrahl

als neu erkannt. Auch andere Vorkommnisse, die sich einmal ändern und dann über längere Zeit oder dauerhaft bestehen bleiben werden mit dieser Methode ständig als ein neues Ereignis detektiert werden.
Um dies zu verhindern, muss sich der Referenzkorpus an die neue Situation dynamisch anpassen können. Dazu verschiebt sich im Sliding-Window-Verfahren der Zeitbereich, aus der die Tweets für den Referenzkorpus übernommen werden (siehe Abbildung 29).
D.h. man hat zum einem den Analysekorpus, der immer aktuelle Tweets enthält und sich somit ebenfalls per Sliding-Window Technik mit der Zeit bewegt, und nun auch den Referenzkorpus, der zwar größer ist, sich aber trotzdem mit der Zeit mit bewegt. So bekommt man immer einen aktuellen Referenzkorpus. Wenn man jetzt das vorherige Gedankenexperiment noch einmal wiederholt, bemerkt man, dass zum Anfang des Ereignisses „WM“ das Ereignis als neu erkannt wird, aber da das Ereignis länger andauert, wandern auch die WM-Tweets, mit den neuen Termhäufigkeiten, mit in den Referenzkorpus ein. Nach einer bestimmten Zeit (je nachdem wie breit die entsprechenden Korpora sind) ist das Ereignis „WM“, mit seinen abweichenden Termhäufigkeiten, kein neues Ereignis mehr und wird nicht mehr detektiert werden. Dabei ist es völlig egal, um welche Sprache es sich bei den Tweets handelt.

3.3.2 Ortsabhängige Termhäufigkeit

Wie bereits in der Einleitung zu Kapitel 3 erwähnt, soll die Ortsinformationen, die den gesammelten Tweets als Metadaten anhaften, für die Analyse direkt verwendet werden. Die zusätzliche Kenntnis des Ortes des Tweets dient dazu, die Ereigniserkennung zu verbessern, um den Ort der erkannten Ereignisse festzustellen. Die Kenntnis des Ortes eines erkannten Ereignisses lässt sich dazu gebrauchen, die erkannten Ereignisse für den Aufenhaltsort eines potentiellen Endnutzer einzugrenzen.
Betrachtet man die Terme in den Tweets im Zusammenhang mit dem Ort ihres Auftretens näher, so fällt auf, dass der Ort einen Einfluss auf die Termhäufigkeit hat, also eine ortsabhängige Termhäufigkeit existiert. Ein Gedankenexperiment hilft auch hierbei, diesen Sachverhalt besser zu verstehen. In Berlin am Brandenburger Tor wird vermutlich häufiger über das Brandenburger Tor getwittert als z.B. in Ilmenau auf dem Campus. Auf dem Campus in Ilmenau wird dagegen mehr z.B. über die Mensa getwittert als am Brandenburger Tor. Weil Nutzer am Brandenburger Tor sicherlich vermehrt mitteilen wollen, wo sie gerade sind (z.B. per Checkins) oder allgemein über den Ort berichten, auf dem Campus hingegen den Followers vielleicht mitteilen möchten, dass man gerade auf dem Weg zur Mensa ist.
Dies kann man auch konkret an realen Daten ablesen. Diesmal geht es um die Terme „Berlin“ und „Hamburg“ und ihrer jeweiligen Anzahl der Nutzung der beiden Terme in den jeweiligen Städten Berlin und Hamburg. Dabei wurden alle Tweets mit Geoinformation aus einem definierten geographischen Bereich und einem bestimmten Zeitraum untersucht, die einen der beiden Terme beinhalten. Wie bereits vermutet

kommt der Term „Berlin“ in den Tweets aus Berlin häufiger vor als in Tweets aus Hamburg. Und umgekehrt genauso. Der Term „Hamburg“ kommt in Hamburg häufiger vor als in Berlin. Die nachfolgende Tabelle zeigt das konkrete Vorkommen der Terme „Berlin“ und „Hamburg“ in den jeweiligen Städten. Als Beispielzeitraum ist hier der Monat August 2012 gewählt worden.

	Tweets mit Term „Berlin“	**Tweets mit Term „Hamburg“**	**Gesamtanzahl der Tweets im Zeitraum**
Berlin *lat: 52,322-52,659* *lon: 13,026-13,851*	15 079	104	78 424
Hamburg *lat: 53,426-53,670* *lon: 9,74-10,245*	155	4 272	21 161

Tabelle 1: Anzahl der Terme "Berlin" und "Hamburg" an den jeweiligen Orten im Zeitraum August 2012

In diesem Zeitraum wurde der Term „Berlin“ in Berlin ca. 145x häufiger getwittert als der Term „Hamburg“. Ebenso umgekehrt wurde der Term „Hamburg“ ca. 28x mehr getwittert in Hamburg als der Term „Berlin“. Die Erkenntnis aus dem Gedankenexperiment konnte auch mit realen Daten erfolgreich verifiziert werden. Es ist somit deutlich zu erkennen, dass der Ort einen Einfluss auf die Termhäufigkeit besitzt.
Die Erkenntnis, dass die Termhäufigkeiten abhängig von dem Verfassungsort des Tweets sind, ist für die Differenzanalyse von Bedeutung, da hier Veränderungen von Termhäufigkeiten untersucht werden. Um die entsprechenden Korpora für die Differenzanlayse zu berechnen, muss somit mit den ortsabhängigen Termhäufigkeiten gerechnet werden. Im konkreten Fall würde dies theoretisch bedeuten, dass man für jeden Ort einen speziellen Referenz- und Analysekorpus benötigt, in welchem nur Tweets des entsprechenden Ortes enthalten sein dürfen. Andernfalls würde das Ergebnis, durch die ortsabhängigen Termhäufigkeiten, bei der Differenzanalyse verfälscht werden.
Ein weiteres Beispiel soll diesen Effekt verdeutlichen. Angenommen in einem Monat würde der Term „Berlin“ in Berlin 15 179 Mal in Tweets auftauchen und in Hamburg 204-mal. Beide mal also 100 Vorkommnisse mehr als im obigen Beispiel. Für den Ort Berlin wäre, dass eine Steigerung der Häufigkeit, gegenüber den obigen ausgesuchten Monat, des Terms von gerade einmal 0,6%, also vernachlässigbar. Für den Ort Hamburg bedeutet dies aber eine Steigerung von 96%. D.h. die Betrachtung des Ortes, ist für die Genauigkeit der lokalen Differenzanalyse entscheidend, um zu beurteilen, ob ein Term signifikant häufiger auftritt als im vergleichbaren Referenzzeitraum.

3.3.3 Ortsabhängige Korpora

Um die Korpora zu errechnen, ist es nötig die ortsabhängige Termhäufigkeit für einen konkreten Ort und für einen konkreten Term zu berechnen (Abbildung 28 (c/d)). Die ortsabhängige Termhäufigkeit ist die Summe des Auftretens des Terms an diesem Ort in einen bestimmten Zeitraum. Da Tweets von verschiedenen Sendern i.d.R. nie exakt die gleiche Koordinate besitzen (wenn die Position z.B. exakt per GPS bestimmt wird), ist die Berechnung nicht trivial. Existieren zwei Tweets, die exakt dieselbe Position besitzen würden, so addiert sich die Häufigkeit ganz normal, da der Ort der gleiche ist. Ist der zweite Tweet sehr weit weg (z.B. mehrere hundert Kilometer), so sollte das Auftreten des Terms dort keinen Einfluss auf die ortsabhängige Termhäufigkeit mehr haben. D.h. dieser Tweet würde mit 0 gewichtet, wenn im Tweet vor Ort die ortsabhängige Termhäufigkeiten errechnet werden würde. Die Tweets, genauer die Termhäufigkeiten

der Terme in dem Tweet, müssen also in Abhängigkeit ihrer Positionen zum analysierenden Tweet gewichtet werden. Nachbar-Tweets im Umkreis, bis zu einem gewissen Radius, haben also einen Einfluss auf die Berechnung der Termhäufigkeit vor Ort.
Der Abstand ab dem ein Tweet keinen Einfluss auf die Berechnung der ortsabhängigen Termhäufigkeit mehr haben sollte, wird vom Autor Effektradius genannt. Tweets mit einem Abstand kleiner als der Effektradius, haben einen Einfluss/Wirkung/Effekt auf die Berechnung der ortsabhängigen Termhäufigkeit. Den Einfluss der Tweets in Abhängigkeit zu deren Abstand zum zu analysierenden Tweet, wird mittels einer Gewichtungsfunktion $f(x)$ abgebildet, wobei x der Abstand der Tweets ist. Die Funktion muss an der Stelle $f(0) = 1$ sein und sollte ab $f(x_e) = 0$ sein, wobei x_e der Effektradius ist. Zwischen diesen zwei Punkten könnte man die Gewichtung mittels einer linearen Funktion abbilden. Doch bietet sich hier eine andere Klasse von Funktionen an, die die gewünschte Gewichtung besser beschreibt. Wenn der Abstand der Tweets zueinander relativ gering ist, so sollte die Gewichtungsfunktion sich nicht so stark absenken wie bei einer linearen Funktion. Dies ist durch folgende Idee begründet, dass Tweets, die relativ nah vom eigenen Ort gesendet werden, rel. hoch gewichtet werden sollten. Ein kleines Beispiel macht das deutlich. Angenommen ein Nutzer sendet einen Tweet auf dem Campus der Universität Ilmenau. Etwas weiter weg wird ebenfalls ein Tweet versendet. Der Sender befindet sich auch auf dem Campus. D.h. bei kurzen Distanzen könnte man die Sendeorte als einen Standort betrachten. Wenn ein anderer Tweet nun von weiter weg gesendet wird, aber noch in der gleichen Stadt, so ist das für den Nutzer auch noch sehr interessant, da es die gleiche Stadt ist, in der er lebt, aber nicht so relevant als wenn es vom Campus käme, wenn wir bei dem obigen Beispiel bleiben. Danach kann die Gewichtung schneller sinken wenn die Entfernungen immer größer werden. Funktionen, die den beschriebenen Gewichtungsverlauf sehr gut abbilden, sind z.B. die Exponentialfunktionen. Somit wird für die Gewichtung der Terme eine einfache Exponentialfunktion verwendet.

$$f(x) = e^{-x^2}$$

Formel 6: Exponentialfunktion zur Gewichtung der Terme

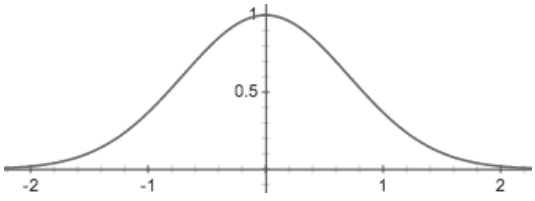

Abbildung 30 Plot der Exponentialfunktion

In Abbildung 30 ist ein Beispielplot dieser Funktion zu sehen. Die einzige Eigenschaft, die die Exponentialfunktion nicht erfüllt ist, dass sie nicht $f(x_e) = 0$ wird. Doch für die nachfolgende Anwendung ist es ausreichend, wenn die Funktion fast Null wird d.h. die Gewichtung so klein wird das man sie ignorieren kann. Für diese Stelle wurde x=2 ausgewählt. Hier beträgt der Wert der Exponentialfunktion ca. 0,02 und ist für die nachfolgenden Berechnungen vernachlässigbar. D.h. Tweets die weiter entfernt sind als

der Effektradius, gehen in die Berechnung der ortsabhängigen Termhäufigkeit in einem so geringen Maße ein, dass es ignoriert werden kann.
Für die Berechnung im Raum benötigt man eine dreidimensionale Version der Exponentialfunktion.

$$f(x,y) = e^{-(x^2+y^2)}$$

Formel 7: Räumliche Exponentialfunktion

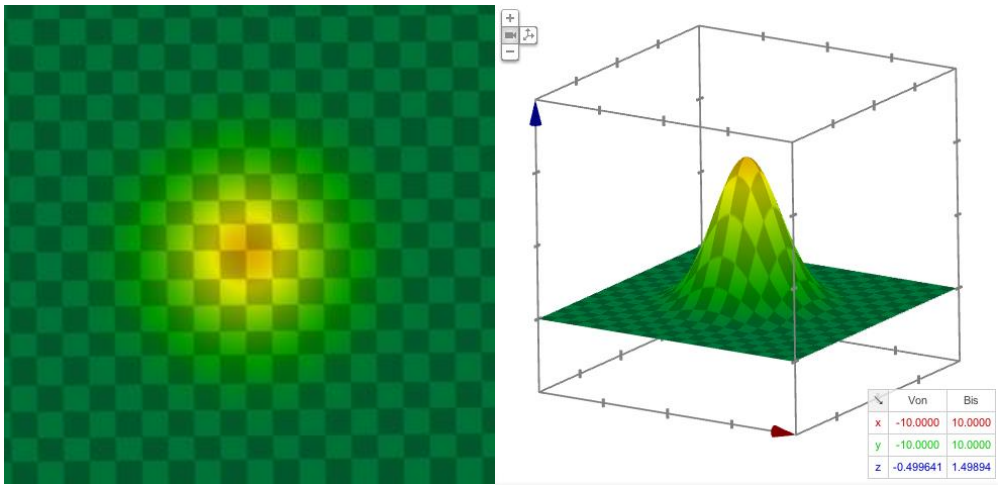

Abbildung 31: Plot der dreidimensionalen Exponentialfunktion

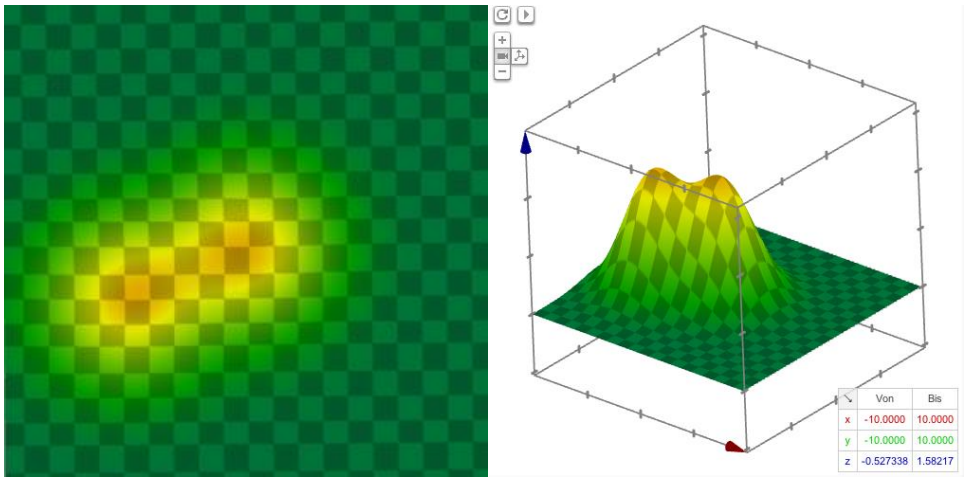

Abbildung 32: Gewichtungsfaktoren zweier Tweets die nah beieinander sind addieren sich auf

Die Abbildung 31 zeigt den Plot der Gewichtungsfunktion für einen Tweet. Abbildung 32 stellt dar, wie sich die Gewichtungsfunktionen aufaddieren, wenn es in der Nähe noch einen Tweet gibt.
Je nachdem wie groß der Effektradius gewählt wird, wird die Kurve der Funktion entweder gestaucht oder gestreckt werden. Die Auswahl des Effektradius nimmt großen Einfluss auf die späteren Schritte der Differenzanalyse, da dadurch die Auswahl der Tweets für die entsprechenden Korpora gesteuert wird.
Da die Berechnungen für reale Orte auf der Erde stattfinden, muss dies bei der späteren Anwendung mit beachtet werden und die Exponentialfunktion mit Korrekturfaktoren erweitert werden, um eine Verformung der Glockenkurve in West-Ost Richtung, aufgrund der Erdkrümmung, zu verhindern.

3.3.4 Differenzanalyse der ortsabhängigen Termhäufigkeiten

Um die Ereignisse in einem bestimmten Gebiet detektieren zu können, müssen die zuvor besprochenen Ideen und Teilalgorithmen nun zu einem Ereignisdetektionsalgorithmus zusammengefügt werden.

3.3.4.1 Erzeugung der Korpora

Als ersten Schritt müssen die, für die Analyse benötigten, Korpora mit Tweets gefüllt werden (Abbildung 28 (a und b)). Dazu werden Tweets aus der zu analysierenden Region ausgewählt. Für den Referenzkorpus wären dies z.B. Tweets der letzten Tage. Dabei ist zu beachten, dass eine genügend große Anzahl von Tweets für den Referenzkorpus zur Verfügung steht, um verlässliche (stabile) ortsabhängige Termhäufigkeiten errechnen zu können. Je nach Tweet-Aufkommen, in der zu analysierenden Region, kann dieser Zeitraum von wenigen Tagen, z.B. im Gebiet New York mit einer sehr hohen Twitter-Nutzung, bis zu einem Monat, z.B. in Deutschland, einen Gebiet mit einer niedrigen Twitter-Nutzung, betragen. Bei praktischen Versuchen hat sich gezeigt, dass ein Referenzkorpus mit ca. ab einer Mio. Tweets eine gute Ausgangsbasis darstellt. Je größer die zu untersuchende Region, umso größer muss auch der Referenzkorpus sein damit für jeden Ort verlässliche Termhäufigkeiten errechnet werden können. Ist der Referenzkorpus zu klein, so stehen auch pro Term nur eine geringe Anzahl von Vorkommen zur Verfügung und es kann keine stabile Termhäufigkeit errechnet werden. Dies führt bei der Differenzanalyse dazu das vermeintliche Ereignisse detektiert werden die aber keine sind (Anstieg der Fehlalarme).

Im zweiten Schritt müssen die Tweets ausgewählt werden, die im Analysekorpus enthalten sein sollen. Also die Tweets, die anschließend untersucht werden sollen. Da neue Ereignisse detektiert werden sollen, braucht man die aktuellsten Tweets aus der zu untersuchenden Region. Je nach Tweet-Aufkommen können das die Tweets der letzten Stunde oder eines kürzeren Zeitraumes sein. Je mehr Tweets man für den Analysekorpus benutzt, umso besser lassen sich im Anschluss die ortsabhängigen Termhäufigkeiten errechnen, da genügend Daten zu Grunde liegen. Nimmt man zu wenig Tweets so kann es sein, dass der zu untersuchende Term nur wenige Male im Korpus enthalten ist und man daher keine sinnvollen Aussagen über veränderte Termhäufigkeiten im Analysekorpus machen kann. Zudem besteht die Gefahr, dass man bei zu wenigen Daten fälschlicherweise Termhäufigkeitsschwankungen erkennt und es somit zu einer Fehldetektion von Ereignissen kommt (dazu mehr im Kapitel 3.3.4.4). Nimmt man aber zu viele Tweets für den Analysekorpus, so ist der Korpus zeitlich zu „breit“ und man erkennt neue Ereignisse erst später, da die neuen geänderten ortsabhängigen Termhäufigkeiten sich erst im Analysekorpus ausbreiten müssen.

3.3.4.2 Berechnung der ortsabhängigen Termhäufigkeiten

Nachdem die Tweets ausgewählt wurden, werden alle Tweets im Analysekorpus einzeln betrachtet und jeder Term in dem Tweet extrahiert. Anzumerken ist hier, dass keine Vorverarbeitung der Daten, also der Tweets bzw. der Terme, stattfindet! Aus Performancegründen könnten an dieser Stelle etwaige Stoppwörter entfernt werden (dieser optionale Verarbeitungsschritt wäre dann aber sprachabhängig durchzuführen), doch ist dies für den nachfolgenden Erkennungsalgorithmus nicht notwendig. Das heißt auch, dass die Sprache der Terme somit egal ist, da keine sprachspezifischen Bearbeitungsschritte ausgeführt werden müssen. Für jeden extrahierten Term, aus den Tweets des Analysekorpus, wird nun die ortsabhängige Termhäufigkeit in den jeweiligen Korpora (Analyse- und Referenzkorpus) nach folgender Formel errechnet.

$$H_{term}(x,y) = \sum e^{-\left(\left(\frac{111*(x_n-x)}{effectRadius*0{,}5}\right)^2+\left(\frac{\frac{12742*\pi*\cos(y_n)}{360}*(y_n-y)}{effectRadius*0{,}5}\right)^2\right)}$$

Formel 8: Formel zur Berechnung der ortsabhängigen Termhäufigkeit an der Stelle x (Breitengrad), y (Längengrad) mit den Korrekturfaktoren zum Ausgleich der Erdkrümmung

Dabei wird die ortsabhängige Termhäufigkeit H_{term} für diesen Term, an genau der Position (x, y) des gerade untersuchenden Tweets, für die zwei Korpora errechnet (Abbildung 28 (c und d)). Wird die ortsabhängige Termkonzentration im Analysekorpus z.B. errechnet, dann fließen alle einzelnen Orte ein, wo der jeweilige Term im Analysekorpus aufgetreten ist (x_n, y_n). Je nach Ort des Tweets werden die umliegenden Tweets durch diese Berechnung unterschiedlich gewichtet und nur die relevanten Tweets, die innerhalb des Effektradius liegen, fließen in die Berechnung ein. Sind die Werte für den Term errechnet, so wird der nächste Term in dem gerade analysierten Tweet vorgenommen bzw. danach der nächste Tweet bis alle Tweets im Analysekorpus untersucht wurden.

3.3.4.3 Untersuchung der errechneten ortsabhängigen Termhäufigkeiten

Nachdem man die ortsabhängigen Termhäufigkeiten für einen konkreten Term und einem konkreten Ort errechnet hat, muss man die $H_{term\_analyse}$, welche man aus dem Analysekorpus errechnet hat, hochrechnen auf den Referenzkorpuszeitraum (Abbildung 28 (e)), um diese Werte direkt miteinander vergleichen zu können.

$$H_{term\_hoch} = H_{term\_analyse} * \frac{t_{referenz}}{t_{analyse}}$$

Formel 9: Hochrechnung der ortsabhängigen Termhäufigkeit auf den Referenzkorpuszeitraum

Die errechnete ortsabhängige Termhäufigkeit aus dem Analysekorpus wird dabei mit dem Verhältnis zwischen dem Analysekorpuszeitraum ($t_{analyse}$) und dem Referenzkorpuszeitraum ($t_{referenz}$) multipliziert. Dadurch wird der hochgerechnete Wert mit der ortsabhängigen Termhäufigkeit, welcher aus dem Referenzkorpus errechnet wurde, direkt vergleichbar.
Ist der hochgerechnete Wert $H_{term\_hoch} > H_{term\_referenz}$ dann tritt der Term häufiger auf als im Referenzkorpus. Um den Wert in den nächsten Schritten besser handhaben zu können, ist es hilfreich, den Wert zu normieren indem man den hochgerechneten Wert H_{term_hoch} zu dem Wert aus dem Referenzkorpus ins Verhältnis setzt.

$$a_{term} = \frac{H_{term\_hoch}}{H_{term\_referenz}}$$

Formel 10: Verhältnis zwischen der hochgerechneten ortsabhängigen Termhäufigkeit und der errechneten ortsabhängige Termhäufigkeit aus dem Referenzkorpus

Ist $a_{term} > 1$, so tritt der Term im Analysekorpus hochgerechnet häufiger auf als im Referenzkorpus (Abbildung 28 (e)). D.h. die ortsabhängige Termhäufigkeit hat sich verändert und der Term wird häufiger genutzt. Dies kann auf ein neues Ereignis hindeuten. Um Falschdetektionen zu vermeiden, ist es wichtig eine Schwelle (a_{min}) zu definieren, ab der man die Änderung der ortsabhängigen Termhäufigkeit als relevant ansieht. Setzt man die Schwelle hier zu niedrig, so gibt es häufig Fehlalarme, da die Termhäufigkeit ganz natürlich im Analysekorpus schwankt, da schon kleine Änderungen

im Vorkommen des Terms und die anschließende Hochrechnung auf den Referenzkorpuszeitraum den a_{term} um den Wert 1 schwanken lassen. In praktischen Versuchen hat sich gezeigt, dass eine Schwelle von 6,5 diese kleinen Schwankungen verlässlich herausfiltern kann. War die Schwelle kleiner so kam es vermehrt zu Fehlalarmen. Nimmt man bei der Schwelle dagegen einen zu großem Wert, so könnten potentielle Ereignisse dadurch verschluckt werden oder aber die Erkennung von neuen Ereignissen verzögert werden. Die Schwelle von 6,5 ist somit eine reine Abwägung wie viele Ergebnisse man haben möchte bzw. wie sensibel das System reagieren soll. Diese Schwelle ist ein wichtiger Parameter mit dem man im System einstellen kann, wie viele Ereignisse man detektiert bekommen möchte. Möchte man ein sehr empfindliches System, welches schon auf kleinste Ereignisse reagiert, aber auch eine höhere Fehlalarmrate hat, so wählt man eine kleinere Schwelle. Möchte man dagegen ein System, welches nur große Ereignisse detektieren soll, so wählt man eine größere Schwelle und das System reagiert erst bei einer größeren ortsabhängigen Termhäufigkeitsänderung. Daher gibt es für die Schwelle keinen „richtigen" Wert sondern der Wert ist abhängig von dem Anwendungsfall für welchen man das System einsetzen möchte. Möchte man ein System haben was sehr früh auf potentielle Ereignisse reagiert aber auch dadurch mehr Fehldetektionen erzeugt, so nimmt man eine kleinere Schwelle z.B. 2. Möchte man dagegen ein System was möglichst keine Fehlalarme produziert und nur auf signifikante Ereignisse reagieren soll, so nimmt man einen größeren Schwellwert z.B. 9 oder höher.

3.3.4.4 Selten vorkommende Terme und Sonderfälle

Zusätzlich zur gerade beschriebenen Schwelle ist es notwendig eine weitere Schwelle einzuführen, die schwach auftretende Terme am jeweiligen Ort herausfiltern soll. Diese selten vorkommenden Terme führen bei der beschriebenen Differenzanalyse zu Fehlalarmen, was folgendes Gedankenexperiment verdeutlichen soll. Gegeben sei ein Referenzkorpus, der eine zeitliche Breite von 30 Tagen hat und ein Analysekorpus, welcher nur Tweets der letzten Stunde enthält. Weiterhin existiert ein Term x, welcher im Referenzkorpus dreimal vorkommt und im Analysekorpus einmal vorkommt. Wenn man allein die Hochrechnung auf dem Referenzkorpuszeitraum vornimmt (zur Vereinfachung ohne Beachtung des Ortes in diesem Beispiel), so erkennt man, dass der Term hochgerechnet 720 mal im Referenzkorpuszeitraum vorkommen würde, was 240 mal mehr wäre als zuvor beobachtet. Dies bedeutet, dass selten auftretende Terme sich nur sehr schlecht zur Differenzanalyse eignen, da ihr Auftreten zu selten ist und es somit zu großen Fehlern im Hochrechnungsschritt kommt. Daher muss es eine ortsabhängige Mindesttermhäufigkeit geben (H_{min}). Die errechnete ortsabhängige Termhäufigkeit aus dem Analysekorpus muss hier größer als eine Mindestschwelle sein. In praktischen Experimenten hat sich hier gezeigt, dass schon eine niedrige Schwelle von drei hier ausreichend ist. D.h. um einen Term zu detektieren, welcher auf ein potentielles neues Ereignis hindeutet, muss folgendes gegeben sein.

$$H_{term\_analyse} > H_{min}$$

$$a_{term} > a_{min}$$

Formel 11: Finale Überprüfung der errechneten Werte mit zuvor definierten Schwellwerten

Sind diese zwei Bedingungen erfüllt, so kann der Term und ggf. der Quell-Tweet als Analyseergebnis ausgegeben werden.

Ein Sonderfall bei der Betrachtung der Terme tritt auf, wenn neue Terme auftreten die zuvor noch nicht im Referenzkorpus aufgetreten sind. In der Annahme, dass der

Referenzkorpus groß genug gewählt wurde, können diese neuen Terme ebenfalls auf neue Ereignisse verweisen, da der Term z.B. ein neues Tag für ein gerade stattfindendes neues Ereignis sein könnte (z.B. eine gerade stattfindende Veranstaltung oder ein anderes Ereignis mit einem speziellem Themen- / Ereignistag). Da der Term nicht im Referenzkorpus auftritt, lässt sich eine Differenzanalyse nicht durchführen. In diesem Fall darf der Term nicht unterschlagen werden, da er höchstwahrscheinlich ein neues Ereignis beschreibt. Es sollte nur die Mindesttermhäufigkeit überprüft werden.

3.3.4.5 Nächste Analyseiteration

Nachdem alle Terme in allen Tweets im Analysekorpus analysiert wurden, ist die aktuelle Analyseiteration abgeschlossen und alle relevanten Terme, welche eine deutlich höhere ortsabhängige Termhäufigkeit aufwiesen, extrahiert. Zum Schluss existiert somit eine Liste mit Termen und deren Orte, an dem die ortsabhängige Termhäufigkeit signifikant erhöht ist gegenüber den Werten aus dem Referenzkorpus. Dies kann an den jeweiligen Orten auf ein gerade stattfindendes Ereignis hindeuten. Durch das hinzuziehen der entsprechenden Quell-Tweets sollte sich nun im Anschluss das entsprechende Ereignis leicht erkennen lassen (Abbildung 28 (g)).
Ist die Analyseiteration beendet, kann sofort oder nach einen festgelegten Zeitintervall eine neue Analyseiteration mit neuen aktuellen Tweets gestartet werden.

3.3.5 Aufbereitung der Ergebnisse

Als Ergebnis der vorangegangenen Differenzanalyse hat man nun eine Liste von Termen. Die Ergebnisterme haben entweder eine ortsabhängige Termhäufigkeit im Analysekorpus die signifikant höher war im Vergleich zum Referenzkorpus bzw. es ist ein neu aufgetretener Term mit einer gewissen hohen ortsabhängigen Termhäufigkeit, der aber noch nicht im Referenzkorpus zu finden war. Zu den Termen hat man den Quell-Tweet, aus dem der Term war, und somit auch den genauen Sendeort des Tweets. Um den Nutzwert der Ereignisdetektion zu erhöhen, ist es angebracht die Ergebnisse weiter aufzubereiten, indem man z.B. die detektierten Terme in Ereignisse separiert (Abbildung 28 (f)) oder die Großereignisse getrennt betrachtet. Es ist auch möglich die Ergebnisse der Analyse mit Beispiel-Tweets oder extrahierten Medien (z.B. Bildern) anzureichern, die das Ereignis näher beschreiben (siehe Kapitel 4.5) (Abbildung 28 (g)).

3.3.5.1 Segmentierung der Ergebnisse in einzelne Ereignis-Cluster

Eine erste Aufbereitung der Ergebnisse besteht darin, die gefundenen Terme und Tweets zu Ereignissen zuzuordnen (Abbildung 28 (f)). Meist finden in einer Region zur gleichen Zeit mehrere Ereignisse parallel statt. Die Terme und Tweets beschreiben also mitunter mehrere Ereignisse, die gerade stattfinden. Ziel ist es die Terme so zu trennen, dass die einzelnen Ereignisse besser sichtbar werden. D.h. die Terme mit den Tweets müssen nach Ereignissen sortiert bzw. gruppiert werden. Ein möglicher Ansatz für diese Gruppierung wäre, dass man die Terme in Gruppen sortiert, welche auch zusammen in einem Tweet auftauchten. Dieser Schritt kann die Ereignisse schon sehr gut trennen doch produziert es mitunter zu große Ereignis-Cluster da bestimmte Terme auch in mehreren Tweets aus unterschiedlichen Ereignissen auftauchen könnten.

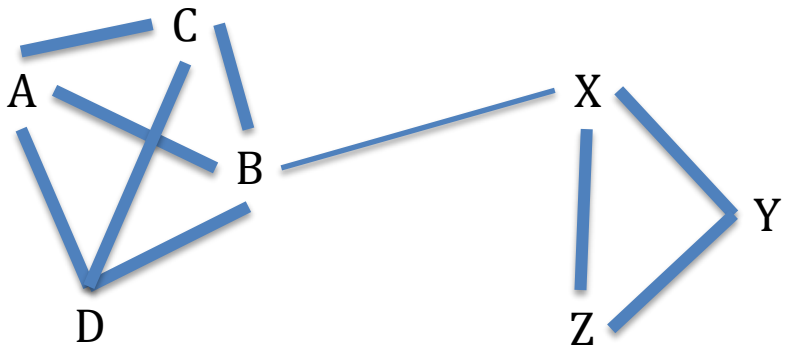

Abbildung 33: Beispiel für die Zuordnung von Ergebnistermen (A-D und X-Z) zu Ereignis-Cluster und dabei auftretende Probleme durch die Verknüpfung von zwei Ereignissen zu einem

In Abbildung 33 ist die beschriebene Situation dargestellt. Die Buchstaben von A-D und X-Z stellen die Ergebnisterme der Differenzanalyse dar. Je häufiger die Terme gemeinsam in Tweets zusammenauftauchen, umso dicker ist die Verbindungslinie zwischen den Termen. Man kann hier deutlich sehen, dass die Ergebnisterme wohl zwei unterschiedliche Ereignisse beschreiben. Trotzdem gab es, in diesem Beispiel, einen Tweet, wo der Term B und X gemeinsam auftrat. Durch diese Verbindung werden die zwei unterschiedlichen Ereignisse als ein Ereignis detektiert, wenn man so vorgeht, wie beschrieben. Um dies zu verhindern, sind aufwendigere Methoden notwendig, um die Terme mit den Quell-Tweets aufzuteilen. So wäre es denkbar, das gebildete Netzwerk durch gezielte Ausdünnungen in disjunkte Ereignis-Cluster zu zerteilen (Entfernung der Verbindung zwischen B und X). Als Beispiel für solch ein Vorgehen sei auf das zuvor erwähnte Paper [52] in Kapitel 2 verwiesen, indem ein vergleichbares Netzwerk in Segmente aufgeteilt wurde.
Nach der Segmentierung können die Terme und die Tweets zu Ereignissen gruppiert dem Nutzer geeignet dargestellt werden.

3.3.5.2 Detektion von Großereignissen

Die Detektion von Termen mit erhöhter Termkonzentration in einer bestimmten Region ist, je nach Zeit, normal. Es findet meist immer irgendwo ein Kongress oder ein anderes Ereignis statt, welches die Analyse erkennt. Je nach Region kann es auch normal sein, dass mehrere Ereignisse parallel stattfinden, z.B. am Wochenende stattfindende Sportereignisse in Deutschland. Doch lässt sich durch eine Betrachtung der Ergebnismenge auch ablesen, ob ein signifikantes Ereignis, z.B. ein gerade stattfindendes katastrophales Ereignis, ähnlich einer Naturkatastrophe oder ein von Menschenhand hervorgerufener Amoklauf, stattfindet.

Abbildung 34: Beispiel einer versendeten E-Mail (mit zusätzlichen Debugging-Daten angereichert), welche auf ein signifikantes Ereignis an der Ostküste der USA hinweist

In diesen Fällen werden mehr Terme detektiert als sonst, da über solch ein Ereignis auch mehr und umfassender getwittert wird (siehe Abbildung 34). Eine feste Ergebnismenge lässt sich aber hier nicht als Schwelle einsetzen, um solch ein Großereignis zu detektieren. Dazu schwankt die Anzahl von Ergebnistermen je Region und Zeit zu stark. In praktischen Versuchen hat sich bewährt, wenn man die durchschnittliche detektierte Termanzahl von gefundenen Termen, für die aktuelle Stunde und der Region aus den Daten der letzten 4 Wochen ermittelt. Die aktuelle Termanzahl wird dann nun mit dieser errechneten Anzahl verglichen (d.h. Vergleich mit dem Durchschnittswert der letzten 4 Wochen für die konkrete Region und die konkrete Stunde). Findet man bei der Analyse z.B. dreimal mehr Terme, als man es erwartet, so könnte dies auf ein signifikantes Ereignis hindeuten. Diese Schwelle kann natürlich wieder individuell je nach Anwendungsfall gewählt werden. Hat man nun solch ein signifikantes Ereignis detektiert, so kann man ggf. den Nutzer über zusätzliche Kommunikationskanäle darüber unterrichten, z.B. per E-Mail oder über eine Twitter Direktnachricht.

3.3.6 Einfluss des Effektradius auf die Analyse

Die Größe des Effektradius hat maßgeblichen Einfluss auf die Ergebnismenge und auf die Charakteristik der Ereigniserkennung. Mit ihm wird die Menge an Tweets gesteuert, die in die Berechnung einfließen sollen, indem man die Breite der Glockenkurve der Exponentialfunktion darüber verändert. Je kleiner der Effektradius ist, umso besser lassen sich die Ereignisse geographisch auflösen. Mit einem kleinen Effektradius lassen sich so z.B. getrennt Ereignisse in Stadtteilen lokalisieren wie z.B. ein Sportereignis im Stadion der Stadt und ein weiteres Ereignis in einem anderen Stadtteil. Nimmt man dagegen einen größeren Effektradius, so würde man die 2 Ereignisse auch noch detektieren aber sie könnten räumlich nicht mehr richtig separiert werden, da die

Glockenkurven breiter sind und sich mehr mit benachbarten Kurven aufsummieren. Um die Auswirkungen des Effektradius deutlicher zu machen, wurden die ortsabhängigen Termhäufigkeiten visualisiert. Die Darstellung wird von dem Autor „Termkonzentrationskarte“ genannt.

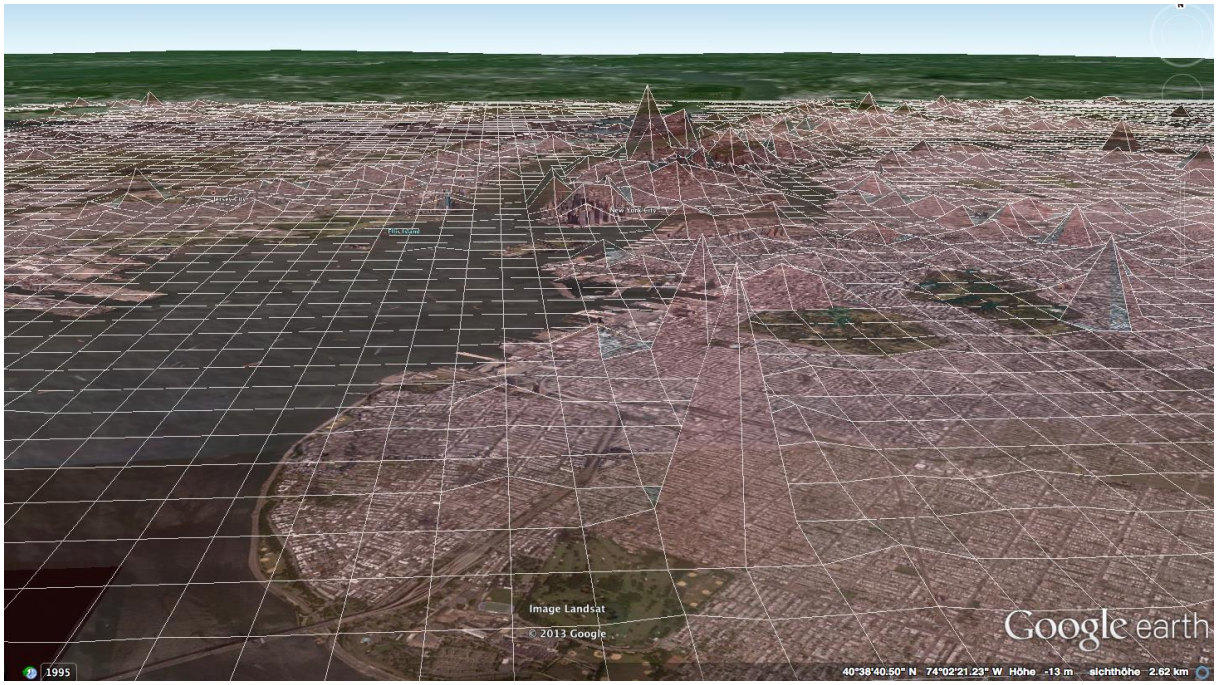

Abbildung 35: Termkonzentrationskarte für den Term "morning" im Referenzkorpus mit den Tweets vom 7.3.-19.03.2014 mit einem Effektradius von 250m

Um die Unterschiede zu sehen wurde in Abbildung 35 und Abbildung 36 die Stadt New York gewählt und der Term „morning“ der häufig auftritt. Im ersten Beispiel in Abbildung 35 ist der Effektradius sehr klein gewählt (250m) und man erkennt, dass die Kurven nicht sehr breit sind und dass sich Kurven nur gegenseitig beeinflussen, also aufsummieren, in Regionen die ein sehr hohes Twitteraufkommen haben wie der Stadtteil Manhattan im hinteren Teil des Bildes. Dies wirkt sich auch auf die potentielle Ergebnismenge aus. Damit ein Term als Ergebnis in Betracht gezogen wird, also eine erhöhte Termkonzentration hat, muss er vor Ort auftreten und in näherer Umgebung, damit die Summe der Kurven an der zu analysierenden Stelle bei der Differenzanalyse (vgl. 3.3.4.3) größer ist als eine definierte Schwelle. Wenn der Effektradius aber zu klein gewählt wird, sind die Kurven schon nach kurzer Entfernung so weit abgefallen, dass sie keinen Einfluss auf andere Vorkommen der Terme haben. D.h. die Kurven können sich nicht gegenseitig aufsummieren und so jemals über eine definierte Schwelle kommen. Wählt man den Effektradius also zu klein, so müssten die Tweets mit den signifikanten Termen aus einem sehr kleinen Bereich kommen, damit die Kurven sich aufaddieren können.
Im nächsten Beispiel wurde ein größerer Effektradius von 4km genommen. Die Abbildung zeigt denselben Term und denselben Referenzkorpus.

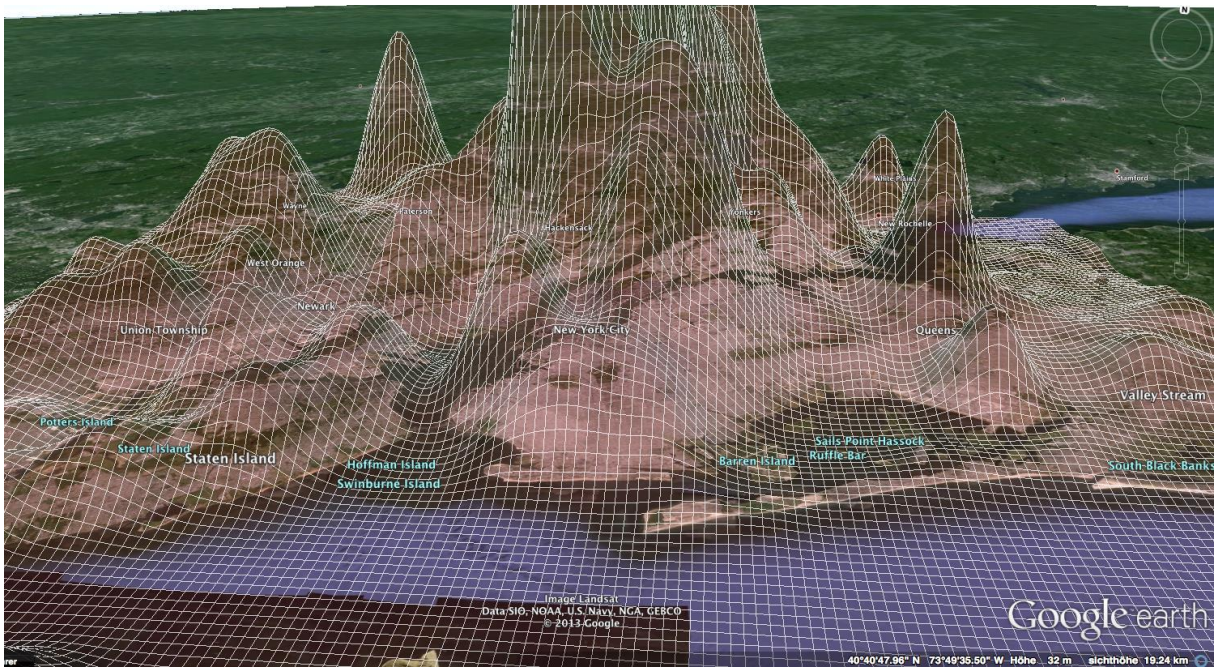

Abbildung 36: Termkonzentrationskarte für den Term "morning" im Referenzkorpus mit den Tweets vom 7.3.-19.03.2014 mit einem Effektradius von 4km

Hier, in Abbildung 36, sieht man deutlich, dass sich die Kurven gegenseitig beeinflussen und sich aufaddieren. Die potentielle Ergebnismenge ist hier größer d.h. die Tweets mit den Termen die auf ein Ereignis hindeuten, können potentiell weiter auseinander sein und es wird trotzdem als Ereignis erkannt. Nimmt man aber einen zu großen Effektradius, so sind die Glockenkurven so breit, dass sie sich alle gegenseitig zu einer großen Glockenkurve aufaddieren.
In wie weit die Ergebnismenge genau durch den Effektradius beeinflusst wird, wird in Kapitel 5 noch einmal an einem konkreten Beispiel gezeigt.
Zusammenfassend lässt sich sagen, dass die Wahl des Effektradius immer vom Anwendungsfall abhängig ist. Möchte man innerhalb einer kleineren Region, z.B. einer Stadt, Ereignisse detektieren und sehr genau lokalisieren, so muss ein kleinerer Effektradius gewählt werden. Doch damit schränkt man unwillkürlich die Ergebnismenge ein bzw. fokussiert sich auf Ereignisse die scharf lokalisiert werden können. Ein Ereignis, welches mehr in der Fläche diskutiert wird und keinen signifikanten geographischen Hotspot hat, könnte so unentdeckt bleiben. Möchte man dagegen eine größere Region analysieren und eine stadtgenaue Auflösung ist ausreichend, so kann mit einen größeren Effektradius gerechnet werden. Dies ist z.B. auch angebracht für Regionen die eine niedrigere Twitter-Nutzung haben wie z.B. in Deutschland. Außerhalb von Ballungsräumen liegen die Tweets in Deutschland mitunter so weit auseinander, das hier mit zu kleinen Effektradien keine Aufsummierungseffekte zu erreichen wären.
Zusammen mit den Schwellwerten der Differenzanalyse (vgl. 3.3.4.3) ist der Effektradius ein Parameter, der die Ereigniserkennung maßgeblich beeinflussen und anpassen kann. Die Werte müssen je nach Anwendungsfall (möchte man nur signifikante Großereignisse detektieren oder auch kleine lokale Ereignisse) und der zu analysierenden Region individuell gewählt werden.

3.4 Weitere Anwendungsmöglichkeit der Algorithmen

Die konzipierten Algorithmen aus Kapitel 3.3 lassen sich auch für andere Problemfelder, in einer abgewandelten Form, einsetzen, um ortsabhängige Analysen mit Echtzeitdaten aus sozialen Netzwerken durchzuführen. Bei der vorgestellten Ereignisdetektion war das Ziel, Veränderungen in der ortsabhängigen Termhäufigkeit zu erkennen, um darüber auf neue Ereignisse zu schließen.
Um zu zeigen, dass die Konzepte auch für andere Aufgaben genutzt werden können, soll nun eine Stimmungsanalyse auf den Tweets realisiert werden. Das bedeutet, dass die Funktion zur Berechnung der ortsabhängigen Termhäufigkeit gegen eine Funktion, welche einen Stimmungswert eines bestimmten Terms berechnet, ausgetauscht werden muss. Der Stimmungswert wird wieder ortsabhängig errechnet. So kann später überprüft werden, ob die Stimmung sich ortsabhängig für einen bestimmten Term geändert hat.
Ziel soll es hier aber nicht sein, eine ausgefeilte Stimmungsdetektion zu konzipieren, die treffsicher alle Tweets korrekt bewerten kann und auf Besonderheiten wie die Erkennung von Ironie oder komplexe Satzstellungen acht nehmen kann, sondern es soll gezeigt werden, dass eine Stimmungsanalyse auch ohne aufwendige Datenvorverarbeitung oder andere Vorarbeiten, wie z.B. vorbewertete Termlisten, möglich sein kann und dass man in Verbindung mit den vorangegangenen Algorithmen neue Anwendungsgebiete erschließen kann. Trotz einer vielleicht ungenauen Stimmungsanalyse bei bestimmten Tweets lassen sich, durch die Masse an Daten, schlussendlich doch interessante Ergebnisse ablesen. Doch dazu später mehr in Kapitel 5.

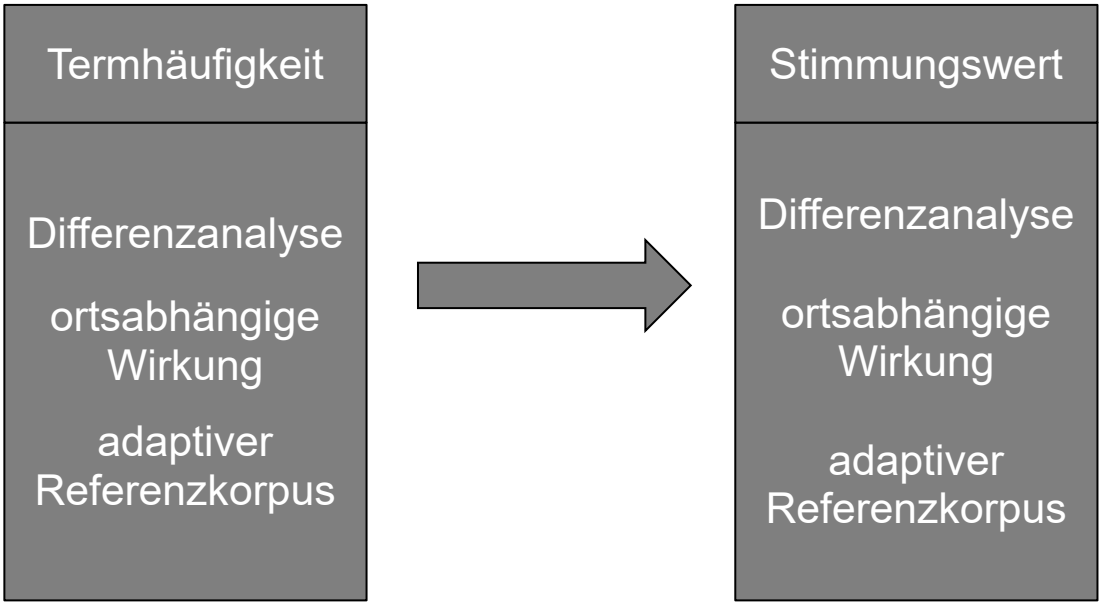

Abbildung 37: Nutzung der konzipierten Algorithmen auf neue Problemfelder

In Abbildung 37 sind die wichtigsten Teile der zuvor konzipierten Ereigniserkennungsalgorithmen skizziert. Bei der Übertragung auf ein neues Problemgebiet werden Teile der Algorithmen wieder verwendet und die entscheidende Funktion, bei der Ereigniserkennung die Errechnung der Termhäufigkeit, gegen eine neue Funktion, hier die Berechnung des Stimmungswertes eines bestimmten Terms und Ortes, ausgetauscht. Das Ziel ist es, mit Hilfe der Analyse ortsgenaue Stimmungsschwankungen zu detektieren. So wie Ereignisse erkannt werden, sollen

darüber dann emotionale Ereignisse erkannt werden, z.B. dass sich die Stimmung bezogen auf einem Term (z.B. gegenüber einer Firma, einem Produkt) sich plötzlich verschlechtert aufgrund von Produktproblemen oder Dienstleistungsausfällen. Auch die Detektion der Stimmung gegenüber Politiker z.B. wäre denkbar, um ihre aktuelle Beliebtheit zu bestimmen. Für was die Stimmungsanalyse bzw. Sentiment-Analyse eingesetzt wird und für welche Anwendungsfälle die Analyse der Schwankungen genutzt werden kann, wurde in Kapitel 2.6.1 bereits erläutert.

3.4.1 Konzeption eines sprachunabhängigen Stimmungswertes

Wie bei der Ereigniserkennung zuvor soll auch hier wieder eine sprachunabhängige Analyse konzipiert werden, was eine Besonderheit darstellt, da die Stimmungsanalysen i.d.R. sprachabhängig sind. Auch auf eine intensive Vorverarbeitung der Daten, wie sie in den vorgestellten Veröffentlichungen in Kapitel 2 häufig zu finden ist, soll verzichtet werden. Um sprachlich nicht eingeschränkt zu sein, soll auf vorgefertigte Listen von Termen mit vordefinierten Stimmungswerten verzichtet werden.
Genutzt wird für die Bestimmung der Stimmung nur eine Liste von so genannten Emoticons. Emoticons werden häufig in Tweets genutzt, um Stimmungen auszudrücken, da sie platzsparend eingesetzt werden können. In Kapitel 2.6.2 wurde gezeigt, dass auch andere Arbeiten Emoticons zur Berechnung eines Sentiment-Wertes einsetzen, doch wurden die Emoticons nur als zusätzliche Informationsquelle genutzt und sich, bei den vorgestellten Arbeiten, nicht komplett darauf gestützt. Der Vorteil der Emoticons ist, dass sie sprachunabhängig sind bzw. sie werden in verschiedenen Sprachen eingesetzt und haben dort die gleiche positive oder negative Bedeutung. Für die Analyse wird eine Auswahl der gängigsten und eindeutig zu bewerteten Emoticons verwendet und diese, der Einfachheit halber, in zwei Klassen, nämlich in positiv und negativ, eingeordnet.

Stimmung	Emoticons
positiv	`:), :-), ;), ;-), :P, ;P, ^.^, :D, ;D, :-D, ;d, -D, ,-)`
negativ	`:(, :-(, :|, :-|, :~(, :'-(, :,(`

Tabelle 2: Genutzte Emoticon-Liste zur Stimmungsanalyse

Um den Stimmungswert, nachfolgend als SENTI-Wert bezeichnet, eines Tweets bzw. eines Terms zu berechnen, geht man folgendermaßen vor. Gegeben sei eine Menge an Tweets, die bewertet werden sollen. Aus dieser Menge an Tweets werden nun alle Tweets entfernt, die kein Emoticon aus Tabelle 2 enthalten oder aber gleichzeitig ein Emoticon aus der positiven und aus der negativen Gruppe enthalten. Die Menge der Tweets, die man nun hat, wird nachfolgend mit C bezeichnet. Hat der Tweet ein positives Emoticon, so bekommt er den SENTI-Wert von 100. Bei einem negativen Emoticon bekommt er den Wert -100. Das führt am Ende dazu, dass der finale SENTI-Wert der Terme bzw. des Tweets ebenfalls zwischen -100, für sehr negativ, bis 100, für sehr positiv, liegt.
Gleichzeitig wird dieser SENTI-Wert auch für jeden Term gemerkt, der in diesem Tweet enthalten war. Hat man alle Tweets in C bewertet, so werden die gemerkten SENTI-Werte der Terme näher betrachtet. Für jeden einzelnen Term werden die gemerkten SENTI-Werte aufsummiert und durch die Anzahl der Werte, also die Anzahl der Tweets in denen sie in C enthalten waren, geteilt. Die Teilung durch die Anzahl ihres Auftretens in C, normalisiert den SENTI-Wert. Der SENTI-Wert eines Terms ist daher:

$$SENTI(w) = \frac{\sum SENTI(C_w)}{count(C_w)}$$

Formel 12: Berechnung des SENTI-Wertes für den Term w

Wobei C_w die Teilmenge der Tweets aus C ist, die den Term w enthält und $count(C_w)$ somit die Anzahl der Tweets aus C_w ist.
Nachfolgend soll ein Beispiel den Algorithmus verdeutlichen. Angenommen der Term w ist in 3 Tweets von C gefunden worden. 2 Tweets enthielten ein positives Emoticon und 1 Tweet ein negatives. Der auf ganze Zahlen gerundete SENTI-Wert des Terms w ist somit:

$$SENTI = \frac{100 + 100 - 100}{3} \approx 33$$

Formel 13: Beispielrechnung zur Berechnung des SENTI-Wertes für den Term t

Ein SENTI-Wert von 33, also ein Wert größer 0, bedeutet eine leicht positive Bewertung.
Um den SENTI-Wert aussagekräftiger zu machen, ist eine neue Schwelle notwendig. Die neue Schwelle gibt an, in wie vielen Tweets in C der Term w mindestens auftreten muss damit man den SENTI-Wert von w berechnen kann. In je mehr Tweets der Wert auftritt, umso aussagekräftiger wird der SENTI-Wert, da er sich auf mehr Daten stützen kann. Gäbe es keine Schwelle und man findet den Term w z.B. nur in einem Tweet mit einem positives Emoticon, so würde der Term, nach der Formel 12, sofort mit einem SENTI-Wert von 100, also sehr positiv bewertet werden.

$$SENTI(w) = \begin{cases} \frac{\sum SENTI(C_w)}{count(C_w)} \; f\ddot{u}r \; count(C_w) \geq m \\ 0 \; f\ddot{u}r \; count(C_w) < m \end{cases}$$

Formel 14: Berechnung des SENTI-Wertes für den Term w mit Beachtung der Mindestauftrittsschwelle m für den Term w in der Tweet-Menge C

Um den SENTI-Wert eines Tweets t zu berechnen, werden die einzelnen SENTI-Werte der Terme in diesem Tweet (w_t), ganz trivial zusammengerechnet und durch die Anzahl der Terme in den Tweet geteilt. Dieses Vorgehen beachtet natürlich so etwas komplexes wie Ironie in einem Text überhaupt nicht. Dieses Vorgehen stellt nur eine einfache Möglichkeit dar etwas über die vermeintliche Stimmung des Tweets auszusagen.

$$SENTI(t) = \frac{\sum SENTI(w_t)}{count(w_t)}$$

Formel 15: Berechnung des SENTI-Wertes für den Tweet t

Die SENTI-Werte von Tweets, die schon ein Emoticon aus Tabelle 2 beinhalten, brauchen nicht neu berechnet werden, da ihr SENTI-Wert ja schon durch das Emoticon vorgegeben ist (entweder 100 oder -100).

3.4.2 Ortsabhängige Stimmungsanalyse

Der Algorithmus zur Bestimmung des SENTI-Wertes kann nun mit den zuvor konzipierten Algorithmen aus Kapitel 3.3 kombiniert werden. Im nachfolgenden wird

hauptsächlich auf die Änderungen der Algorithmen eingegangen. Ansonsten läuft die Analyse vergleichbar mit den beschriebenen Verfahren aus Kapitel 3.3 ab.

3.4.2.1 Erzeugung der Korpora

Im ersten Schritt müssen wieder die entsprechenden Korpora erzeugt werden. Wieder werden nur Tweets genommen die auch eine Koordinate enthalten. Hinzu kommt, das nur Tweets genommen werden, welche ein Emoticon aus Tabelle 2 enthalten. Da dies die Tweet-Menge sehr stark einschränkt, müssen die Korpora (Analyse- und Referenzkorpus) zeitlich breiter ausgelegt sein, damit genug Tweets für die Analyse in den Korpora bereitstehen.
Die Korpora bewegen sich, wie in Kapitel 3.3 beschrieben, gemeinsam durch die Zeit. Auch hier haben wir wieder einen adaptiven Referenzkorpus. Dies ermöglicht wieder, dass Änderungen in der Stimmung mit der Zeit gelernt werden und nicht mehr als neu detektiert werden. Durch den zeitlich breiteren Analysekorpus ist die Schnelligkeit der Erkennung geringer als bei der Detektion von neuen Ereignissen, da die zu analysierende Menge an Tweets auch geringer ist. Weiterhin muss sich ein geänderter SENTI-Wert erst im Analysekorpus ausbreiten, damit er bei einem Vergleich mit dem Referenzkorpus Wert erkannt werden kann.

3.4.2.2 Berechnung des ortsabhängigen SENTI-Wertes

Wenn die Korpora erzeugt sind, kann der nächste Schritt beginnen, indem durch alle Tweets des Analysekorpus iteriert wird und für jeden Term eines Tweets der SENTI-Wert für den jeweiligen Ort errechnet wird. Dabei wird der SENTI-Wert für den Term im Analysekorpus sowie im Referenzkorpus für den zu analysierenden Ort bestimmt. Zur Berechnung wird folgende Formel eingesetzt, welche auf der Formel 8 basiert (Korrekturfaktoren wegen der Erdkrümmung sind hier aber, zur besseren Darstellung weggelassen), die in Kapitel 3.3.4.2 beschrieben wurde.

$$SENTI_w(x,y) = \frac{\sum SENTI_n * e^{-\left(\left(\frac{x_n - x}{effectRadius*0{,}5}\right)^2 + \left(\frac{y_n - y}{effectRadius*0{,}5}\right)^2\right)}}{\sum e^{-\left(\left(\frac{x_n - x}{effectRadius*0{,}5}\right)^2 + \left(\frac{y_n - y}{effectRadius*0{,}5}\right)^2\right)}}$$

Formel 16: Berechnung des ortsabhängigen SENTI-Wertes für einen Term (einfachheitshalber ohne Korrekturfaktoren zur Ausgleich der Erdkrümmung)

Der Einfachheit halber sind wieder die Korrekturfaktoren weggelassen, um eine Verkrümmung der Kurven auszugleichen, da die realen Berechnungen ja auf der Erde stattfinden.
Für den Term w soll an den Ort x, y der ortsabhängige SENTI-Wert errechnet werden. Dazu wird der jeweilige Korpus nach weiteren Tweets durchsucht die auch den Term w beinhalten. Die Tweets im Korpus haben je ein Emoticon aus der Tabelle 2, denn dies war eine Bedingung bei dem Bau des Korpus. Somit ist bekannt, ob der Tweet positiv oder negativ bewertet wurde. Dieser Wert, also entweder -100 oder 100, ist der $SENTI_n$-Wert. Multipliziert wird das Ganze mit dem Wert der Exponentialfunktion, die bestimmt, wie weit der Wert überhaupt mit in die Berechnung eingeht. Bei dieser Berechnung ist es nun nötig, dass dieser Wert noch durch die Summe der Exponentialfunktionswerte geteilt werden muss, um den Wert zu normieren.
Um den ortsabhängigen SENTI-Wert zu berechnen, ist die Mindestauftrittsschwelle zu beachten, die in Kapitel 3.4.1 erläutert wurde. Liegen zu wenige Daten vor, kann der SENTI-Wert nicht verlässlich ermittelt werden und es wird der nächste Term bzw. der nächste Tweet betrachtet. D.h. die Formel zur Berechnung des ortsabhängigen SENTI-Wertes muss noch um die Überprüfung erweitert werden.

$$SENTI_w(x,y) = \begin{cases} \frac{\sum SENTI_n * e^{-\left(\left(\frac{x_n - x}{effectRadius*0{,}5}\right)^2 + \left(\frac{y_n - y}{effectRadius*0{,}5}\right)^2\right)}}{\sum e^{-\left(\left(\frac{x_n - x}{effectRadius*0{,}5}\right)^2 + \left(\frac{y_n - y}{effectRadius*0{,}5}\right)^2\right)}} & für \sum e^{-\left(\left(\frac{x_n - x}{effectRadius*0{,}5}\right)^2 + \left(\frac{y_n - y}{effectRadius*0{,}5}\right)^2\right)} \geq m \\ 0 & für \sum e^{-\left(\left(\frac{x_n - x}{effectRadius*0{,}5}\right)^2 + \left(\frac{y_n - y}{effectRadius*0{,}5}\right)^2\right)} < m \end{cases}$$

Formel 17: Berechnung des ortsabhängigen SENTI-Wertes für einen Term mit Beachtung der Mindestauftrittsschwelle an den Ort x,y

Das Ergebnis ist dann der ortsabhängige SENTI-Wert für den Term w an der Stelle x, y, der auf den Daten des jeweiligen Korpus errechnet wurde.
Der Wert des Effektradius kann für die Analyse natürlich verschieden zu dem Effektradius bei der Ereignisdetektion sein. Auch hier ist der Effektradius wieder nach Anwendungsfall und Tweet-Dichte (Region) anzupassen.

3.4.2.3 Untersuchung der errechneten ortsabhängigen SENTI-Werte

Im Gegensatz zur Ereignisdetektion, müssen die errechneten SENTI-Werte nicht erst hochgerechnet werden, sondern können sofort miteinander verglichen werden, da es sich bei beiden um bereits normalisierte Werte handeln.
Für jeden Term werden die zwei errechneten SENTI-Werte, einmal der ermittelte Wert aus dem Analysekorpus und einmal aus dem Referenzkorpus, miteinander verglichen. Konnte ein Wert nicht errechnet werden, da er an der Stelle x, y z.B. zu wenig aufgetreten ist, geht man zum nächsten Term über. Ist die Differenz der Werte größer als eine festgelegte Schwelle, so deutet dies auf eine signifikante Stimmungsschwankung an dem jeweiligen Ort hin und sollte dem Nutzer präsentiert werden. Auch hier ist es sinnvoll die Tweets mitzuliefern, in welchen diese Schwankung detektiert wurde, um das stimmungsändernde Ereignis näher zu beschreiben.
Der Wert der Schwelle ist wieder nach eigenen Bedürfnissen anzupassen. Auch hier gibt es nicht den richtigen Wert. Möchte man kleinere Schwankungen detektieren, so muss die Schwelle niedriger angesetzt werden.
Wurden alle Tweets im Analysekorpus analysiert, so hat man als Ergebnis alle Terme und deren Tweets, wo eine bestimmte signifikante Stimmungsschwankung stattgefunden hat. Diese Ergebnisse können nun den Nutzer geeignet präsentiert werden oder weiter aufbereitet werden wie in Kapitel 3.3.5 beispielhaft für die Ereigniserkennung erläutert wurde. Nach der Präsentation der Ergebnisse oder nach einen bestimmten Zeitintervall kann dann eine neue Analyseiteration beginnen.

3.5 Zusammenfassung der konzipierten Algorithmen

In den vorrangegangenen Abschnitten wurde gezeigt, wie eine Ereignisdetektion auf den Daten des sozialen Netzwerkes Twitter realisiert werden kann. Dabei lag das Hauptaugenmerk darauf die geographische Koordinate bei der Analyse besser zu nutzen, um ortsabhängige Analysen zu realisieren und die Qualität der Ereignisdetektion zu verbessern, indem ortsabhängige Besonderheiten (z.B. ortsabhängige Termhäufigkeiten) berücksichtigt werden. Auch die Echtzeitfähigkeit der Algorithmen war ein Ziel, welches realisiert wurde z.B. durch adaptive Korpora, die sich dynamisch mit der Zeit mitbewegen und durch eine ressourcenschonende Verarbeitung der eintreffenden Daten. So ist keine aufwendige Datenvorverarbeitung nötig. Dies bringt weitere Vorteile mit sich. So ist die Analyse sprachunabhängig, da die Analyse nicht auf sprachspezifische Vorverarbeitungsschritte wie z.B. Stemming, Filterung von anderssprachigen Termen usw. angewiesen ist. Somit ist die Gefahr, dass

ergebnisrelevante Eingangsdaten irrtümlich herausgefiltert werden, nicht gegeben. Auch eines der größten Probleme der vorgestellten Arbeiten in Kapitel 2.3, die Filterung von „Noise“, spielt für die konzipierte Analyse keine Rolle, da sie darauf nicht angewiesen ist. Ganz im Gegenteil, je mehr Daten in die Analyse eingehen, umso besser können die Ereignisse detektiert werden, da stabilere ortsabhängige Termhäufigkeiten gebildet werden können und so Abweichungen, die auf ein Ereignis hinweisen können, besser und eindeutiger detektiert werden können.
Zuletzt wurde gezeigt, dass die Algorithmen sich mit geringen Modifikationen auch auf andere Probleme anwenden lassen. Dazu wurde eine sprachunabhängige Stimmungsanalyse konzipiert, die ebenfalls auf den Twitterdaten angewandt und die Stimmungsanalyse ortsabhängig durchführen kann. Auch dort ist eine aufwendige Vorverarbeitung auf Aufbereitung der Daten nicht nötig und somit bietet sich auch dieses Verfahren für eine Echtzeitanalyse an.
Das nächste Kapitel beschreibt, wie die konzipierten Algorithmen prototypisch in ein Analysesystem umgesetzt wurden. Das Kapitel 5 widmet sich im Anschluss daran den Ergebnissen, die mit diesen Prototypen erreicht werden können, und zeigt auf, wie gut die konzipierten Algorithmen funktionieren.

4 Realisierung einer ortsabhängigen Analyse in Echtzeitdaten sozialer Netzwerke

In diesem Kapitel geht es um die Realisierung der im vorherigen Kapitel konzipierten Algorithmen in Gestalt eines Prototypen mit dem Namen „See What Happens“ (SWH), welcher Ereignisse aus einem Twitter-Datenstrom heraus detektieren kann. Weiterhin wurde auch eine Stimmungsanalyse umgesetzt, um auch hier zu zeigen, dass neben der Anwendung als Ereignisdetektor, auch weitere Einsatzgebiete der Analysesoftware möglich sind. Es soll hier gezeigt werden, wie das Analysesystem aufgebaut ist, wie bestimmte Probleme gelöst wurden und wie die Software optimiert wurde, um eine gute Performance zu erzielen.

4.1 Entwicklungsziele

Das Ziel bei der Entwicklung des Prototypens war es, ein System zu realisieren, welches aktuelle Ereignisse aus dem Twitter-Datenstrom, von aktuellen Tweets, erkennen und dem Nutzer geeignet präsentieren kann. Dabei sollen nur Tweets betrachtet werden, die eine geographische Koordinate besitzen, d.h. Tweets bei dem der Nutzer den eigenen Standort mit hinzugefügt hat. Der Prototyp soll dabei modular sein, damit während der Entwicklung leicht neue Ansätze oder veränderte Algorithmen zur Ereignisdetektion getestet werden können. Auch soll der Prototyp die Möglichkeit besitzen gleichzeitig mehrere Analysen parallel durchzuführen und es sollen bestimmte Zeitabschnitte mit geänderten Parametern nachgerechnet werden können, um eine optimale Wahl von Parametersets zu finden und die Ergebnisse direkt miteinander vergleichen zu können.
Da die Ergebnisse auch verschiedenen Nutzern präsentiert werden, müssen die Ergebnisse, z.B. auf einer Website, abrufbar sein. Dabei war ein wichtiges Ziel, dass das System über die Website nicht angreifbar ist oder sich darüber die Analyse negativ beeinflussen lässt, um eine kontinuierliche Analyse zu gewährleisten. Weiterhin soll das System so robust sein, dass ein Problem bei einer Analyse das restliche Analysesystem nicht ins Stocken geraten lässt oder es unterbricht. Ein letzter Punkt ist natürlich die Performance der Analyse. Die Analyse muss so performant ablaufen, dass sie in Echtzeit eine größere Region analysieren kann.
In dieser ersten Phase ist die Software als reiner Prototyp zu verstehen, der vorrangig entwickelt wurde, um im Zuge dieser Arbeit Algorithmen auf den realen Twitter-Daten zu testen. Darüber hinaus beinhaltet die Software bereits einige Funktionen, die in einem späteren Service, zu dem die Software ausgebaut werden soll, auch vertreten sein sollen. Folgende Funktionen bietet die Software, im jetzigen Stadium (Stand Januar 2016), einem potentiellen Nutzer:

- Darstellung der aktuellen Ereignisse je nach Region auf einer Website
- Anreicherung der gefundenen Ereignisse mit Beispiel-Tweets und Bildmaterial
- Darstellung des Ereignisses in einer so genannten Heatmap zur Visualisierung der Orte der detektierten Terme
- Stimmungsanalyse-Ergebnisse der gefundenen Ereignisse, Terme und Beispiel-Tweets
- Benachrichtigung per E-Mail oder Direct-Message (Twitter) bei signifikanten Ereignissen (z.B. Katastrophen)
- optionale Erzeugung von Google-Earth-Overlays zur Darstellung der ortsabhängigen Termkonzentration auf einer Karte
- Abo-Funktion zum abonnieren bestimmter Regionen, um auch über kleinere Ereignisse benachrichtigt zu werden

- Darstellung statistischer Daten pro Region (Tweets pro Stunde usw.)

4.2 Aufbau des Analysesystems

Die SWH-Software wurde in mehreren unabhängigen Programmteilen implementiert. Die Software teilt sich auf in dem Importer, die eigentliche Analysesoftware und die Website. Der schematische Aufbau ist in Abbildung 38 dargestellt. Der Importer sowie die Analysesoftware selbst sind in Java [110] realisiert. Die Aufgabe des Importer ist es, die Tweets über die Twitter-Streaming-API [95] entgegen zunehmen und für weitere Analysen (und spätere vergleichende Berechnungen) in eine MySQL-Datenbank [111] abzuspeichern. Die Analysesoftware, wovon mehrere unterschiedliche Varianten parallel laufen können (z.B. Ereigniserkennung und Stimmungsdetektion), kann auf die Datenbank zugreifen, um die gewünschten Tweets abzurufen.

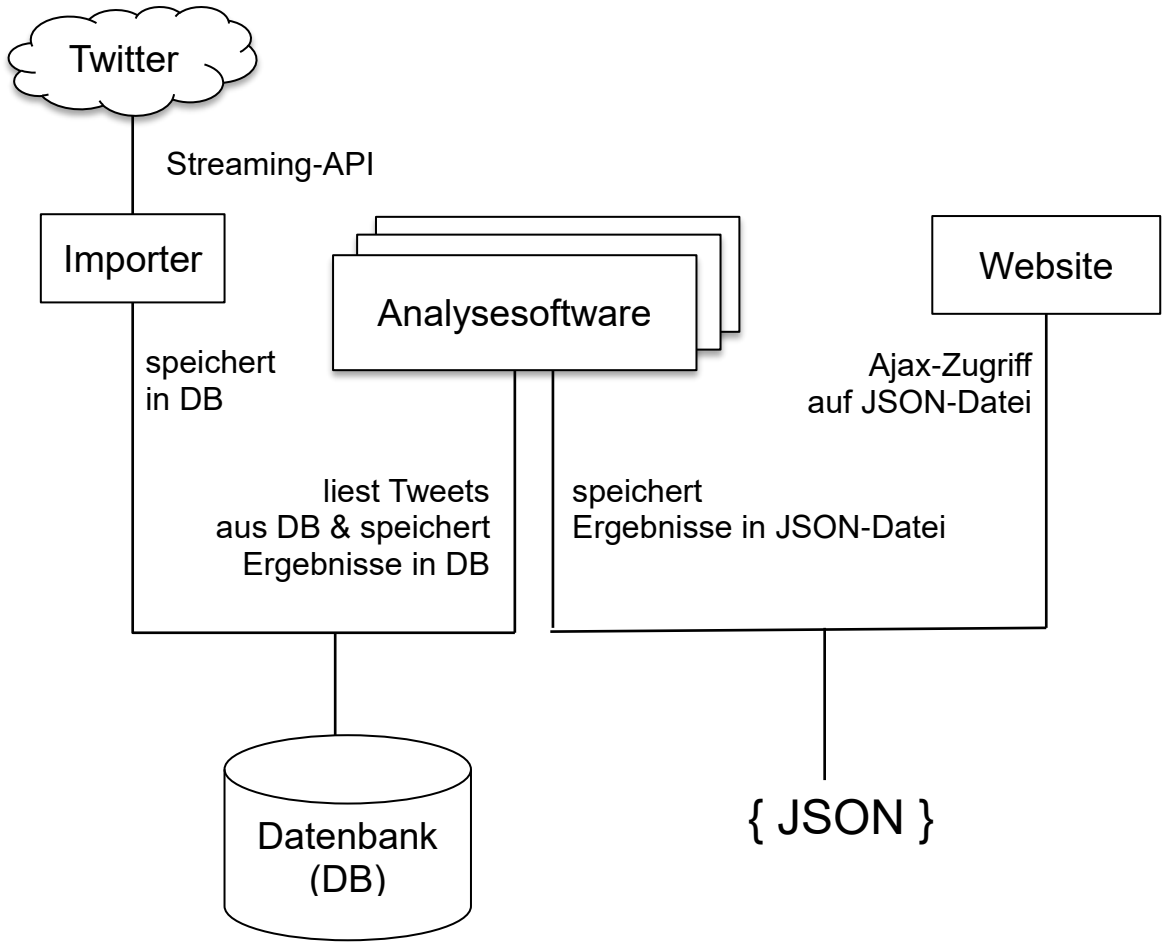

Abbildung 38: Aufbau von SWH mit Importer, Analysesoftware und Website zur Präsentation der Ergebnisse

Die Ergebnisse der Analyse werden zum Schluss ebenfalls in der Datenbank gespeichert. Weiterhin erfolgt eine Speicherung der Ergebnisse in einer JSON-Datei. JSON bzw. die JavaScript Object Notation ist ein Datenformat, welches sowohl von Menschen auch als von einem Programm leicht gelesen werden kann und für den Datenaustausch zwischen Client und Server eingesetzt wird. In JavaScript kann aus JSON leicht wieder ein JavaScript Objekt erzeugt werden. [112] Diese erzeugte JSON-Datei ist wiederum über einen HTTP-Server zugreifbar und wird über JavaScript (Ajax) von der Website dynamisch geladen. Dabei kann zum einem das aktuelle JSON geladen werden, welches

die aktuellen Ergebnisse der Analyse enthält, aber auch eine ältere Version, um ältere Ergebnisse der Analyse auf der Website darzustellen. Die Website präsentiert final die aufbereiteten Ergebnisse der Analyse dem Nutzer. Die Seite ist somit von der Analyse komplett entkoppelt und könnte theoretisch auf einen separaten Rechner liegen. Die Nutzer des Systems haben somit nur Zugriff auf die Website und die JSON-Datei mit den Ergebnissen der Analyse. Durch die Separierung der Teilaufgaben in eigene Programme wird die Stabilität des gesamten Analyseprozesses erhöht. Gibt es z.B. einen Programmfehler in einer Analysesoftware, so kann z.B. der Importer ungestört weiter laufen, um Tweets zu importieren.

4.3 Importer

Der Importer ist die Schnittstelle zu Twitter. Er bekommt über die Streaming-API in Echtzeit von Twitter die neusten Tweets. Als Bibliothek für den Zugriff auf die Twitter Streaming-API wird Twitter4J [113] eingesetzt. Über einen Filter, den man bei der Abfrage angeben kann, werden nur Tweets aus bestimmten Regionen abgerufen. Somit ist auch gewährleistet, dass nur Tweets mit einer geographischen Koordinate zurück geliefert werden.
Zu Beginn der Entwicklung an SWH wurden nur Tweets aus dem geographischen Gebiet von Deutschland und seinem Umland gesammelt. Während der weiteren Entwicklung kamen noch die Gebiete an der Ostküste der USA (von Boston bis Washington, im Weiteren als Boswash [114] bezeichnet (**Bo**ston - **Wash**ington)) und das Gebiet um San Francisco (Bay Area) hinzu, um zu validieren wie sich die Analyse mit anderssprachigen Texten und höheren Tweet-Aufkommen verhält.

Region	**Längengrade**	**Breitengrade**
Deutschland (mit Umland)	5° E bis 15° E	47° N bis 55° N
Boswash (USA Ostküste)	-78° W bis -69° W	38° N bis 43,1° N
Bay Area (San Francisco)	-122,6° W bis -121,5° W	37° N bis 38,2° N

Tabelle 3: Koordinaten der erfassten Regionen in Dezimalgrad

Abbildung 39: Die erfassten Regionen (Bay Area, Boswash, Deutschland) markiert auf Google Maps.

In Tabelle 3 sind die Koordinaten der erfassten Regionen zu finden und in der Abbildung 39 sind die Regionen in einer Karte markiert.
Wenn die Tweets eintreffen, werden sie auf komplette Vollständigkeit der Koordinaten überprüft. Wenn keine exakten Koordinaten enthalten sind, z.B. wenn der Tweet über eine Website oder andere Programme erstellt wurde, die kein Zugriff auf die GPS-Daten hatten [115], oder der Nutzer die exakte Position nicht veröffentlichen möchte wird die Position gemittelt. Ist die Position sehr ungenau angegeben, so wird der Tweet verworfen. In einer früheren Version der Software (bis August 2015) wurden alle

Tweets verworfen, die keine exakten Positionsangaben besaßen. Es konnte aber im Verlaufe des Jahres 2015 beobachtet werden, dass nun meist ungenaue Positionsangaben anstatt exakte Positionen in den Streaming-Daten enthalten sind. Dies ist damit zu begründen, dass die Software-Clients der Nutzer aus Datenschutzgründen die Position nur noch ungefähr angeben. Da diese ungenauen Angaben nun die Mehrzahl der Tweets betrifft, können diese ungenauen Positionsdaten nun nicht weiter ignoriert werden und fließen in die Berechnung mit ein. Sind die eingetragenen Koordinaten des Tweets brauchbar, so wird der Tweet in einer MySQL-Datenbank für weitere Analysen hinterlegt. Neben den Text des Tweets und dessen Koordinaten, liefert die Streaming-API noch sehr viel mehr Daten mit [116] von denen einige davon mit in der Datenbank gespeichert werden. Dies ist z.B. der exakte Zeitstempel des Tweets, die User-ID und User-Namen des Nutzers usw. Diese zusätzlichen Daten werden zur späteren Abfrage der Tweets aus der Datenbank (z.B. Zeitstempel) benötigt bzw. die Daten werden dann auf der Website mit angezeigt wie z.B. der User-Namen des Nutzers. Vollständigkeitshalber ist in Abbildung 40 das Entity-Relationship-Modell der zwei Tabellen dargestellt, in welche die Daten aus der Streaming-API gespeichert werden. Die twitterstatus-Tabelle enthält vornehmlich die Daten des Tweets selbst wie z.B. den Zeitstempel (created), die Koordinaten (lat, lon), den Inhalt des Tweets (text) usw. Die Nutzerdaten des Senders des Tweets werden dagegen in der Tabelle twitteruser hinterlegt. Trifft ein Tweet von einem Nutzer ein, der schon in der twitteruser-Tabelle vorhanden ist, so werden die Daten des Nutzers nur aktualisiert.

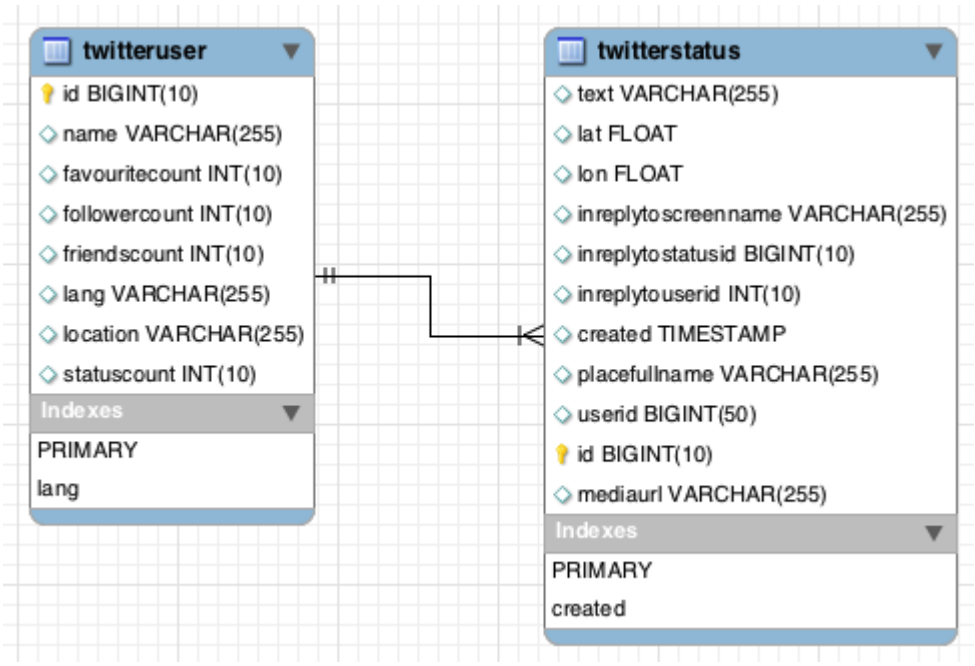

Abbildung 40: Entity-Relationship-Modell der zwei Tabellen in dem die eintreffenden Twitter-Daten gesammelt werden

4.4 Analysesoftware

Durch die Modularisierung des Analysesystems kann es mehrere Analyse-Programme geben die parallel laufen können. Neben der eigentlichen Ereignisdetektion und Stimmungsdetektion wurden während der Entwicklung weitere spezielle Programme entwickelt, um spezifische Werte zu ermitteln, z.B. zur Untersuchung wie die SENTI-Werte bei spezifischen Termen über die Zeit schwanken, ein Programm zur Erzeugung von Termkonzentrationskarten, ein Programm zur Analyse des Einflusses des

Effektradius auf die Ergebnismenge bei der Ereignisdetektion, usw. Die Ergebnisse dieser speziellen Analyseprogramme werden in Kapitel 5 näher erläutert. Hier soll es aber nun um die eigentliche Ereignisdetektionssoftware gehen.
Die Ereignisdetektion (hier ist auch die Stimmungsdetektion gemeint, da sie für die gleichen Regionen und zu gleichen Zeiten parallel zur Ereignisdetektion mit durchgeführt wird) wird in den oben beschriebenen Regionen (Bay Area, Boswash, Deutschland) durchgeführt. Für die Stadt New York wird eine extra Analyse durchgeführt, obwohl die Stadt in der Boswash-Region enthalten ist. Die zusätzliche Analyse für die Stadt New York nutzt kleinere Effektradien und soll Ereignisse in New York auf Stadtteilniveau realisieren. Durch die hohe Nutzung von Twitter in der Stadt, stehen für solch eine detaillierte Analyse genügend Daten zur Verfügung.

4.4.1 Analysezeitpunkte

Um die Analysen so zu gestalten, dass im Zuge dieser Dissertation, bestimmte Zeitabschnitte später noch einmal mit anderen Parametern oder Algorithmen nachgerechnet werden können und die Ergebnisse direkt miteinander vergleichbar sind, müssen die Zeitpunkte zu der die Analysen stattfinden genau übereinstimmen. Wenn dies nicht der Fall wäre, so hätte der Zeitpunkt des Analyseprogrammstarts einen Einfluss auf die Analyse, da je nach Startzeitpunkt unterschiedliche Tweets in den Korpora sich befinden würden. Hat der Analyse-Korpus z.B. eine zeitliche Breite von einer Stunde, so würde man vom Startzeitpunkt bis eine Stunde vor dem Startzeitpunkt alle empfangenen Tweets in diesen Korpus packen. Somit hätte der Startzeitpunkt einen Einfluss auf den Inhalt der Korpora. Um dies zu verhindern, wurde folgendes festgelegt: Die Ereignisdetektion finden genau alle 10 Minuten zu vollen 10 Minuten statt d.h. z.B. 9:10 Uhr, 9:20 Uhr, 9:30 Uhr, usw. Wird die Analysesoftware z.B. um 9:15 Uhr gestartet, so wird eine Analyse für 9:10 Uhr errechnet. Hat der Analysekorpus z.B. eine zeitliche Breite von einer Stunde so kommen für die 9:10 Uhr Analyse alle Tweets in den Analysekorpus, aus der gerade zu untersuchenden Region, die zwischen 8:10 Uhr und 9:10 Uhr eingetroffen waren. Durch dieses festgelegte Raster kommen bei einer späteren Analyse genau die gleichen Tweets wieder in den Analysekorpus und die Ergebnisse sind dadurch miteinander vergleichbar.
Der Referenzkorpus wird dagegen nur einmal am Tag neu berechnet. Sein Raster wurde auf 0:00 Uhr festgelegt. Ein Beispiel soll dies verdeutlichen. Angenommen die zeitliche Breite beträgt 10 Tage und die Analysesoftware wurde am 13.04.2014 um 9:15 Uhr gestartet. So enthält der Referenzkorpus Tweets die zwischen den 03.04.2014 0:00 Uhr und 13.04.2014 0:00 Uhr eingetroffen sind. Somit ist auch für den Referenzkorpus immer genau definiert, welche Tweets er enthält.

4.4.2 Generierung der Korpora

Bei der Generierung der entsprechenden Korpora werden die erforderlichen Tweets aus der Datenbank ausgelesen und daraus die Korpora erzeugt. Für jede zu analysierende Region müssen die geeigneten Referenz- sowie Analysekorpora erstellt werden. Bei den vier Regionen, die untersucht werden, bedeutet dies, dass die Ereigniserkennungssoftware gleichzeitig vier Referenz- bzw. vier Analysekorpora vorhalten muss bzw. acht Analysekorpora, wenn man die Stimmungsanalyse mit einberechnet (der Referenzkorpus der Ereignisdetektion wird für die Stimmungsanalyse mitgenutzt). Die Referenzkorpora werden einmal pro Tag neu erzeugt und stehen nach der Erzeugung für alle nachfolgenden Analysen des Tages zur Verfügung. Die Analysekorpora müssen alle 10 Minuten, also zu Beginn einer neuen Analyseiteration neu erzeugt werden. Um diesen Vorgang zu optimieren, wird hier immer der vorherige Analysekorpus wieder verwendet.

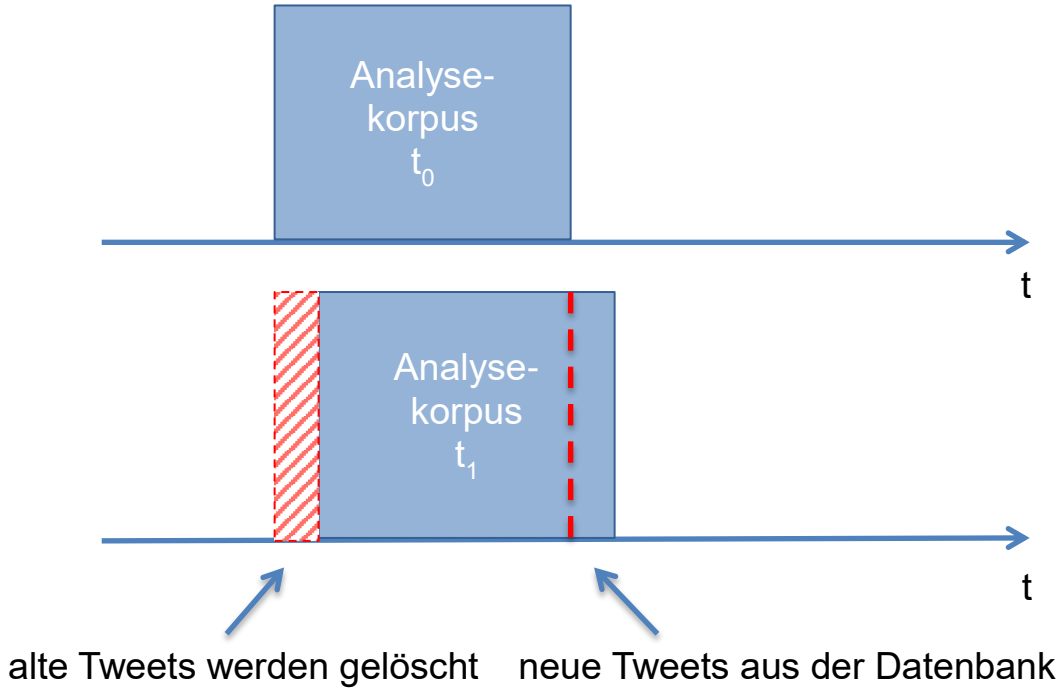

Abbildung 41: Optimierte Aktualisierung des Analysekorpus

Abbildung 41 zeigt diese optimierte Aktualisierung. Es wird der Analysekorpus im vorherigen Schritt verwendet und alle alten Tweets, welche im nächstem Schritt (t_1) nicht mehr enthalten sind, entfernt. Nur die neuesten Tweets, die also im Schritt t_0 noch nicht enthalten sind, werden aus der Datenbank abgerufen. Bei der verwendeten 10-Minuten-Taktung der Analyse werden so immer die neuesten Tweets der vergangenen 10 Minuten abgefragt, was das System stark entlastet.
Wurden die Tweets aus der Datenbank abgerufen, so werden im Anschluss daran Tweet Bursts herausgefiltert, die von einzelnen Nutzern kommen. Es wurde während der Entwicklung beobachtet, dass manche Nutzer, sei es zum Spaß oder um die eigenen Followers zu trollen [117], kurz hintereinander Tweets mit gleichen Texten senden. Da die Position der Tweets alle von den gleichen Ort kommen, wird so die Termkonzentration an dem Ort ziemlich stark erhöht was dazu führte, dass solche Aktionen z.T. in der Ergebnismenge mit landeten. Um diesen Effekt zu verhindern, wurde eine Karenzzeit eingeführt von 15 Minuten für jeden Nutzer. Hat ein Nutzer einen Tweet geschrieben, so werden alle nachfolgenden Tweets, die in den nächsten 15 Minuten von diesem Nutzer kommen ignoriert.
Wurde die Karenzzeit überprüft, so wird jeder Tweet einzeln betrachtet und in die Terme zerlegt. Diese Terme kommen nun in den Korpus, der durch eine Hashtabelle (HashMap) [118] realisiert wurde.
Der Aufbau der Hashtabelle ist in Abbildung 42 dargestellt. Der Term stellt hier den Key der HashMap dar und der Value ist ein Objekt der CorpusDataItem-Klasse. Diese Klasse besitzt neben ein paar Verwaltungsfunktionen den eigentlichen Vektor aus CoordinateTimeItem-Objekten worin das eigentliche Vorkommen der Terme gespeichert wird sowie noch weitere Informationen, wie z.B. aus welchen Tweet dieser Term stammt, den Ort des Tweets, die Nutzer-ID und den Zeitstempel des Tweets.

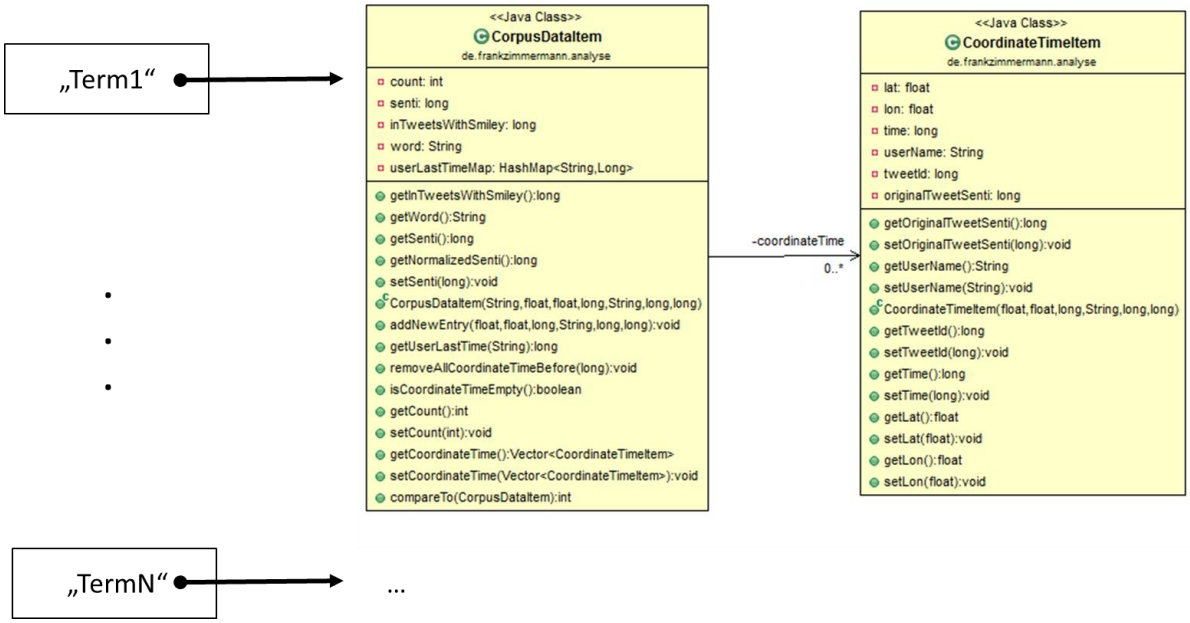

Abbildung 42: Aufbau der Korpus-HashMap

Dieser Daten sind zuvor aus der twitterstatus-Tabelle der Datenbank ausgelesen worden. Ist der Term schon in der HashMap enthalten, d.h. ein CorpusDataItem-Objekt für diesen Term ist vorhanden, so werden nur die zusätzlichen Daten des Auftretens des Terms im CoordinateTimeItem-Vektor vermerkt (hinzufügen bzw. entfernen CoordinateTimeItem-Daten geschieht über die Verwaltungsfunktionen des CorpusDataItem-Objekts). Diese HashMap stellt somit den eigentlichen Korpus (im Arbeitsspeicher) dar.

4.4.3 Analyse des Korpus und Ergebnisaufbereitung

Sind die Korpora erzeugt, wird jeder Term in der Analysekorpus-HashMap analysiert, ob die Termkonzentration an dem jeweiligen Ort des Auftretens, signifikant erhöht ist. Dazu werden die ortsabhängigen Termhäufigkeiten errechnet (aus dem Analysekorpus für den Ort und aus dem Referenzkorpus für diesen Ort) und anschließend miteinander verglichen. Der genaue Algorithmus zur Ereignisdetektion wurde schon in Kapitel 3.4.2 erläutert und soll hier nicht Gegenstand der Beschreibung sein.
Die erforderlichen Daten zur Berechnung der Termkonzentration, d.h. die Orte des Auftretens des Terms, befinden sich in den HashMaps in den jeweiligen CoordinateTimeItem-Objekten.
Hat man im vorherigen Schritt alle Terme gefunden, die eine signifikant erhöhte ortsabhängige Termhäufigkeit haben (größer als eine festgelegte Schwelle), so wird im nächsten Schritt versucht die Terme so zu gruppieren, dass Ereignis-Cluster entstehen. Realisiert wird dies mit den NewsCluster-Objekten, die für je ein Ereignis stehen. Jedes NewsCluster-Objekt beinhaltet wiederum einen Vektor von Tweet-Objekten, die die Beispiel-Tweets repräsentieren, welche dem Ereignis zugeordnet wurden. Da man für jeden Term genau weiß aus welchen Tweet er stammt (gespeichert im CoordinateTimeItem-Objekt), können die betreffenden Tweets untersucht werden, ob sie auch andere Terme aus der Ergebnismenge enthalten. Wenn dies so ist, dann gehören sie zu diesem Ereignis-Cluster und werden in das jeweilige NewsCluster-Objekt aufgenommen. Auf eine aufwendigere Ereignis-Clusterung, wie sie in Kapitel 3.3.5.1 skizziert wurde (z.B. einer Ausdünnung des entstandenen Graphen), wurde bei der Implementierung dieses Prototypens einfachheitshalber verzichtet.

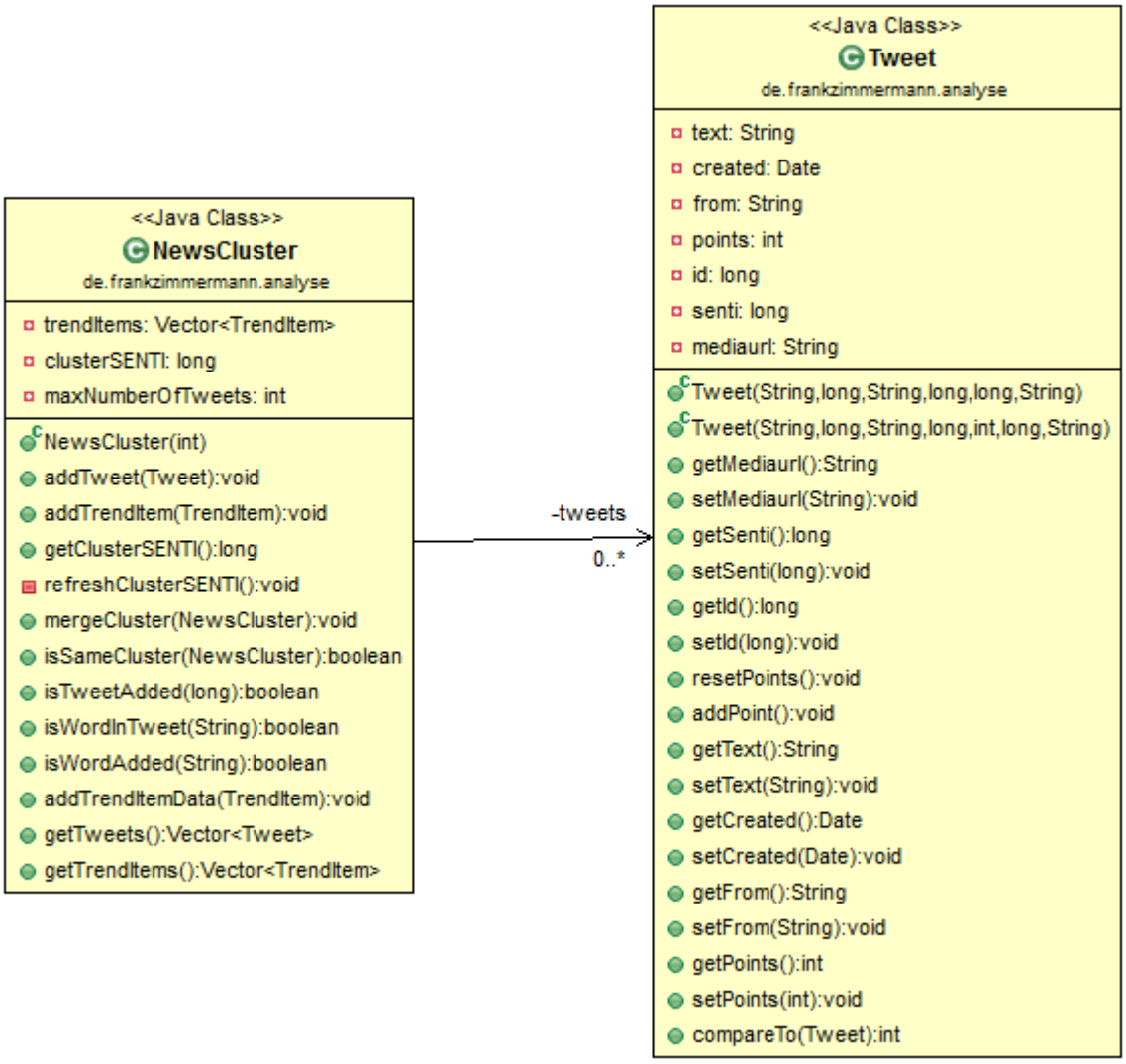

Abbildung 43: UML-Klassendiagramm um den Zusammenhang zwischen der NewsCluster-Klasse und der Tweet-Klasse darzustellen

Die Ergebnisse der Analyse werden in die JSON-Datei und in die Datenbank geschrieben. Die Abbildung 44 zeigt einen Auszug aus einer generierten JSON-Datei.
Die Ergebnisse werden zusätzlich in die Datenbank gespeichert, da mit diesen Daten errechnet wird, ob die Menge der Ergebnisse zu einem bestimmten Zeitpunkt normal oder stark erhöht ist. Dazu wird, wie bereits in Kapitel 3.3.5.2 erwähnt, der Durchschnitt der Menge an detektierten Ereignissen für die aktuelle Stunde über die letzten 4 Wochen gemittelt.
Ist die aktuelle Anzahl von detektierten Ereignissen mehr als 3-mal so groß, so ist von einem Großereignis auszugehen und die Nutzer können gesondert informiert werden. Die Analysesoftware schickt in diesem Fall eine E-Mail an eingetragene Nutzer mit den Ergebnissen der Analyse, aufbereitet nach den entdeckten Ereignis-Clustern (Beispiel E-Mail in Abbildung 45 dargestellt). Die SWH-Software ist auch in der Lage Twitter-Direct-Messages an bestimmte Nutzer zu schicken. Somit wird Twitter selbst wieder als Plattform genutzt, um die Ergebnisse der Analyse zu verbreiten.

```
"trendItems": [
    {
        "word": "#thanksgiving",
        "averageFrequency": 0,
        "shortTimeFrequency": 0,
        "ascent": 831,
        "termConcentrationShort": 28.617337774938523,
        "termConcentrationLong": 590.2707905439702,
        "lat": 40.7807,
        "lon": -73.9685,
        "tweetId": 669584106649477100,
        "senti": 100,
        "userName": "275681557",
        "fileName": "tag_thanksgiving",
        "items": [
            {
                "word": "#thanksgiving",
                "averageFrequency": 0,
                "shortTimeFrequency": 0,
                "ascent": 831,
                "termConcentrationShort": 28.617337774938523,
                "termConcentrationLong": 590.2707905439702,
                "lat": 40.7807,
                "lon": -73.9685,
                "tweetId": 669584111397429200,
                "senti": 100,
                "userName": "241802289",
                "fileName": "tag_thanksgiving",
                "items": []
            },
            {
                "word": "#thanksgiving",
                "averageFrequency": 0,
                "shortTimeFrequency": 0,
                "ascent": 623,
                "termConcentrationShort": 6.404732417440063,
                "termConcentrationLong": 169.97232425950958,
```


Abbildung 44: Auszug aus einer JSON-Ergebnis-Datei zur Darstellung der Ergebnisse auf der Website

Abbildung 45: Benachrichtigungs-E-Mail der SWH-Software bei der Detektion von ungewöhnlich hoher Aktivität in einer Region. Hier anlässlich des Jahrestages des Boston Attentats.

4.4.4 Zusätzliche Funktionen

Die SWH-Software bietet auch optionale Features an, die über kleine Änderungen im Quellcode zugeschaltet werden können. So kann der Prototyp für jeden Term, der eine erhöhte Termhäufigkeit hat, eine Termkonzentrationskarte aus den aktuellen Daten des Analysekorpus und des Referenzkorpus anlegen. Um die Termhäufigkeit besser zu erkennen, werden die Werte um den Faktor 1000 verstärkt. Als Ergebnis erhält man ein Overlay für Google Earth, welches sich über die analysierte Region legt und die Termhäufigkeit mit Bergen und Tälern optisch anzeigt. Je höher die Termhäufigkeit an einen bestimmten Ort ist, umso höher sind die überlagerten Berge. Die SWH-Software

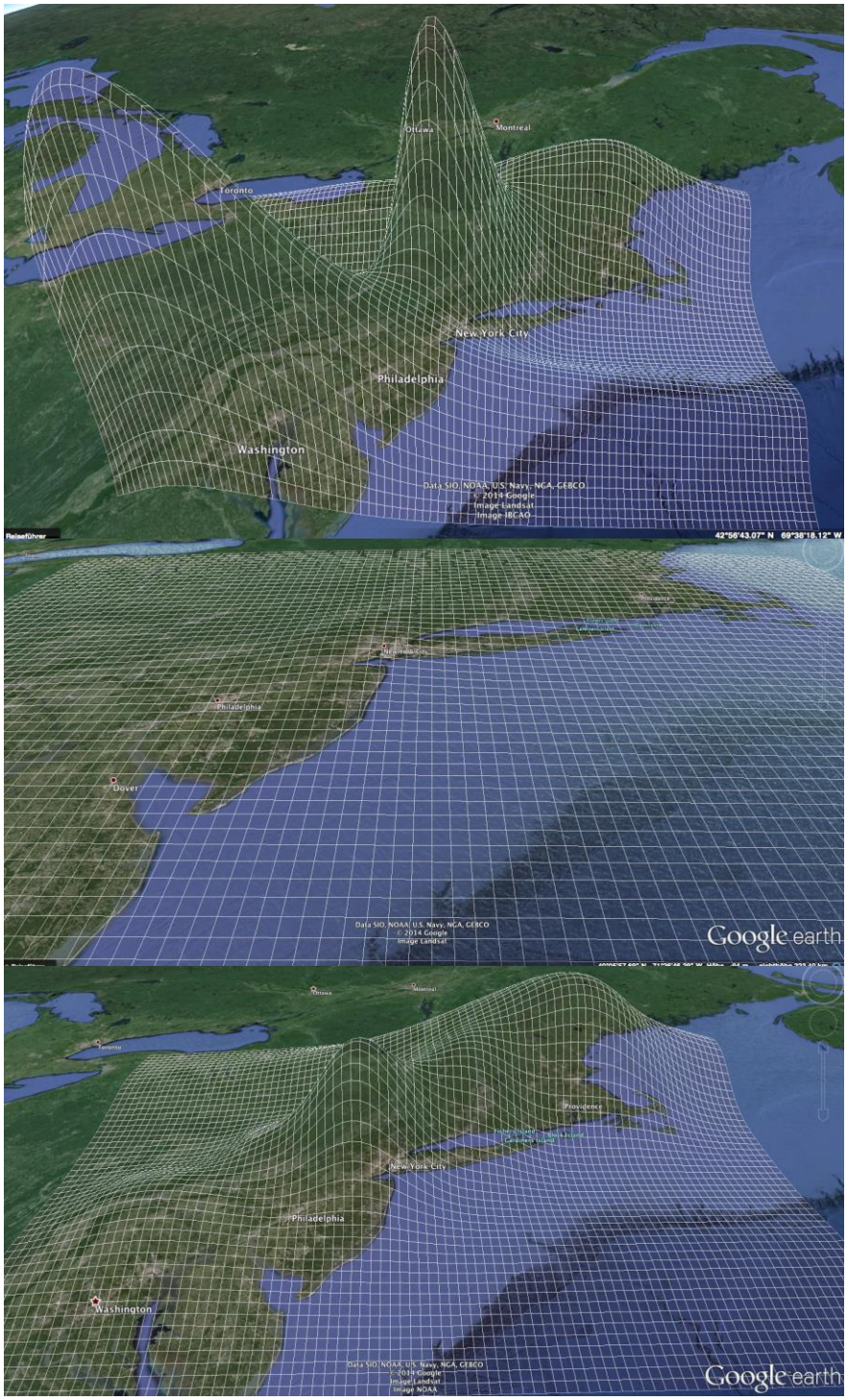

Abbildung 46: Errechnete Termhäufigkeitskarten für den Term "alert" in der Region Boswash am 11.03.2014 20:20 Uhr. Von oben nach unten: Termhäufigkeitskarte berechnet aus dem Referenzkorpus, Analysekorpus und Differenzkarte.

generiert dazu kmz-Dateien (komprimierte kml-Dateien (sihe Beispiel Auszug in Abbildung 46)), die mit dem Google-Earth-Programm angezeigt werden können (erzeugte kmz-Datei importieren). Die Funktion, ein teiltransparentes Overlay über eine Karte zu legen (siehe Abbildung 46), gibt es standardmäßig nicht. Das Overlay wird mittels erzeugten Linien und Teilflächen zusammengesetzt, die zusammen ein teiltransparentes Overlay ergeben. [119] In einer weiteren Termkonzentrationskarte wird die Differenzanalyse optisch dargestellt, indem die Termhäufigkeit des Analysekorpus hochgerechnet wird und der Wert des Referenzkorpus an der Stelle abgezogen wird. Als Ergebnis erhält man eine Karte mit z.T. extrem steilen Erhebungen. Genau an diesen Stellen wurden die signifikanten Termhäufigkeitsänderungen detektiert. D.h. man erhält eine Karte, die die potentiellen Orte der Ereignisse anzeigt. Aufgrund des erhöhten Rechenaufwands zur Erstellung der Karten, kann diese Funktion deaktiviert werden.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
<name>Term Concentration Layer by Frank Zimmermann</name>
<Style id="transRedPoly">
<LineStyle>
<width>1</width>
<color>80ffffff</color>
</LineStyle>
<PolyStyle>
<color>100000ff</color>
</PolyStyle>
</Style><Placemark>
    <name>Absolute</name>
    <visibility>1</visibility>
    <styleUrl>#transRedPoly</styleUrl>
    <Polygon>
      <tessellate>1</tessellate>
      <altitudeMode>absolute</altitudeMode>
      <outerBoundaryIs>
       <LinearRing>
         <coordinates>
           5.0,47.0,1000
           5.1,47.0,1000
           5.1,47.1,1000
           5.0,47.1,1000
           5.0,47.0,1000
        </coordinates>
       </LinearRing>
      </outerBoundaryIs>
    </Polygon>
  </Placemark><Placemark>
    <name>Absolute</name>
    <visibility>1</visibility>
    <styleUrl>#transRedPoly</styleUrl>
    <Polygon>
      <tessellate>1</tessellate>
      <altitudeMode>absolute</altitudeMode>
      <outerBoundaryIs>
       <LinearRing>
         <coordinates>
           5.1,47.0,1000
           5.2,47.0,1000
           5.2,47.1,1000
           5.1,47.1,1000
           5.1,47.0,1000
        </coordinates>
       </LinearRing>
      </outerBoundaryIs>
    </Polygon>
```

Abbildung 47: Auszug aus einer generierten kml-Datei zur Anzeige der Termkonzentrationskarte

Als Beispiel sind in Abbildung 46 die Termhäufigkeitskarten des Terms „alert“ für die Region Boswash abgebildet. Die Karten wurden während des normalen Analysezyklus

am 11.03.2014 um 20:20 Uhr berechnet (beliebig ausgewählter Zeitpunkt). Die erste Karte zeigt die Termhäufigkeit aus den Daten des Referenzkorpus an. Deutlich zu sehen ist, dass es starke Erhebungen im Gebiet New York und Washington gibt. In der zweiten Karte sieht man die Daten aus dem Analysekorpus. Hier sind trotz Verstärkung der Werte kaum Erhebungen in der Karte sichtbar. Erst die Hochrechnung der Daten und Differenz mit den Daten des Referenzkorpus macht das gehäufte Auftreten des Terms, vor allem im Norden, dem Gebiet um Boston, der zu analysierenden Region sichtbar.
Eine weitere Funktion der SWH-Software ist, dass man eine detailliertere Ereignisdetektion für eine bestimmte Region abonnieren kann. Dazu wird als Rückkanal ebenfalls Twitter genutzt. Mit einem Tweet an den Twitter-Account der SWH-Software kann man für die gerade aktuelle Position die Ereignisse abonnieren. Dabei muss die aktuelle Position in dem Tweet mit übermittelt werden. Die SWH-Software hört über die Streaming-API auf diese speziellen Tweets und merkt sich die Abonnementen und deren Position in einer Tabelle. Da der Nutzer die Nachrichten für diese spezifische Region abonniert hat, ist davon auszugehen, dass er auch an kleineren Ereignissen aus dieser abonnierten Region interessiert ist. Zusätzlich wird daraufhin die Ereignisdetektion zusätzlich mit einer niedrigeren Schwelle berechnet, um auch kleinere Ereignisse zu detektieren und den Abonnementen zur Verfügung zu stellen. Zu jeder abonnierten Region wird überprüft, ob sich solche Ereignisse im Umkreis des Abonnenten befinden. Wenn ja, dann werden die Nutzer über Twitter-Direct-Messages über die Ereignisse unverzüglich informiert. Mit einem Stopp-Tweet kann dieses Benachrichtigungs-Abo jederzeit wieder aufgehoben werden.
Zusätzlich zur Ereignisdetektion, wo der Ort in die Berechnung mit eingeht, wird auch eine Ereigniserkennung durchgeführt, wobei der Ort ignoriert wird bzw. wo der Effektradius unendlich ist. Diese Analyse wird als Vergleich mitberechnet, um die Unterschiede zu sehen in wie weit die Hinzunahme des Ortes die Analyse verbessert. Mehr zu den Ergebnissen in Kapitel 5.

4.5 Präsentation der Ergebnisse

Die Ergebnisse der Ereignisdetektion und der Stimmungsanalyse werden auf einer Website dem Nutzer präsentiert. Diese Website ist die einzige Komponente der SWH-Software worauf der Nutzer direkt Zugriff hat. Die Ergebnisse der Analyse werden von der Software in eine JSON-Datei geschrieben, die von der Website über Ajax geladen wird. Um die Seite so zu gestalten, dass man einen guten Überblick über die detektierten Ereignisse bekommt, wurde das Bootstrap-Framework [120] und das jQuery-Framework [121] eingesetzt. Das Bootstrap-Framework wurde ursprünglich von den Twitter-Entwicklern bereitgestellt. Es umfasst vordefinierte CSS-Klassen, die genutzt werden können, um bestimmte Elemente einer Website zu designen. Ebenso gibt es spezielle funktionale Elemente wie Menus, Tooltipps usw. die ebenfalls genutzt werden können. Die Website selbst ist im Responsive-Design gestaltet, dies bedeutet, dass sie sich dem jeweiligem Gerät in der Darstellung anpasst. Somit ist die Website per Desktop Browser bis hin zum Smartphone nutzbar.
Die Abbildung 48 zeigt ein Screenshot der Website. In der Kopfzeile der Webseite befinden sich jeweils Menus mit dem man die gewünschte Analyse, die Region (Area) und die Zeit auswählen kann. Gerade ausgewählt ist die Darstellung der Ergebnisse der Ereignisdetektion (Event Detection) und die Region New York. Weiterhin gibt es noch die „SENTI Detection“, was die Stimmungsanalyse ist und die „Event Detection (without position)“, was die Ereigniserkennung ohne Beachtung des Ortes ist und zuletzt „Settings“, welches die eingestellten Parameter und die Menge an eintreffenden Tweets

anzeigt. Bei den Regionen kann man die Regionen „Germany“, „New York“, „Boswash“ oder „SF Bay Area“ auswählen, um die entsprechenden Ergebnisse zu sehen.

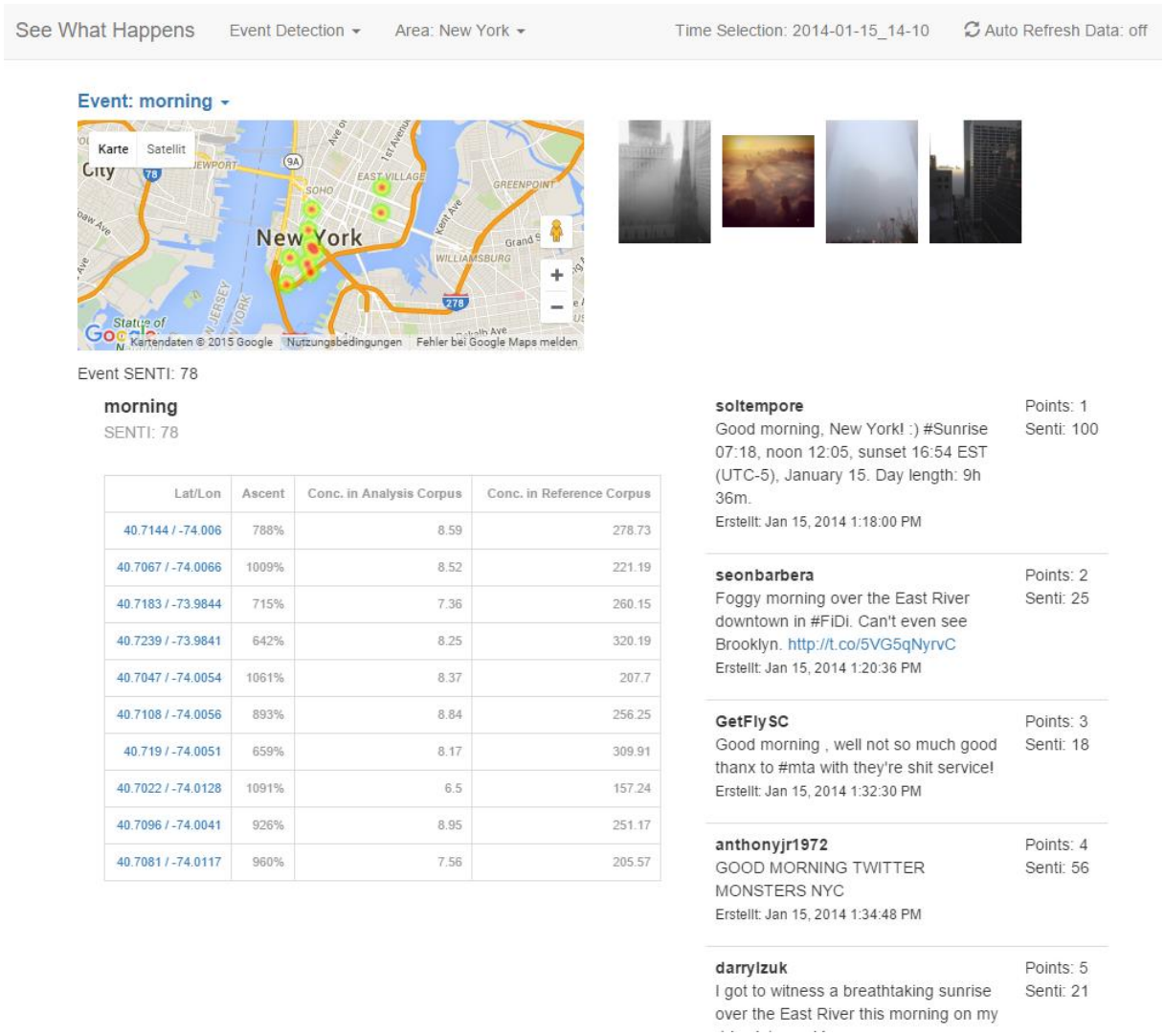

Abbildung 48: Screenshot der Website der SWH-Software mit einem detektierten Ereignis in New York.

Im „Time Selection“ Menu kann man angeben ob man die aktuellen Analyseergebnisse oder ältere Ergebnisse sich anschauen möchte. Will man immer die neuesten Ergebnisse sehen so aktualisiert sich die Seite automatisch und lädt dabei immer die JSON-Daten der neuesten Ergebnisse per Ajax nach. Möchte man diese automatische Aktualisierung nicht, so lässt sie sich ebenfalls deaktivieren „Auto Refresh Data: off“.
Im Zentrum der Seite in Abbildung 48 findet man die Analyseergebnisse gruppiert nach Ereignissen. Die Ereignisse lassen sich mit einem Klick aufklappen. Im Screenshot ist ein geöffnetes Ereignis zu sehen, der nur einen Term („morning“) hat. Sofort erkennbar ist für jedes Ereignis die erzeugte Heatmap. In dieser Heatmap sind die Orte der detektierten Terme des ausgewählten Ereignisses eingezeichnet. Die Heatmap zeigen somit visuell wo das Ereignis stattfindet bzw. in welchen Bereichen die signifikant erhöhten Termkonzentrationen auftraten. Um das detektierte Ereignis näher zu beschreiben, wurden während der Analyse die Links zu den Bildern, die den Tweets hinzugefügt wurden, extrahiert und werden nun gesondert angezeigt. In diesem Fall handelt es sich gerade um einen nebeligen Morgen mit Sonnenaufgang in New York. Diese besondere Wettersituation führte dazu, dass viele New Yorker über den Sonnenaufgang im Nebel auf Twitter schrieben und Fotos dazu posteten. Dies wurde

von der SWH-Analyse als ein Ereignis detektiert und dargestellt. Die extrahierten Bilder beschreiben somit zusätzlich das Ereignis. Unter den Bildern auf der rechten Seite wurden Tweets extrahiert von den Orten, wo der Term signifikant erhöht war. Diese Tweets dienen ebenso dazu, das detektierte Ereignis näher zu beschreiben. In der linken Hälfte findet man die Liste der detektierten Terme. Darunter befindet sich eine Tabelle mit einem Auszug der detektierten Orte, wo diese Termhäufigkeit für diesen Term signifikant erhöht war. Die Tabelle zeigt die Termhäufigkeit des jeweiligen Terms im Analysekorpus und Referenzkorpus und den detektierten Anstieg. Die Längen- und Breitengrade beschreiben für jeden Eintrag exakt die Position, wo die Werte berechnet wurden. Die Koordinaten selbst sind noch einmal verlinkt zu einer Google-Maps-Karte die per Klick in einem neuen Fenster sich öffnet und den Ort genau anzeigt.
Zu guter Letzt wurden für die detektierten Terme, des Ereignisses an sich und die extrahierten Tweets der SENTI-Wert errechnet und mit auf der Seite dargestellt, um zu sehen, wie die Terme bzw. die Tweets stimmungsmäßig bewertet wurden.

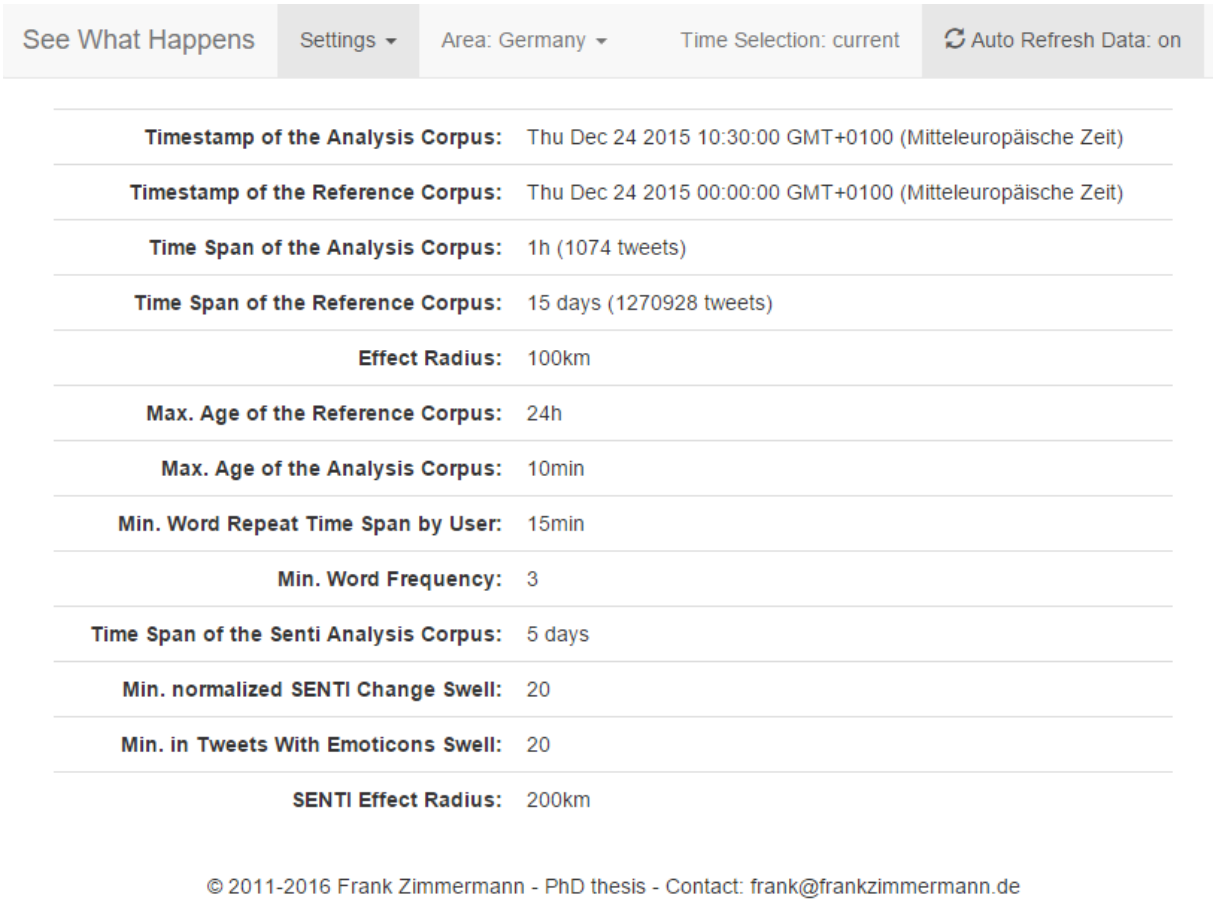

Abbildung 49: "Settings" Übersicht der SWH-Software während der Analyse

Unter den Punkt „Settings“ (siehe Abbildung 49) lassen sich die gerade für die ausgewählte Region eingestellten Parameter abrufen, wie z.B. die Größe des Effektradius, aber auch aktuelle Werte wie z.B. der Zeitstempel der letzten Analyseiteration, die Anzahl der Tweets im Referenz- bzw. Analysekorpus, usw.

4.6 Hardware und Performance

Die komplette SWH-Software inklusive der MySQL-Datenbank, läuft auf einem Mac mini (vom Stand Mitte 2011) mit Core i5 (2415M) „Sandy Bridge“ mit 2,3GHz Taktfrequenz, 8GB RAM und einer 256GB SSD. D.h. auf einem schon etwas älteren Rechner. In Anbetracht der Größe der zu analysierenden Regionen, der Menge an zu verarbeiteten Tweets (pro Tag ca. 590 000 Tweets) und den pro Tag 143-mal laufenden Analyseiterationen, kann die SWH-Software, im jetzigem Prototypenstadium, als ausreichend performant beschrieben werden. Um alle Korpora nach einem Neustart neu zu erstellen, benötigt die Software ungefähr 15 Minuten. Die Aktualisierungen danach wurden, wie in Kapitel 4.4.2 so optimiert, dass sie schneller durchgeführt werden können. Neben diesen Optimierungen gibt es auch weitere Optimierungen um z.B. die Objekte im Speicher so klein wie möglich zu halten, um die Analysegeschwindigkeit weiter zu erhöhen. So werden nur Daten in den Objekten (HashMap) vorgehalten, die auch unbedingt für die Analyse notwendig sind. Bei der Generierung der HashMaps wird auch darauf geachtet, dass veraltete Daten gelöscht werden bevor eine neue HashMap erzeugt wird, um kurzzeitige Speichernutzungsspitzen durch doppelte Objekte (altes Objekt & neu erzeugtes Objekt welches noch nicht der Variable zugewiesen wurde) zu vermeiden. Um den Arbeitsspeicherverbrauch weiter zu optimieren, wurden testweise die Korpora in die Datenbank verlagert. Um die Performance dort zu optimieren, sollten die Korpora-Tabellen nur im Arbeitsspeicher vorliegen (MEMORY Tables) um die vielen Abfragen und Manipulationen in den Korpora zu beschleunigen. Es zeigte sich aber, dass durch dieses Vorgehen die Analyse im Vergleich viel langsamer lief, als wenn die Korpora direkt im Analyseprogramm im Arbeitsspeicher manipuliert werden. Daher wurde diese Art der Optimierung wieder verworfen und die Korpora werden direkt im Analyseprogramm im Speicher gehalten, um verarbeitet zu werden.
Möchte man in Zukunft noch größere Regionen untersuchen, so ist dazu mehr Arbeitsspeicher nötig um die zusätzlichen Daten zu erfassen (performanterer Rechner) bzw. die Software müsste weiter optimiert oder parallelisiert werden damit der Algorithmus ggf. auf mehreren Rechnern laufen kann, um die Ereignisse aus größeren Regionen zu detektieren. Die Parallelisierung müsste so erfolgen, dass mehrere Rechner an der Ereignisdetektion beteiligt sind. Die Korpora müssten dazu in disjunkte Teile zerlegt und verteilt werden. Wie im MapReduce-Ansatz müsste der Algorithmus in getrennten Schritten ablaufen und die Ergebnisse zum Schluss eingesammelt werden. Hierzu können dann Frameworks eingesetzt werden wie z.B. Spark Streaming [102] welches in Kapitel 2.7.4 kurz vorgestellt wurde.

5 Ergebnisse der Analysen

In diesem Kapitel geht es um die Ergebnisse der Analysen, die mit der vorgestellten SWH-Software durchgeführt wurden. Es sollen die Hypothesen auf denen die konzipierten Detektionsalgorithmen beruhen, welche in Kapitel 3 erarbeitet wurden, überprüft werden. Dabei sollen die Ergebnisse der Analysen beurteilt werden, ob das Ziel der Analysen, also der Ereignisdetektion, erfüllt werden konnte. Dabei werden unter anderem die detektierten Ereignisse und die Zeitpunkte der Detektion näher betrachtet und mit den etablierten Medien verglichen, um zu beurteilen, wie schnell die Ereignisse detektiert werden können. Auch die Ergebnisse weiterer speziellerer Untersuchungen werden hier näher begutachtet, um z.B. zu untersuchen, welchen Einfluss der Effektradius auf die Ergebnismenge hat oder welche Erkenntnisse man aus der Stimmungsdetektion bzw. den ermittelten Stimmungsschwankungen ziehen kann.
Im ersten Teil dieses Kapitels stehen die Analysen der Ereigniserkennung sowie die Überprüfung der Hypothesen im Mittelpunkt. Im zweiten Teil folgen die Analysen zur Stimmungsdetektion.

5.1 Ereignisdetektion

Um die eigene Ereignisdetektionsalgorithmen zu evaluieren, sollten die Ergebnisse der SWH-Software mit anderen lauffähigen Systemen, die in den Veröffentlichungen aus Kapitel 2 betrachtet wurden, verglichen werden. Trotz intensiver Recherche konnte kein lauffähiges System online gefunden werden, welches auf den in den Veröffentlichungen aus Kapitel 2 beschriebenen Algorithmen aufbaut. Es gibt natürlich Dienstleister, welche eine Ereigniserkennung als Dienst anbieten, doch sind deren Algorithmen als Blackbox zu sehen und ein objektiver Vergleich kann hier nicht stattfinden, da nicht bekannt ist nach welchen Verfahren dort Ereignisse detektiert werden, wie hoch der manuelle Anteil der Detektion ist und welche Daten dafür genutzt werden.
Auch existierende Benchmarks wie in [28] erwähnt, konnten hier nicht eingesetzt werden, da die Menge an benötigten ortsbezogenen Nachrichten nicht vorhanden waren. Auf die fehlenden Möglichkeiten eines objektiven Benchmarks für Ereignisdetektionsalgorithmen auf den Twitter-Daten geht auch dieses Paper [122] näher ein.
Um dennoch die eigenen Algorithmen einer objektiven Bewertung zuzuführen, wurde die SWH-Software so erweitert, dass sie sowohl mit als auch ohne die Neuerungen Berechnungen durchführen kann. D.h. SWH läuft mit starren Referenzkorpus statt eines adaptiven und ohne die Einberechnung der Position der Kurzmitteilungen. Diese Ergebnisse konnten dann mit der Analyse verglichen werden, bei der die neuen Funktionen enthalten waren, um zu sehen, welche konkreten Verbesserungen die Neuerungen haben.

5.1.1 Verifikation der Hypothesen

Der entscheidende Kern bzw. die Grundidee der in Kapitel 3 konzipierten Ereignisdetektionsalgorithmen kann mit Hilfe von zwei Hypothesen zusammengefasst werden.

1. Mit einem dynamischen Referenzkorpus lässt sich eine bessere Differenzanalyse durchführen, da längerfristige temporäre Termkonzentrationsschwankungen berücksichtigt werden können.
2. Ortsabhängige Referenz- bzw. Analysekorpora verbessern die Differenzanalyse von georeferenzierten Eingangsdaten, da mit ihnen lokale Termkonzentrationsschwankungen berücksichtigt werden können.

Des Weiteren kommen die Detektionsalgorithmen ohne Vorverarbeitung der Eingangsdaten aus d.h. die ganze Analyse ist sprachunabhängig.

5.1.1.1 Überprüfung der ersten Hypothese

Im ersten Schritt soll der Sachverhalt in einem Gedankenexperiment näher betrachtet werden. Gegeben sei ein Referenzkorpus von einer zeitlichen Breite von vier Tagen (Boswash Bsp.) und ein Analysekorpus mit einer zeitlichen Breite von 30 Minuten (192 mal kleiner). Im Referenzkorpus befinden sich 1 Mio. Nachrichten und im Analysekorpus 5000 Nachrichten. Jede Nachricht enthält, der Einfachheit halber, jeweils einen Term. Insgesamt gibt es fünf Terme (W_0-W_4), die im Referenzkorpus zum Zeitpunkt t_0 alle gleichhäufig vorkommen (absolute Häufigkeit jedes Terms im Referenzkorpus ist Hn=200 000 bzw. relative Häufigkeit h_n=0,2).
Die absoluten Häufigkeiten der Terme im Analysekorpus zum Zeitpunkt t_0 sieht folgendermaßen aus:

Term	H_{term}	H_{term_hoch}	a_{term}
W_0	1250	240000	1,2
W_1	1500	288000	1,44
W_2	500	96000	0,48
W_3	750	144000	0,72
W_4	1000	192000	0,96

Tabelle 4: Absolute Häufigkeiten im Analysekorpus zum Zeitpunkt t_0

Die Termanstiegsschwelle sei bei diesem Beispiel 1,3. Damit wäre der Term W_1 über dieser Schwelle und könnte somit als Ergebnis dem Nutzer präsentiert werden.
Zum Zeitpunkt t_1 wird ein weiterer Analysekorpus gebildet. Dabei ist zu beachten, dass zwischen t_0 und t_1 ein signifikanter Zeitraum liegt (z.B. 1 Jahr). Die Häufigkeiten der Terme und die errechneten Anstiege, welche mit dem Referenzkorpus aus t_0 erzeugt wurden, sehen folgendermaßen aus:

Term	H_{term}	H_{term_hoch}	a_{term}
W_0	1550	297600	1,488
W_1	1200	230400	1,152
W_2	600	115200	0,576
W_3	600	115200	0,576
W_4	1050	201600	1,008

Tabelle 5: Absolute Häufigkeiten im Analysekorpus zum Zeitpunkt t_1

In diesem Analysekorpus ist es der Term W_0 der über der Termanstiegsschwelle von 1,3 liegt. Doch wurde hier der veraltete Referenzkorpus aus t_0 zur Berechnung verwendet.
Wird der Referenzkorpus für den Zeitpunkt t_1 errechnet, ergibt sich folgende Verteilung der Terme im Referenzkorpus, der wieder 1 Mio. Nachrichten mit je einem Term beinhaltet.

Term	H_{term}	h_{term}
W_0	300000	0,3
W_1	200000	0,2
W_2	50000	0,05
W_3	250000	0,25
W_4	200000	0,2

Tabelle 6: Absolute Häufigkeiten im Referenzkorpus zum Zeitpunkt t_1

Wird jetzt noch einmal die Ereignisdetektionsberechnung für den Zeitpunkt t_1 mit diesem aktuellem Referenzkorpuswerten durchgerechnet, ergeben sich folgende Werte.

Term	H_{term}	H_{term_hoch}	a_{term}
W_0	1550	297600	0,992
W_1	1200	230400	1,152
W_2	600	115200	2,304
W_3	600	115200	0,4608
W_4	1050	201600	1,008

Tabelle 7: Neubetrachtung des Analysekorpus mit aktuellen Referenzkorpusdaten für den Zeitpunkt t_1

Das Ergebnis ist nun ein völlig anderes und hat sich sogar in das Gegenteil verkehrt. Der Term, der über der Termanstiegsschwelle liegt, ist nun W_2 und nicht W_0. Der Term, der absolut am wenigstens im Analysekorpus auftrat, taucht aber im Verhältnis zum aktualisierten Referenzkorpus, am Häufigsten auf. Ist der Referenzkorpus signifikant veraltet, so spiegelt er nicht mehr die Referenz wieder. Es ist natürlich, dass sich Termkonzentrationen im Laufe der Zeit ändern z.B. wird im Laufe der Zeit in den Nachrichten über andere Personen berichtet, weil sie neue Ämter begleiten, oder Themen über die vermehrt berichtet wurde, sind nicht mehr relevant. Diese Änderungen sollte mit adaptiven Referenzkorpora Rechnung getragen werden, da veraltete Referenzkorpora zu einem falschen Analyseergebnis führen können und somit neue Ereignisse nicht mehr korrekt detektiert werden können.
Um die Überlegungen mit realen Daten zu verifizieren, wurde ein beliebiger Zeitpunkt der Ereignisdetektion zusätzlich mit einem zwei Jahre alten Referenzkorpus nachgerechnet.

Ergebnisse mit aktuellem Referenzkorpus	Ergebnisse mit zwei Jahre altem Referenzkorpus
morning	#tweetmyjobs
saturday	#job
	morning
	saturday
	weekend

Tabelle 8: Ereignisdetektionsergebnisse von New York am 25.10.2014 18 Uhr

In den obigen Tabelle 8 der Ereignisdetektion erkennt man die Unterschiede. Es wurde bei beidem Analysen die Terme „morning“ und „saturday“ herausgefiltert als signifikant erhöht aber bei der Analyse mit dem veralteten Referenzkorpus wird zusätzlich ein neuer Ereignis-Cluster erkannt und zwar mit den Tags „#tweetmyjobs“ und „#job“. Dabei ist anzumerken, dass mit diesen Tags die Nutzer bekannt geben, welchen Job sie

gerade haben. Dies war vor zwei Jahren noch nicht so verbreitet wie heute. Die veränderte Nutzung dieser Tags ist in dem aktuellen Referenzkorpus enthalten und somit wird das Mitteilen des eigenen Jobs nicht mehr als neues Ereignis detektiert, da es inzwischen normal geworden ist. Noch deutlicher wird es bei dem nächsten Beispiel.

Ergebnis mit aktuellem Referenzkorpus	**Ergebnis mit zwei Jahre altem Referenzkorpus**
#empowerri	#veteranjob
#goodmorning	#transportation
Gameday	#retail
Ham	#nursing
5k	#sales
Lastnight	#healthcare
Hungover	#job
Breakfast	#goodmorning
Waking	#tweetmyjobs
Goodmorning	#jobs
	#empowerri
	soliant
	technician
	5k
	ham
	sales
	bae
	lastnight
	hungover
	tbh
	ebola
	aerotek

Tabelle 9: Ereignisdetektion (detektierte Terme) vom 25.10.2014 15:20 Uhr mit aktuellem bzw. zwei Jahre altem Referenzkorpus

In Tabelle 9 sieht man die Analyse mit aktuellem und veraltetem Referenzkorpus. Die Ergebnismenge der Analyse mit einem veralteten Referenzkorpus ist sehr viel größer. Z.B. wird dort auch der Term „ebola“ mit detektiert, aber in der Analyse mit aktuellem Referenzkorpus nicht. Dies liegt daran, dass mehr über Ebola im Oktober 2014 geschrieben wurde als im Oktober im Jahr 2012, da die Krankheit in einigen afrikanischen Ländern ausgebrochen war und erste Fälle in den USA aufgetreten sind. D.h. das System hat das erhöhte Auftreten des Terms „Ebola“ erkannt und es ist nun normal und stellt zu diesem Zeitpunkt kein akut auftretendes neues Ereignis dar.
Somit konnte an diesen Beispielen gezeigt werden, dass die Aktualität des Referenzkorpus einen großen Einfluss auf die Ergebnisse der Ereignisdetektion hat. Nur mit einem aktuellen Referenzkorpus ist es möglich zu entscheiden, ob eine bestimmte Termkonzentration zu diesem Zeitpunkt normal oder schon signifikant erhöht ist. Da sich die Nachrichten- bzw. Themenlage ständig ändert, sollte hier die Referenz ständig adaptiert werden, um verlässlich aktuelle Ereignisse zu detektieren.

5.1.1.2 Verifikation der zweiten Hypothese

Wie bereits in Kapitel 3.3.2 gezeigt wurde, kann die Termkonzentration von bestimmten Termen in Abhängigkeit des Ortes sehr unterschiedlich sein. In dem Beispiel mit realen Daten in Kapitel 3.3.2 konnte bereits exemplarisch gezeigt werden, dass eine gleiche absolute Zunahme des Auftretens eines Terms an einem Ort überhaupt keine Relevanz haben kann und an einem anderen Ort, wo der Term gewöhnlich nicht so häufig auftritt, eine signifikante Steigerung der Termkonzentration für diesen Ort darstellt.

Mit Hilfe von ortsabhängigen Referenz- bzw. Analysekorpora sollen deshalb die Analyseergebnisse der Differenzanalyse, mit adaptiven Referenzkorpus, weiter verbessert werden. Eine Verbesserung der Ergebnisse kann sich hier auf zwei Arten auswirken. Zum einen wird die Qualität der Terme der Ergebnismenge verbessert werden. D.h. die Anzahl der fehldetektierten Terme welche das gefundene Ereignis nicht genauer beschreiben ist geringer. Zum anderen können erst dadurch Ereignisse detektiert werden die ohne die Betrachtung des Ortes nicht entdeckt geworden wären.

Diese Effekte sollen nun mit Daten aus Ereignisdetektionen gezeigt werden. Um die nächsten Schritte besser zu verstehen, hier noch einmal eine Kurzfassung der Analyseschritte. Während der Ereignisdetektion wird jeder Tweet des Analysekorpus überprüft, ob an der Stelle, von wo der Tweet gesendet wurde, die Termkonzentration des entsprechenden Terms signifikant erhöht ist. Dabei wird die Termkonzentration des Terms für genau die Stelle des Tweets mit Hilfe der Daten aus dem Referenzkorpus und einmal aus den Daten des Analysekorpus ermittelt. Die Termkonzentrationen werden dann miteinander verglichen. Es wird also immer exakt die Termkonzentration für den jeweiligen Ort des zu untersuchenden Tweets errechnet.

Für das nachfolgende Experiment sollen die Ergebnisse einer Ereignisdetektion mit Ortsbetrachtung mit den Ergebnissen ohne Ortsbetrachtung verglichen werden. D.h. es erfolgt eine normale Differenzanalyse mit den Daten statt (incl. adaptiven Referenzkorpus). Dazu wurden die Ergebnisse für die Region New York vom 05.02.2016 18:00 Uhr genutzt. An diesem Tag gab es in New York wieder einen Wintereinbruch und es schneite wieder.

In Tabelle 10 kann man deutlich erkennen, dass das Ereignis, der Wintereinbruch, erkannt wurde. In der Ereignisdetektion, die zur gleichen Zeit und mit denselben Daten stattfand, aber den Ort nicht betrachtet, sieht man viel mehr Terme in der Ergebnismenge. Auch hier lässt sich das Ereignis in den Termen ablesen doch treten hier viel mehr Terme auf, die keinen Bezug zum Ereignis erkennen lassen und hier nicht relevant erscheinen.

Die Analyse ohne die Betrachtung des Ortes liefert somit eine hohe Anzahl von Fehlalarmen. Die Analyse mit ortsabhängigen Korpora hat hier also die Qualität der Ergebnismenge verbessert.

Ergebnisse der Ereignisdetektion mit Ortsbetrachtung	**Ergebnisse der Ereignisdetektion ohne Ortsbetrachtung**
#snowday	#friday
#winter	#snow
#snow	#sbvote
friday	#nationalweatherpersonsday
winter	#dialysis
snowy	#tgif
snow	#winterwonderland
weekend	#broncos
snowing	#winter
crane	#white
	#snowing
	#worldnutelladay
	#longisland
	#keeppounding
	#snowday
	johnny
	fridays
	snowing
	crane
	taking
	trees
	collapse
	injured
	panthers
	snows
	snowy
	manziel
	i'm
	crush
	super
	wonderland
	reports
	orange
	nutella
	bowl
	construction

Tabelle 10: Ergebnisse der Ereignisdetektion einmal mit Ortsbetrachtung und einmal ohne vom 05.02.2016 18:00 Uhr in der Region New York

Im nächsten Analyse-Beispiel, in Tabelle 11 und Abbildung 50, wird ersichtlich wie durch eine ortsabhängige Differenzanalyse es erst möglich wurde ein Ereignis zu erkennen. Als Beispiel dient hier die Ereignisdetektion aus der Region Deutschland vom 17.02.2016 um 13 Uhr.

Ergebnisse der Ereignisdetektion mit Ortsbetrachtung	**Ergebnisse der Ereignisdetektion ohne Ortsbetrachtung**
rheinland-pfalz	kalt
	kritik

Tabelle 11: Ergebnisse der Ereignisdetektion einmal mit Ortsbetrachtung und einmal ohne vom 17.02.2016 13 Uhr in der Region Deutschland

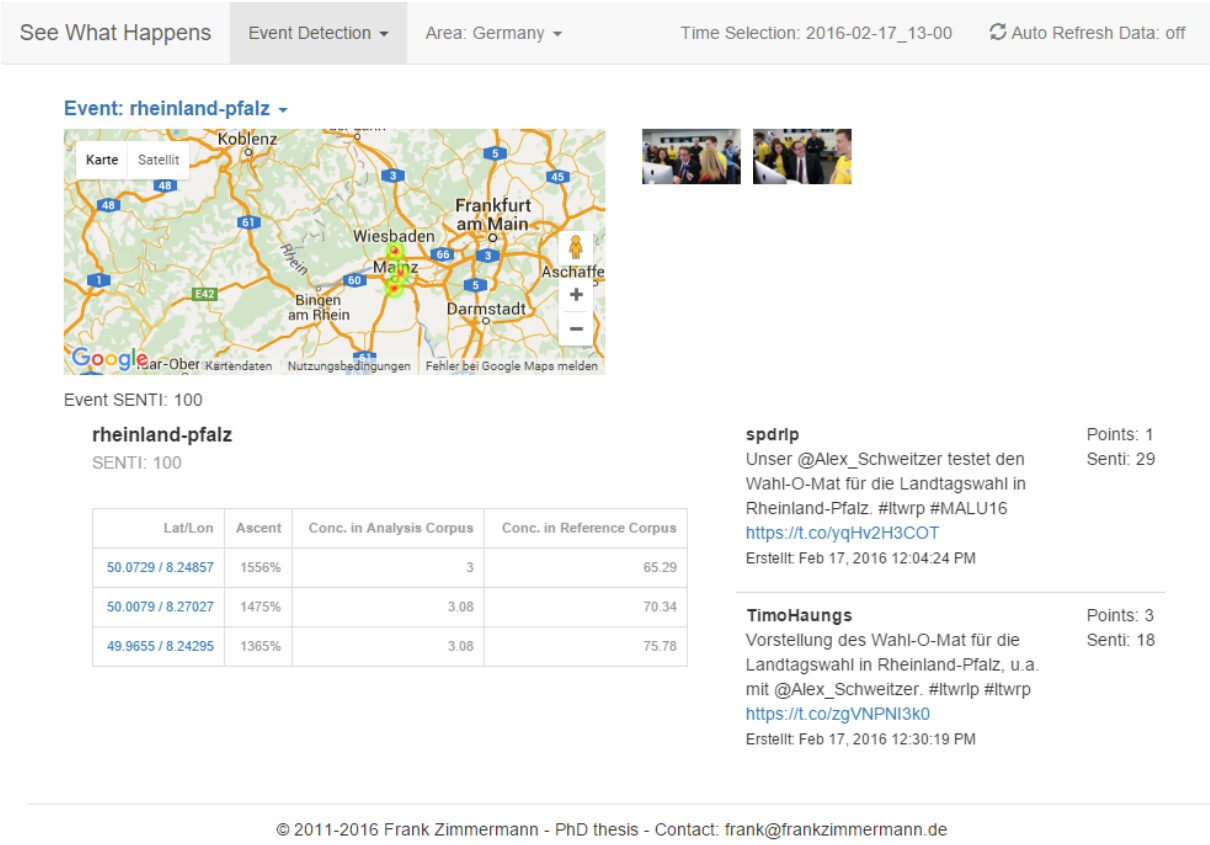

Abbildung 50: SWH Ergebnisseite zur Analyse vom 17.02.2016 13 Uhr in der Region Deutschland

Um den detektierten Term „rheinland-pfalz“ als Ereignis zu verstehen, muss man die extrahierten Bilder und Tweets sich anschauen. Es handelt sich hier um ein lokal begrenztes Ereignis. Es wurde auf einer Veranstaltung die aktuelle Version des Wahl-O-Mat für die kommende Landtagswahl in Rheinland-Pfalz vorgestellt. In der Analyse ohne Ortsbetrachtung taucht dieses lokale Ereignis dagegen überhaupt nicht auf. Somit wurde es durch die Ortsbetrachtung es erst ermöglicht dieses Ereignis überhaupt zu detektieren.

Will man eine gut funktionierende Ereignisdetektion haben, so ist es notwendig zum einen mit einem aktuellem Referenzkorpus zu arbeiten, damit die aktuellen Termkonzentrationen bekannt sind. Weiterhin ist es notwendig bei einer genauen ortsbasierten Analyse, wie sie hier durchgeführt wird, jeweils die entsprechenden Korpora für den jeweiligen Ort zu bilden und mit diesen zu arbeiten. Dadurch wird zum

einen die Qualität der Ergebnismenge verbessert bzw. es können erst dadurch lokale Ereignisse erkannt werden die ohne Ortsbetrachtung sonst nicht entdeckt werden würden.

5.1.2 Betrachtung der Ergebnisse der Ereignisdetektion

Im Information Retrieval Bereich wird die Qualität der Ergebnisse einer Recherche mit Hilfe von bestimmten Werten ausgedrückt wie z.B. Recall (Trefferquote), Precision (Genauigkeit) und Fallout (Ausfallquote) [123]. Der Recall-Wert ist die Wahrscheinlichkeit mit der ein relevantes Dokument gefunden wird. Der Precision-Wert sagt dagegen aus wie hoch die Wahrscheinlichkeit ist, dass ein gefundenes Dokument auch relevant ist und der Fallout-Wert gibt an, mit welcher Wahrscheinlichkeit ein irrelevantes Dokument gefunden wird [123]. Die beschriebenen Werte werden folgendermaßen ermittelt:

$$P = \frac{TP}{TP + FP}$$

Formel 18: Berechnung des Precision-Wertes [32]

$$R = \frac{TP}{TP + FN}$$

Formel 19: Berechnung des Recall-Wertes [32]

$$F = \frac{2 * P * R}{P + R}$$

Formel 20: Berechnung des Fallout-Wertes [32]

Wobei TP für „True Positive“ steht, FP für „False Positive“ und FN für „False Negative“. Diese Beurteilung von Algorithmen über die vorgestellten Werte funktioniert aber nur, wenn man das Ergebnis vorher kennt. Dies ist der Fall wenn die Algorithmen z.B. einen Benchmark-Datensatz analysieren, bei dem die Ergebnisse der Analyse schon bekannt sind. Somit wird es möglich die Algorithmen miteinander zu vergleichen. Z.B. wird dies bei Algorithmen gemacht, die Zeitangaben aus Texten extrahieren wie in [31] [32] [33].
Für diese Ereignisdetektion gibt es keinen expliziten Benchmark-Datensatz bzw. es wurde kein Datensatz erstellt, da die Erstellung solch eines Benchmark-Datensatzes auch mit einem gewissen Aufwand einhergeht. Auch wurde die Analyse nicht darauf angepasst einen bestehenden Benchmark-Datensatz zu verarbeiten, da die Tweets in solchen Datensätzen (soweit bekannt) keine Tweets mit Ortsangaben sind.
Somit sollen die Ergebnisse der Ereignisdetektion allgemein betrachtet werden und ein Augenmerk auf die Geschwindigkeit der Detektion gelegt werden, da diese Maßzahl sehr gut z.B. mit etablierten Medien verglichen werden kann.
Die Entwicklung der SWH-Software begann im Frühjahr 2011 und lieferte im Sommer 2011 die ersten Ergebnisse in Form von detektierten Ereignissen. Zuerst nur auf das Gebiet von Deutschland limitiert, kamen später, im April 2012, auch die Gebiete in den USA dazu. Während dieser Zeit wurde die Software weiter entwickelt bis zu diesem jetzigem Stand, der im vorherigen Kapitel 4 beschrieben wurde. Die Ergebnisse der SWH-Software wurden somit über einen längeren Zeitraum beobachtet.
Die erkannten Ereignisse beschreiben, was aus Sicht der Nutzer gerade passiert. Diese Ereignisse sind nicht immer vergleichbar mit den Nachrichten der etablierten Medien.

So findet man z.B. politische Ereignisse, worüber z.B. etablierte Medien berichten, seltener in der Ergebnismenge. Was wohl daran liegt, dass von diesen Ereignissen nicht genügend Tweets mit Ortsinformationen vorliegen. Dies könnte bedeuten das diese Themen nicht so häufig mobil getwittert werden und man so wenige Tweets mit Positionsdaten zur Verfügung hat. Sportereignisse findet man dagegen bei den detektierten Ereignissen der SWH-Software häufiger, da sich hier mehr Tweets finden lassen von Zuschauern, die von diesen Ereignissen vor Ort berichten. Über die detektierten Ereignisse bekommt man sogar mehr Informationen z.B. die Stimmung vor dem eigentlichen Sportereignis, aktuelle Fotos aus Sicht der Zuschauer vor Ort und viele Details mehr, die man in einem Zeitungsartikel i.d.R. nicht wiederfinden würde.
Die detektierten Ereignisse der SWH-Software beschreiben momentan immer was gerade an einem bestimmten Ort geschieht.
In Abbildung 51 sieht man die unterschiedlichen Sportereignisse, die parallel stattfanden und getrennt als eigenständige Ereignisse an diesem Samstag im Dezember detektiert wurden. Neben den Sportereignissen fanden auch andere Ereignisse statt, die die Menschen beschäftigten oder woran sie teilnahmen, hier in diesem Beispiel die Weihnachtsmärkte und eine Partei-Veranstaltung (buko13). Auch so allgemeine Ereignisse wie etwa Feiertage werden durch die Analyse erkannt, wie in der nachfolgenden Abbildung 51 zu sehen ist.
Neben der Erkennung des eigentlichen Ereignisses „Ostern“, in Abbildung 52 zu sehen, wurden auch dazu passende Tweets und Bilder ausgewählt, die das Ereignis näher beschreiben sollen. Auch wenn das Ereignis „Ostern“ sehr trivial ist, sieht man doch wie gut die Beispiel-Tweets und die dazu ausgewählten Bilder passen. Nicht nur in dem Gebiet Deutschland wurde z.B. das Oster-Ereignis registriert, sondern auch in den Gebieten der USA wurde zur gleichen Zeit das Ereignis erkannt.

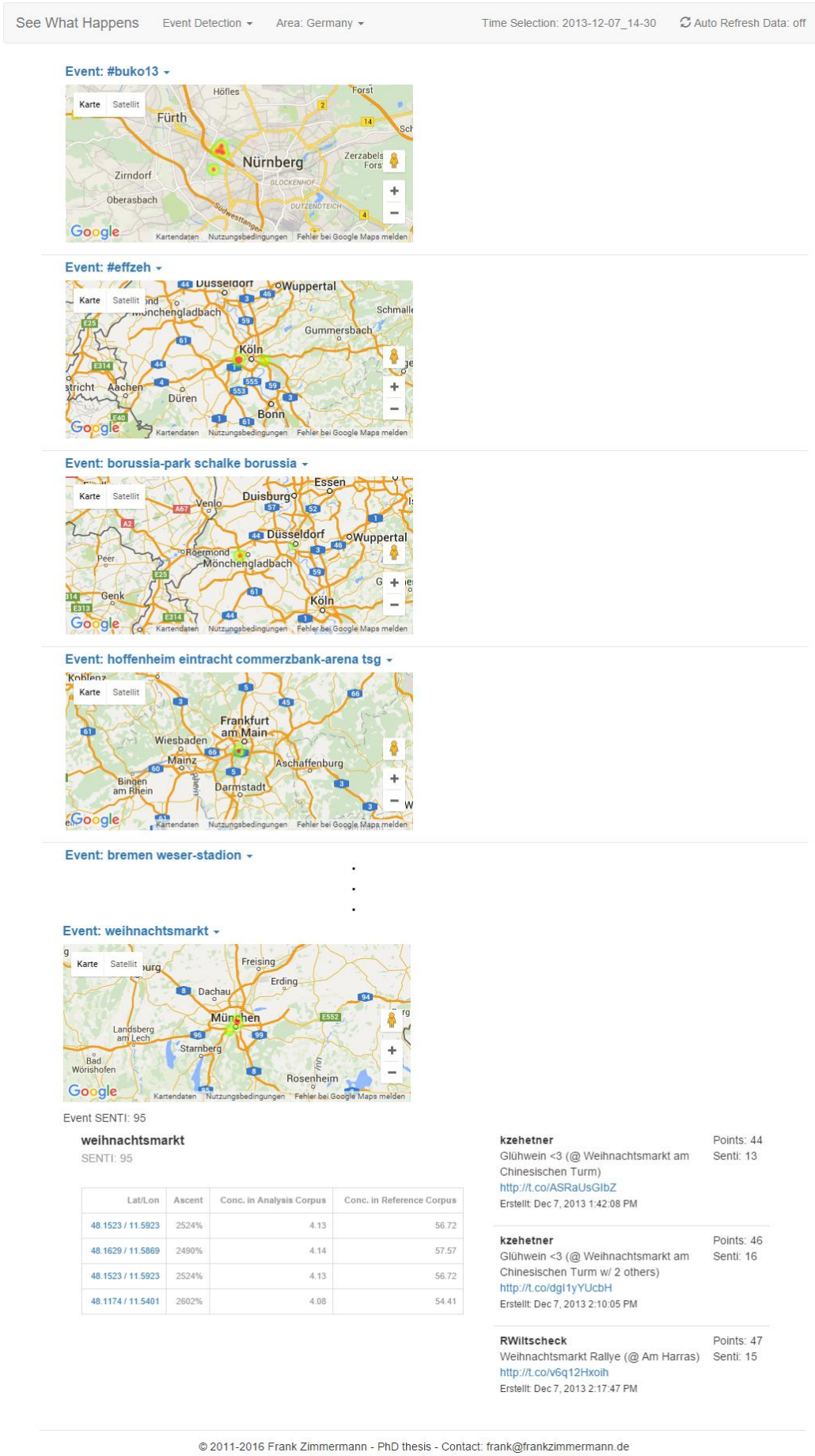

Abbildung 51: Beispielhafte detektierte Ereignisse aus Sport und Kultur am 7. Dezember 2013 in Deutschland.

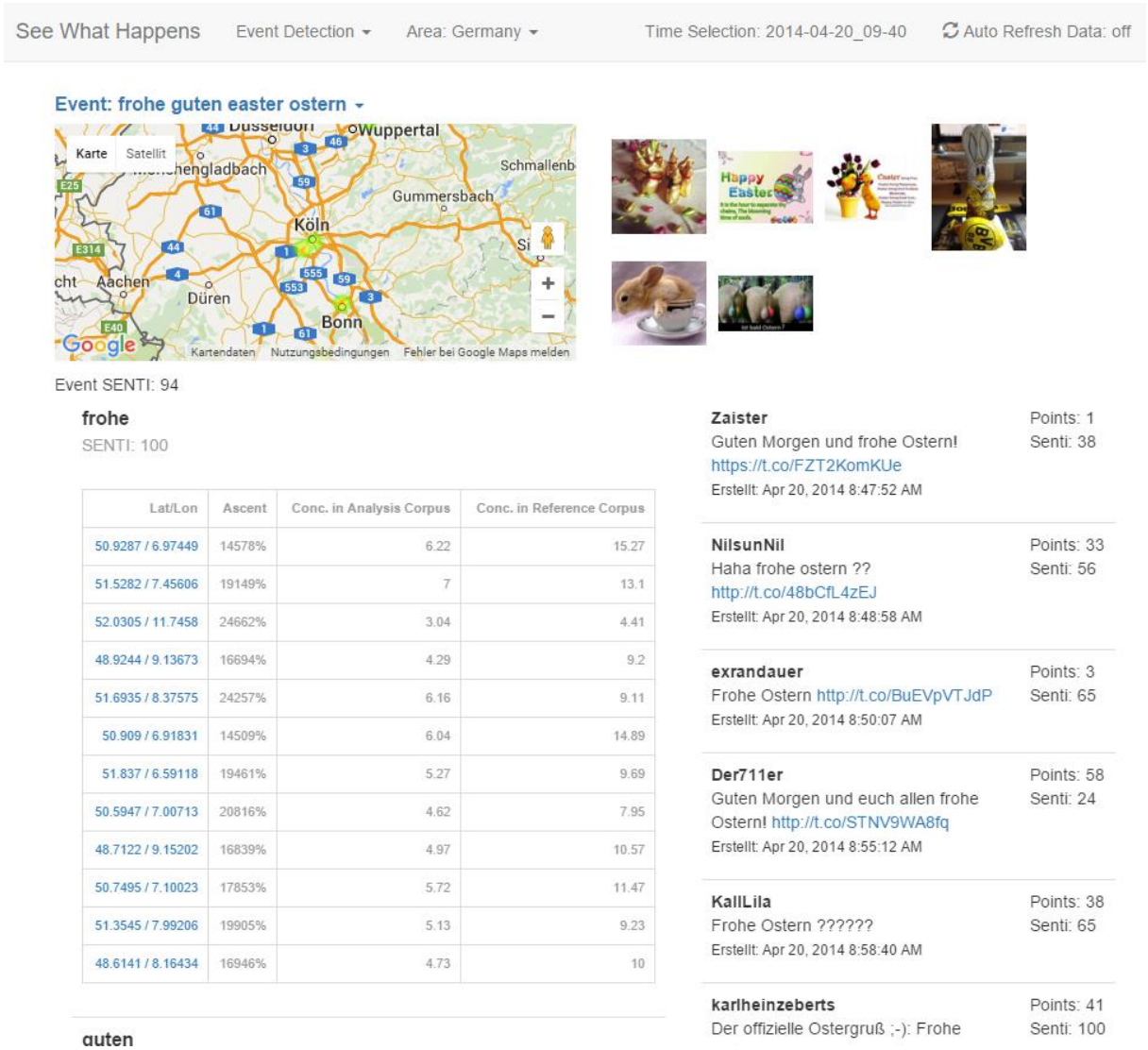

Abbildung 52: Erkennung des Ereignisses "Ostern" in der Analyse mit Präsentation dazu passender Bilder und Tweets.

Neben den eher positiven Ereignissen wurden natürlich auch negative Ereignisse wie z.B. Katastrophen detektiert. Z.B. am 12.03.2014 als in New York ein Gebäude explodierte [124] wurde dieses Ereignis als ein signifikantes Ereignis detektiert und die Nutzer der SWH-Software per E-Mail informiert. Allein die detektierten, signifikant in ihrer Häufigkeit auftretenden Terme, beschreiben das Ereignis schon sehr gut (siehe Abbildung 53).

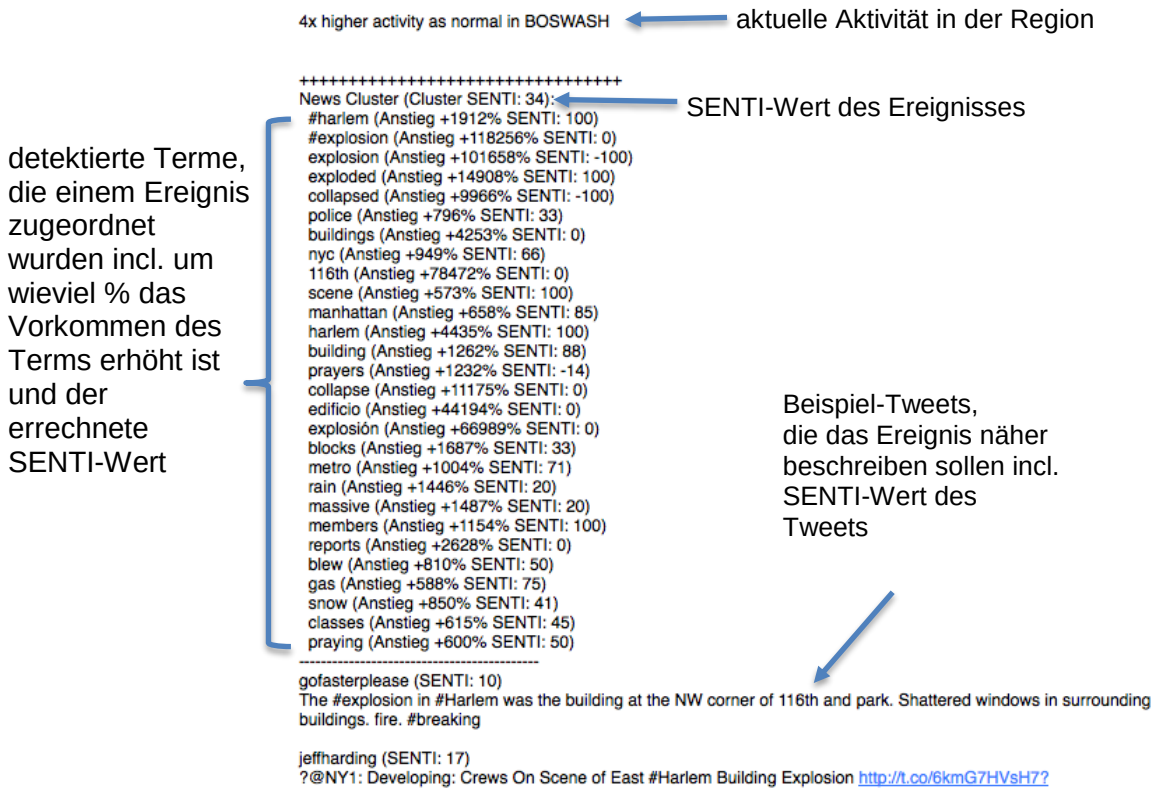

Abbildung 53: SWH-E-Mail, die auf ein signifikantes Ereignis hinweist, hier die Explosion eines Gebäudes in Harlem am 12.03.2014.

Am selben Tag ereignete sich in San Francisco ein Großbrand [125]. Auch dieses Ereignis wurde durch die SWH-Software detektiert und die dazu relevanten Tweets und Bilder extrahiert. Dieses Ereignis ist in Abbildung 54 zu sehen. Während das Ereignis stattfand, hat die SWH-Software sofort eine große Anzahl von Bildern des Ereignisses extrahieren können. Man konnte das Feuer von den verschiedensten Blickwinkeln und Entfernungen sehen und sich so sehr gut selbst ein Bild von der Lage vor Ort machen. In diesem Augenblick hatte man durch die SWH-Analyse mehr Bildmaterial zur Verfügung als das man auf den News-Seiten finden konnte. Die zusätzlichen extrahierten Tweets konnten die Situation vor Ort weiterhin gut beschreiben.

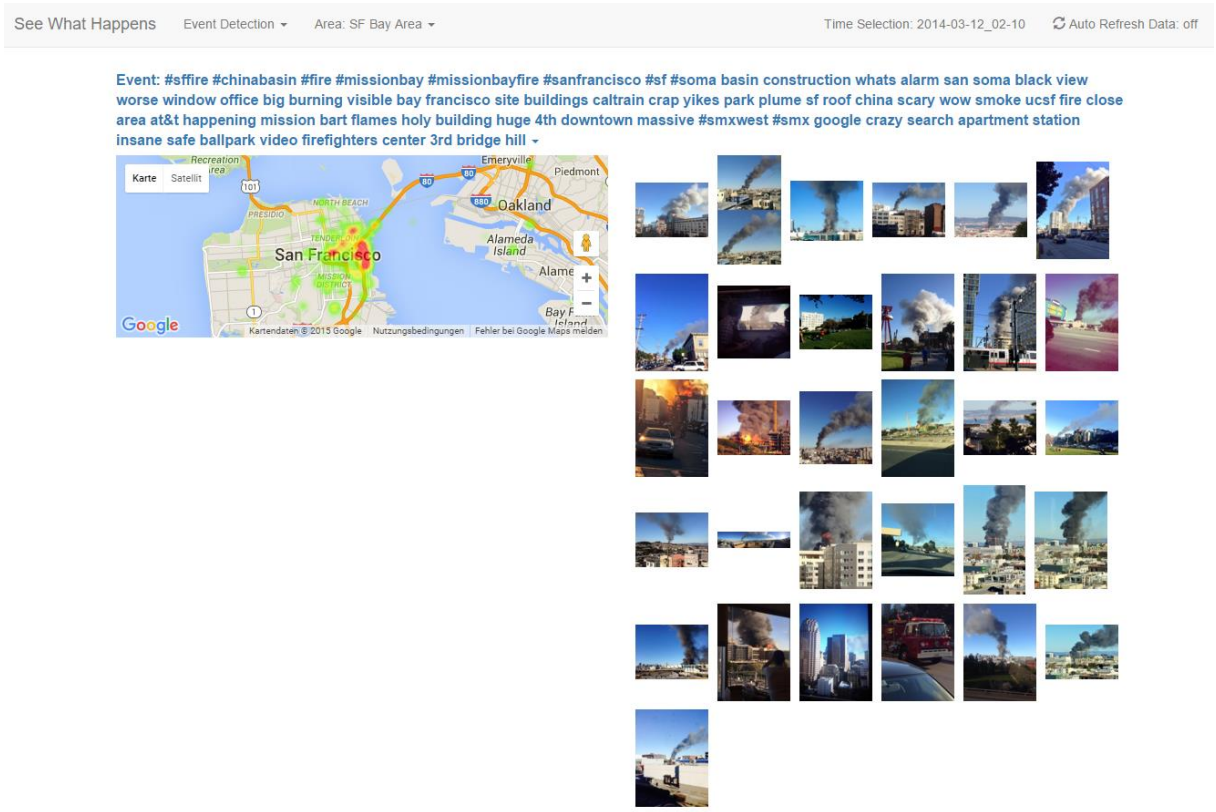

Abbildung 54: Von der SWH-Software detektierter Großbrand in San Francisco am 12.03.2014.

Auch Naturkatastrophen wie z.B. Erdbeben konnte die SWH-Software detektieren.

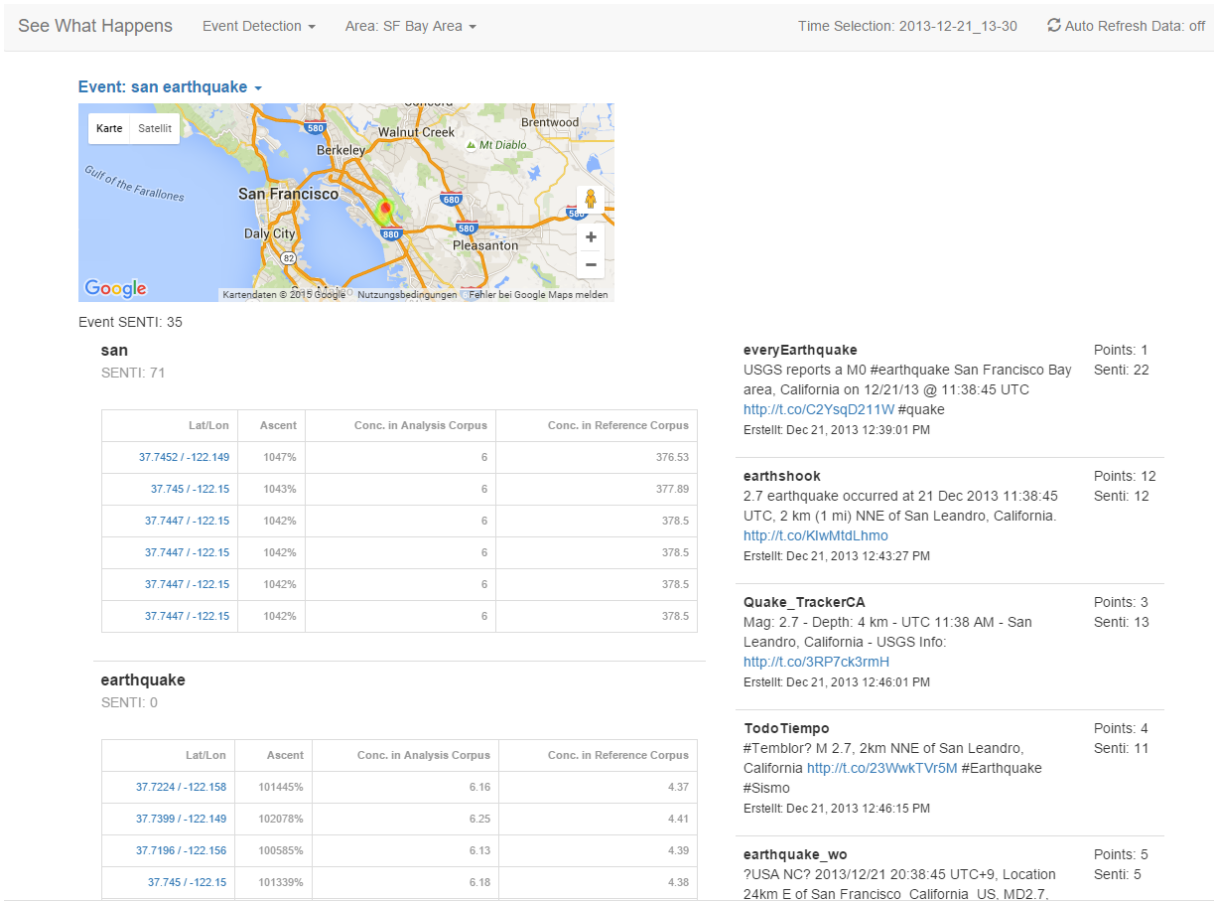

Abbildung 55: Detektiertes Erdbeben am 21.12.2013 im Gebiet der Bay Area.

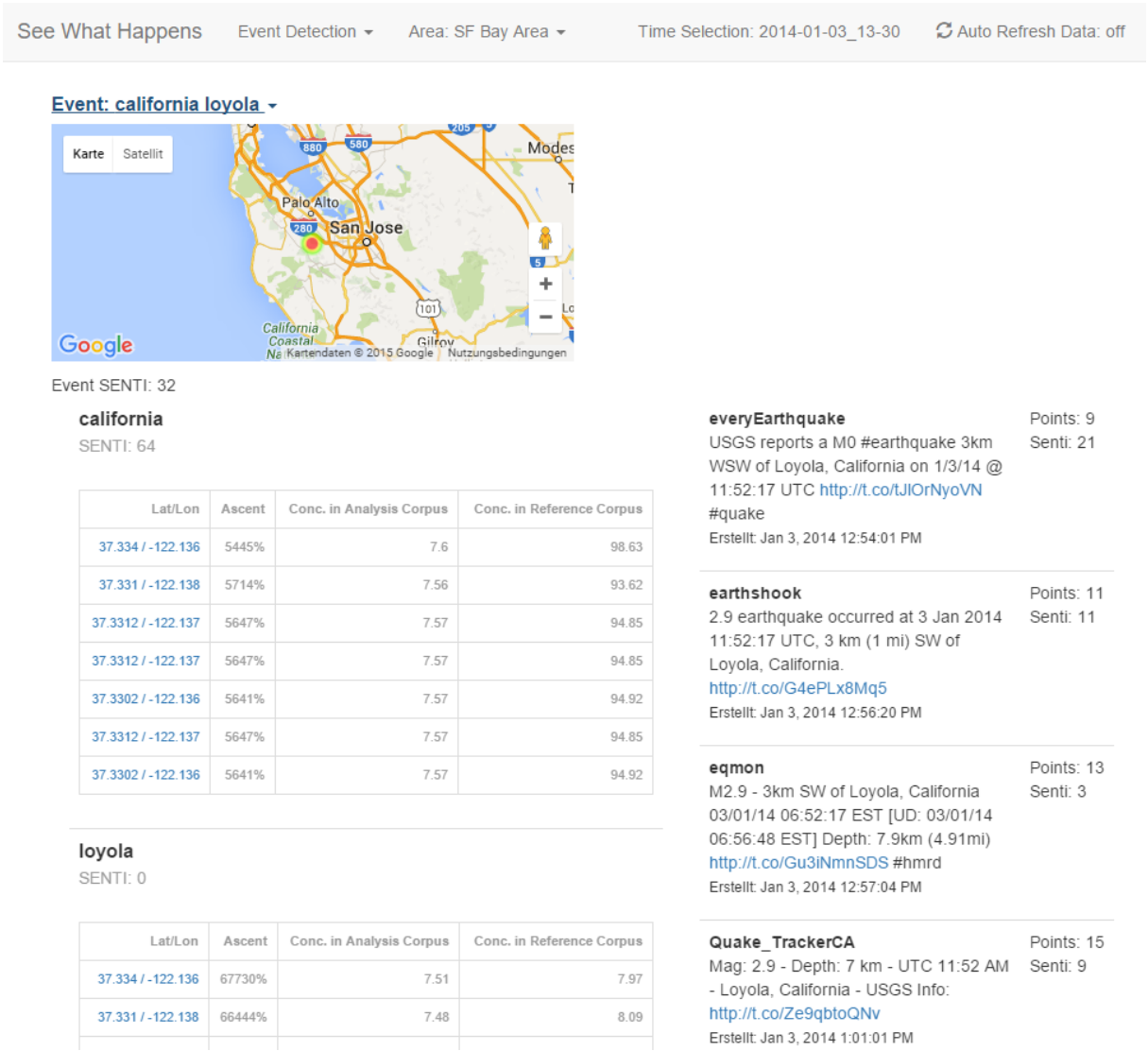

Abbildung 56: Ein weiteres detektiertes Erdbeben am 03.01.2014 ebenfalls in der Region der Bay Area.

Die Abbildung 55 und die Abbildung 56 zeigen jeweils das Ergebnis dieser Erdbebendetektion. Waren bei den Detektionssystemen in [43], [126] und in [41] noch spezifische Systeme notwendig, die gezielt den Datenstrom nach bestimmten Termen durchsuchten, um Brände oder Erdbeben frühzeitig zu erkennen, so erkennt die SWH-Software diese Ereignisse automatisch und sprachunabhängig. Dies ist auch damit begründet das Messstellen, die Erdbeben detektieren, ihre Ergebnisse automatisch selbst über Twitter veröffentlichen. Diese Tweets von den Sensoren und die Tweets der Nutzer stellen erst die Datengrundlage dar aus denen die Ereignisdetektion das Ereignis so schnell erkennen kann.
Die bisher vorgestellten detektierten Ereignisse sollten einen kurzen Überblick geben, welche verschiedenen Ereignisse mit dem SWH-System detektiert werden konnten. Ganz ausgelassen wurde z.B. die Detektion von Konferenzen, Messen oder anderen Veranstaltungen. Auch diverse Sportereignisse sowohl in Deutschland als auch in den untersuchten Gebieten in den USA werden gleichermaßen erkannt wie z.B. (Halb-) Marathon-Läufe, Baseball- oder Football-Spiele usw. Die zusätzlich getrennte Untersuchung in New York mit einem kleineren Effektradius führt dazu, dass dort noch detaillierter Ereignisse detektiert werden können. So kann man z.B. in New York mit niedrigerer Schwelle und Effektradius arbeiten, um noch regional kleinere Ereignisse zu detektieren.

5.1.3 Geschwindigkeit der Ereignisdetektion

Ein weiteres Ziel der SWH-Software bzw. der konzipierten Algorithmen ist es, die Ereignisse so schnell wie möglich zu detektieren um ggf. schneller als etablierte Medien zu sein. Der zeitliche Vorsprung den man durch die Detektion der Ereignisse im Twitter-Datenstrom erreichen kann, soll vollständig an den Nutzer weitergegeben werden.
Im nachfolgenden sollen dazu einige signifikante Ereignisse näher untersucht werden, um zu ermitteln, zu welchen Zeitpunkten die SWH-Software das Ereignis das erste Mal detektiert hat und wann parallel dazu die etablierten Medien z.B. lokale sowie überregionale News-Seiten das erste Mal darüber berichtet haben. Für diese zeitlichen Betrachtungen wurden gezielt katastrophale Ereignisse ausgewählt, um sicher zu gehen, dass diese Art der Nachrichten von den Medien sofort veröffentlicht werden, um die Geschwindigkeit der Ereignisdetektion der etablierten Medien gut bestimmen zu können. Die untersuchten Ereignisse mussten zudem alle in den untersuchten Regionen der SWH-Software stattgefunden haben. Die Ereignisse werden in chronologischer Reihenfolge betrachtet. Alle nachfolgenden Zeitangaben beziehen sich immer auf die Mitteleuropäische Zeit.

5.1.3.1 Newtown Amoklauf

Der Amoklauf an der Sandy Hook Elementary School in Newtown (Connecticut) am 14.12.2012 war eines der schlimmsten Amokläufe an einer Schule in den USA. Der Täter erschoss 28 Menschen, darunter 20 Kinder. [127] Der recherchierte zeitliche ist in Abbildung 57 dargestellt.
Die Presse (NBCConnecticut) wurde direkt, über Twitter, während der Tatzeit über den Vorfall durch einen Nachbarn, der Schüsse hörte, informiert. Zur gleichen Zeit ging auch der erste Notruf bei der örtlichen Polizei ein. Ca. 15 Minuten später tweetete NBCConnecticut, die nun bereits vorgewarnt war, dass die State Police auf den Weg zur Schule sei. Eine weitere viertel Stunde später erst, retweetete die Nachrichtenagentur Reuters einen weiteren Tweet von NBCConnecticut, dass die State Police eine Schießerei an der Newtown Elementary School untersucht. Selbst hatte Reuters bis zu diesem Zeitpunkt noch keine eigene Meldung zu diesem Vorfall veröffentlicht. Eine weitere viertel Stunde später, also 16:30 Uhr detektiert die SWH-Software das Ereignis „shooting“ in Newtown. Ab diesem Zeitpunkt wurde dieses Ereignis von der SWH-Software erkannt und war über die SWH-Website sichtbar. Erst über eine weitere halbe Stunde später berichtete erstmals Reuters über diesen Vorfall mit einem Breaking News Tweet. Danach dauerte es über eine Stunde bis die hiesigen großen Nachrichtenseiten über dieses Ereignis berichten. In dieser Zeit hat die SWH-Software dieses Ereignis bereits als ein sehr signifikantes Ereignis erkannt (vgl. Kapitel 3.3.5.2) und informiert die Nutzer der Software per Direct Message über Twitter und per E-Mail über die stark erhöhte Aktivität und beschreibt das Ereignis.
Da NBCConnecticut schon während der Tatzeit von einem Nachbarn informiert wurde, konnte der Nachrichtensender potentiell frühzeitig berichten doch hat man in der zeitlichen Abfolge gesehen, dass zwischen dem ersten Hinweis und der ersten Meldung eine rel. große Zeitspanne vergangen ist. Als die SWH-Software das Ereignis erkannte, hatte Reuters selbst noch keine Meldung veröffentlicht und nur Meldungen von NBCConnecticut retweeten. Das Ereignis und die Bewertung, dass dieses Ereignis ein signifikantes Großereignis ist, konnte die SWH-Software somit sehr früh erkennen.

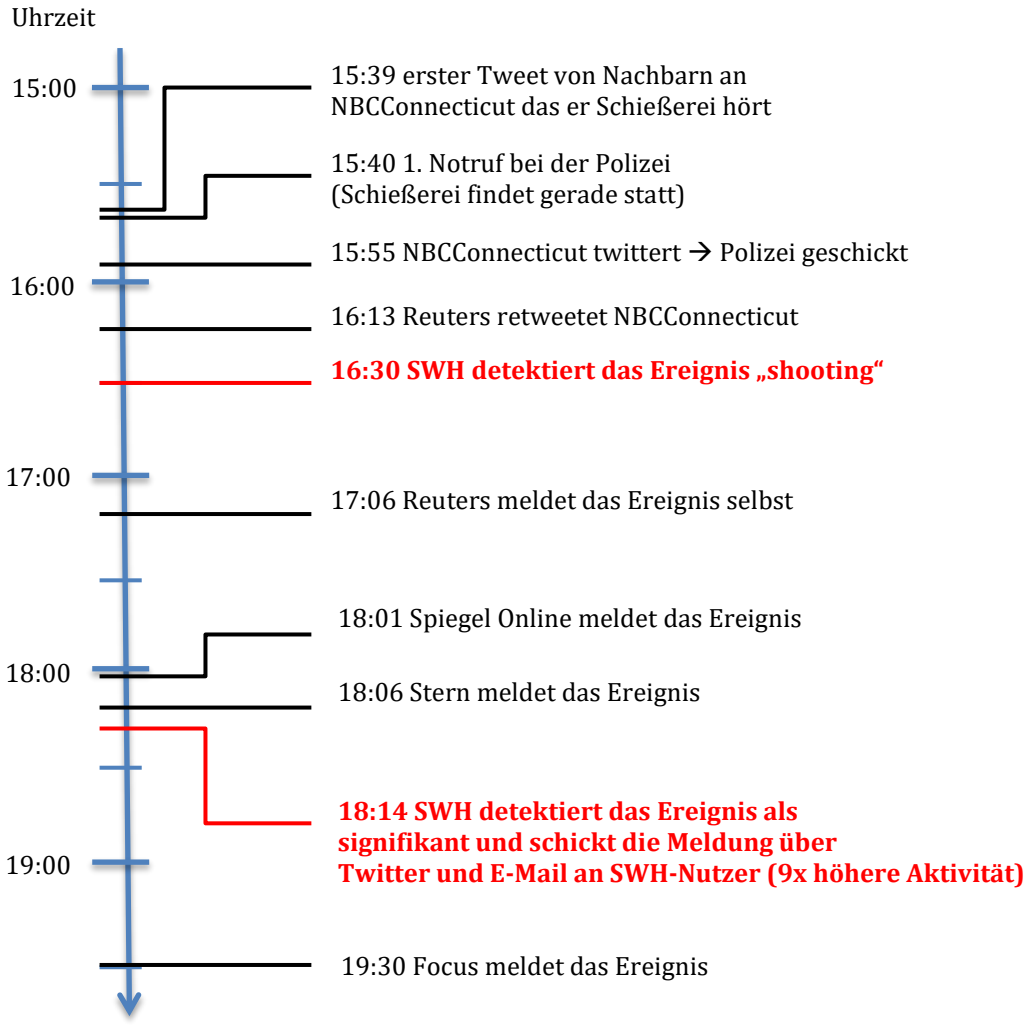

Abbildung 57: Zeitlicher Ablauf des Ereignisses und wer wann was berichtete. Quellenangabe zu den Zeitpunkten: 15:39 [128], 15:40 (aus div. Nachrichtentexten), 15:55 [129], 16:13 [130], 17:06 [131], 18:01 [132], 18:06 [133], 19:30 [134]

5.1.3.2 Boston Marathon Anschlag

Der Boston-Marathon Anschlag fand am 15.04.2013 statt. Auf der Zielgeraden explodierten, im kurzen Abstand, zwei Bomben die drei Menschen tötete und 264 Menschen verletzte. [135] Der recherchierte zeitliche ist in Abbildung 58 dargestellt.

Der Boston-Marathon an sich ist eine sehr populäre Sport-Veranstaltung mit der längsten Tradition nach dem Olympischen Spielen und findet immer am jährlichen Patriots‘ Day in Boston statt. [136] Durch diese hohe Popularität und der Größe der Veranstaltung, war schon zuvor sehr viel Presse vor Ort und berichtete von dem Marathon-Ereignis.

Die SWH-Software hatte den Boston-Marathon schon den ganzen Tag als Ereignis erkannt und auf der SWH-Website als Ereignis präsentiert. Das Ereignis des Anschlages konnte aber auch hier sehr schnell detektiert werden. Zwischen der Explosion und der Detektion vergingen nur 10 Minuten.

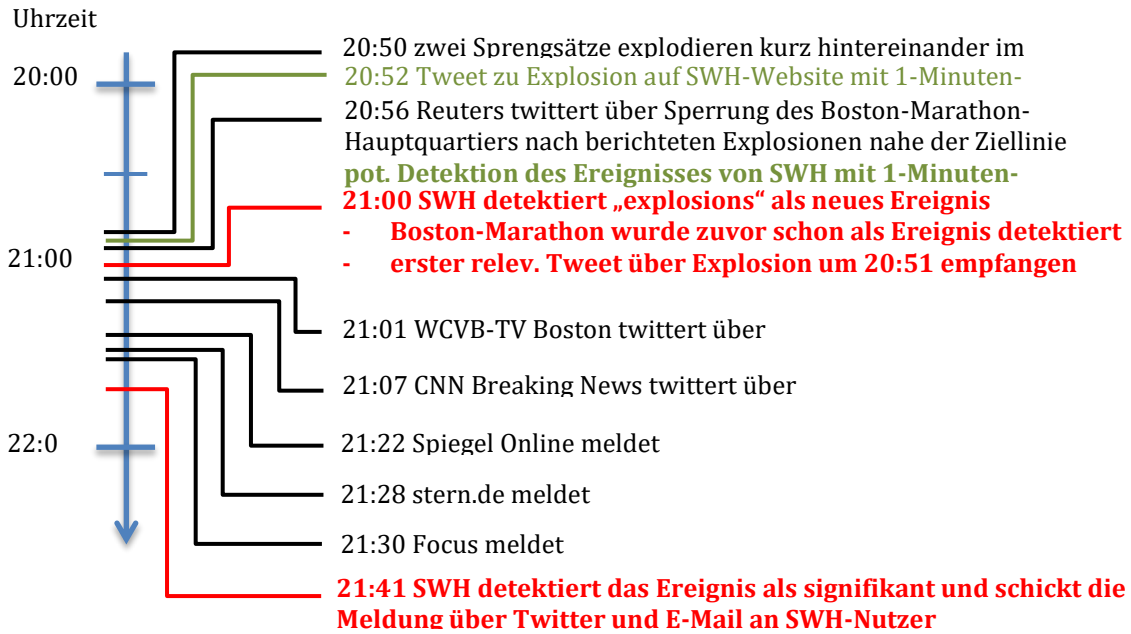

Abbildung 58: Zeitlicher Ablauf des Ereignisses und wer wann was berichtete. Quellenangabe zu den Zeitpunkten: 20:50 [137], 20:56 [138], 21:01 [139], 21:07 [140], 21:22 [137], 21:28 [141], 21:30 [142]

Der Vergleich mit den etablierten Medien, zeigt hier, dass die Vorbedingungen für die SWH-Analyse sehr ungünstig waren. Zum einem war ein großes Presseaufgebot schon vor Ort des Geschehens als die Explosionen geschahen und zum anderen wirkte sich das 10-minütige Analyseraster der SWH-Software als extrem ungünstig aus. Gerade als die letzte Analyse um 20:50 Uhr abgeschlossen war, erreichte der erste Tweet über eine Explosion um 20:51 Uhr das SWH-System. Durch das Analyseraster von 10 Minuten begann die nächste Analyse erst wieder um 21:00 Uhr. Dennoch konnte, trotz den ungünstigen Vorbedingungen, die SWH-Analyse auch hier zeitnah Ergebnisse liefern. Die SWH-Analyse konnte das Ereignis 6 Minuten früher detektieren als CNN und detektierte das Ereignis als ein signifikantes Ereignis als bei den hiesigen Nachrichtenseiten gerade erst die ersten Meldungen eintrafen.
Da sich hier das 10-minütige Analyseraster als sehr ungünstig erwiesen hat, wurde im Nachhinein dieser Zeitbereich noch einmal analysiert. Diesmal fand eine Analyse jede Minute statt. In Abbildung 58 sind an den grünen Einträgen dazu die Ergebnisse abzulesen. Der erste Tweet zur Explosion erreichte das System um 20:51 Uhr. Schon in der Analyse um 20:52 Uhr fand sich dieser Tweet unter den Beispiel-Tweets zu diesem Ereignis auf der SWH-Website. Jedoch war zu diesem Zeitpunkt der Term „explosions“ noch nicht als signifikant erhöht detektiert worden. Dies war erst bei der Analyse um 20:56 Uhr der Fall. Die Abbildung 59 zeigt hier die gekürzte Ansicht dieser Analyse. Zu erkennen ist dort das der Term „explosions“ als signifikant erhöhter Term registriert wurde. Die Tweets, welche ausgewählt wurden, um das Ereignis näher zu beschreiben und den Term „explosions“ enthielten, wurden farblich markiert. Somit konnte zeitlich mit der Reuters Twitter-Meldung gleichgezogen werden. Das Ereignis konnte mit dem 1-Minuten-Raster bereits nach 6 Minuten detektiert werden.
Ein Jahr später, zum Jahrestag des Anschlages, wurde das Gedenken an den Anschlag selbst, als ein Ereignis von der SWH-Software detektiert. Das ist in Abbildung 60 deutlich zu sehen. Auch der stattfindende Boston-Marathon wurde ebenfalls wieder als Ereignis erkannt.

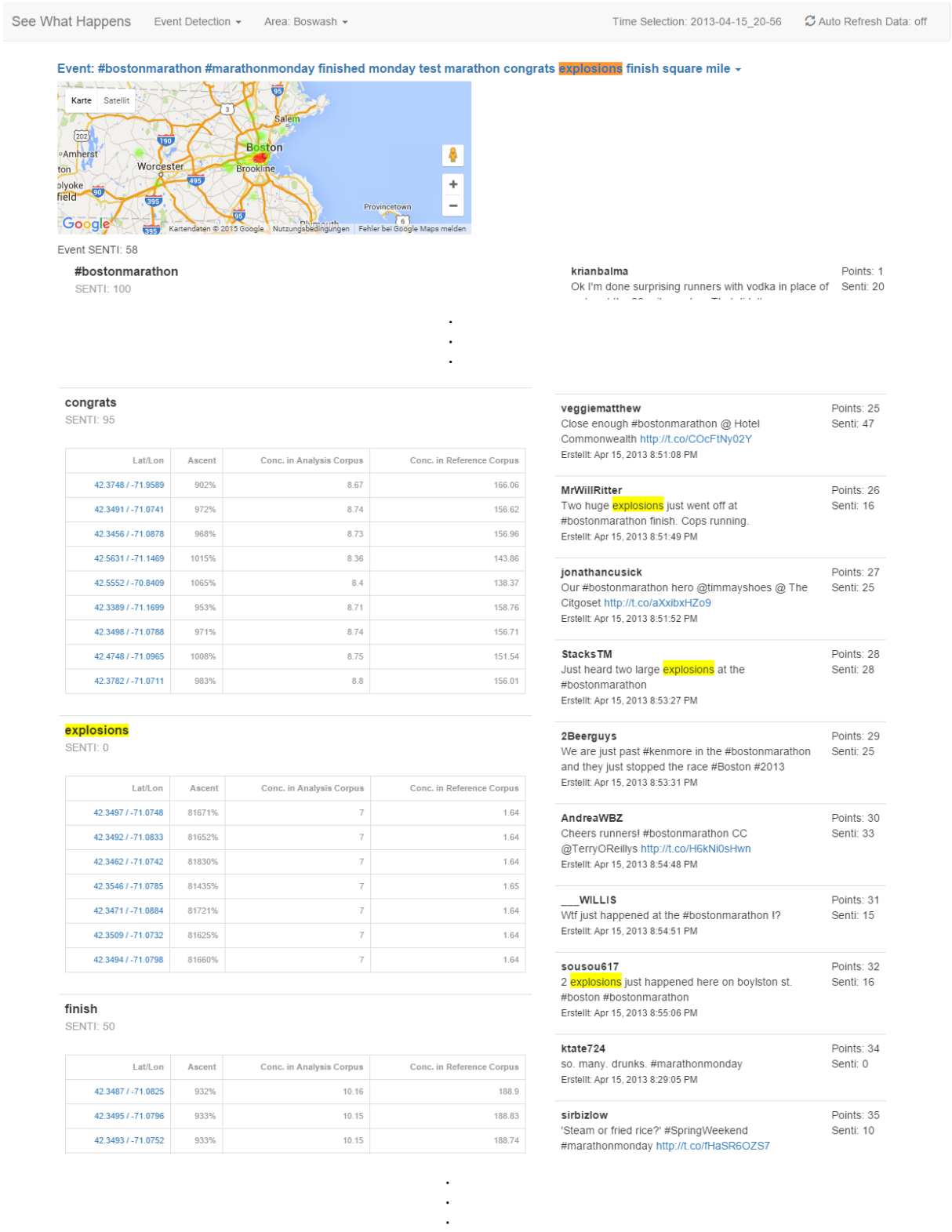

Abbildung 59: Ausschnitt der Boston-Marathon-Analyse vom 15.04.2013 um 20:56 Uhr

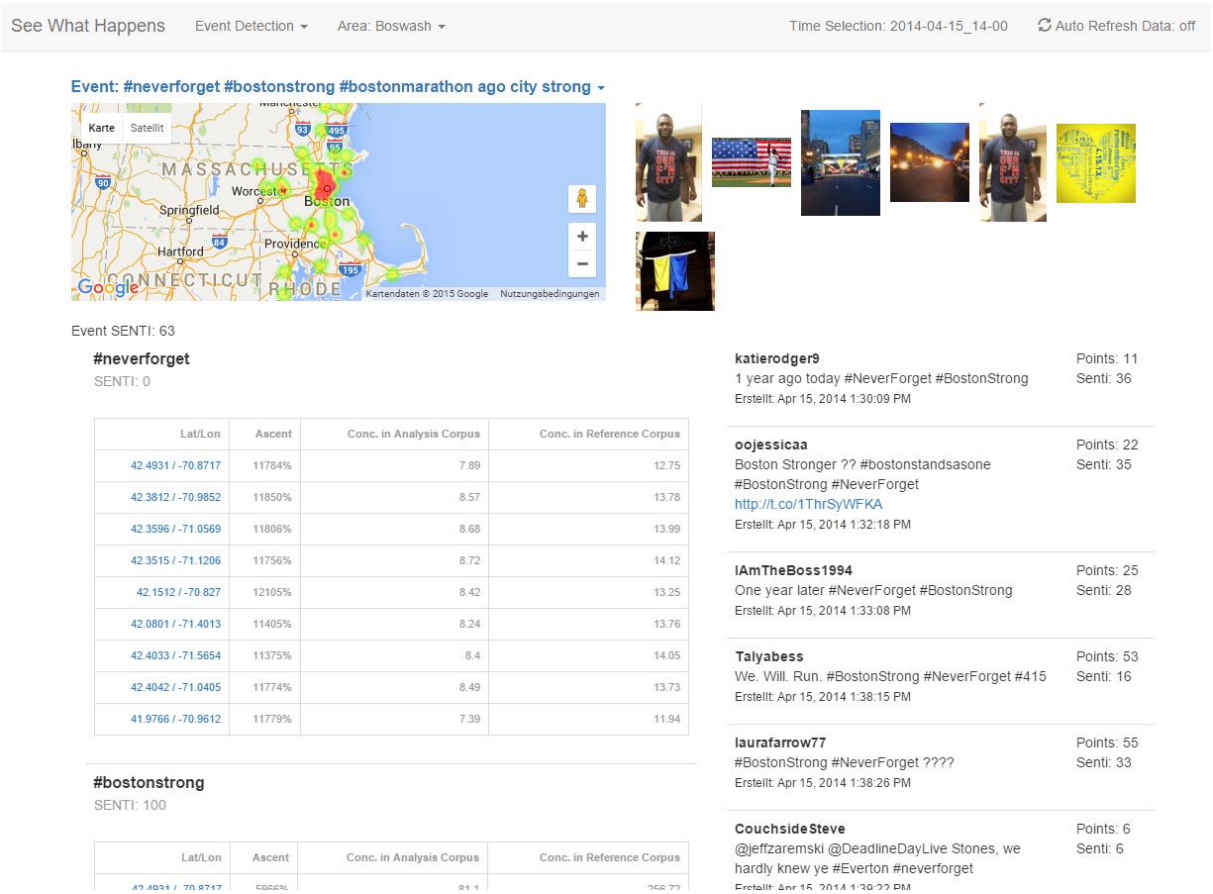

Abbildung 60: Gedenken an den Boston-Marathon Anschlag nach einem Jahr

5.1.3.3 Erdbeben in Los Angeles

In Los Angeles gab es am 17.03.2014 ein leichtes Erdbeben der Stärke von 4,4 auf der Richterskala [143].
Folgender zeitlicher Ablauf wurde recherchiert (Abbildung 61):

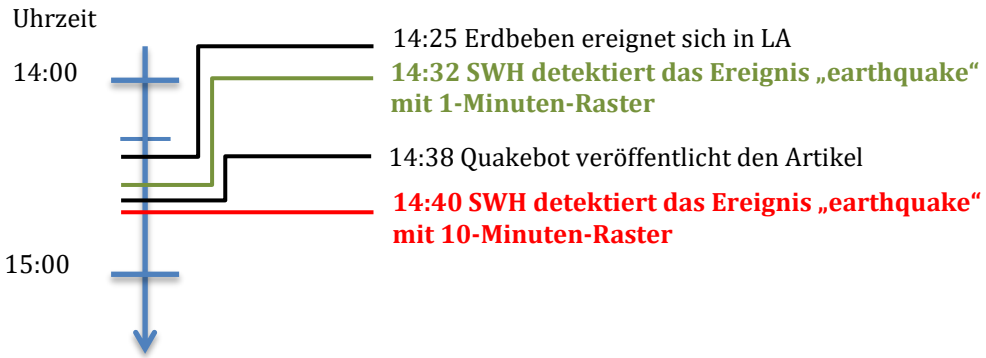

Abbildung 61: Zeitlicher Ablauf des Erdbebens und wer wann was erkennt

Obwohl das Ereignis außerhalb der detektierten Regionen lag, konnte es dennoch aus dem Tweet-Strom detektiert werden. Der Grund war z.B. der, dass dieses Ereignis z.T. live im Fernsehen übertragen wurde [143]. Somit wurde über dieses Ereignis sofort im ganzen Land berichtet. Die Detektion soll hier nicht mit etablierten Medien verglichen werden, sondern mit einem System, welches nach dem Erdbeben für Schlagzeilen gesorgt hat, nämlich Quakebot. Ein Journalist der Los Angeles Times hatte sich für den Fall eines Erdbebens ein Programm geschrieben, welches die Meldungen des U.S.

Geological Survey entgegen nimmt und diese Daten in einen vorgeschriebenen Artikel einfügt. Der somit generierte Artikel muss nur noch von dem Autor freigeschaltet werden. Somit konnte die Los Angeles Times als erstes über das Erdbeben berichten [144].
Quakebot konnte hier 2 Minuten vor der SWH-Software, welche im 10-Minuten-Raster lief, einen Artikel veröffentlichen. Dieser Vorsprung ist in diesem Fall sehr gering für ein System, das genau auf solch ein Ereignis implementiert wurde. SWH konnte somit auch hier, obwohl das Ereignis außerhalb der detektierten Region lag, in Vergleich mit Quakebot, ein sehr gutes Ergebnis abliefern.
Bei einer Neuberechnung der Ereignisdetektion mit einem 1-Minuten-Raster konnte das Ereignis dagegen 8 Minuten früher detektiert werden. Somit lag die Detektion 6 Minuten vor Quakebot. Der Term wurde in der Region Boswash als signifikant erhöht detektiert und wurde zu einem anderem Ereignis mit dazugerechnet welches den St. Patrick's Day und den gerade stattgefundenen Schneefall beschreibt. Dieses Ergebnis ist in Abbildung 62 zu sehen.

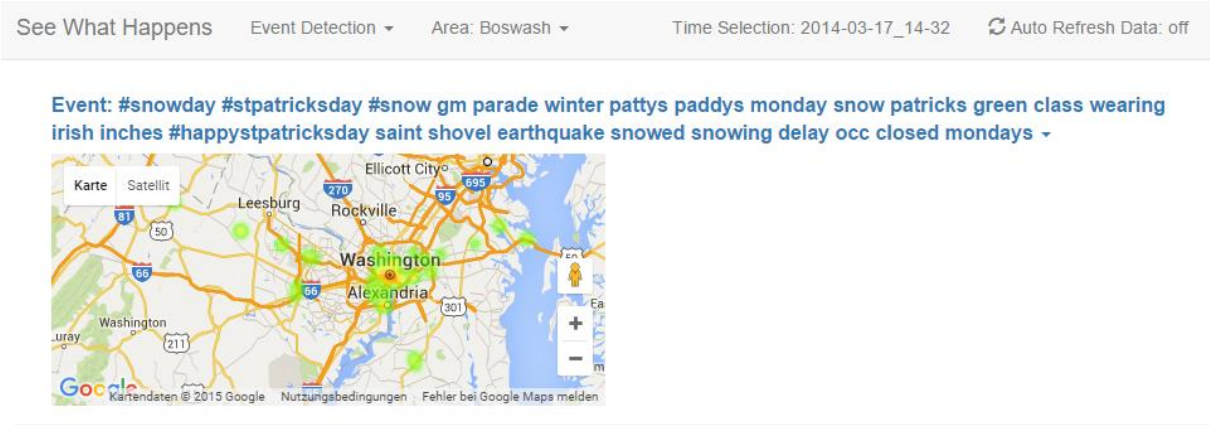

Abbildung 62: Analyseergebnis der Ereignisdetektion in der Region Boswash vom 17.03.2014 14:32 Uhr

An den zuvor ausgewählten großen Ereignissen konnte beispielhaft aufgezeigt werden, wie die SWH-Software große Ereignisse detektiert und wie die Geschwindigkeit im Vergleich mit den etablierten Medien dazu einzuordnen ist. Es konnte gezeigt werden, dass die SWH-Software zeitnah die Ereignisse detektieren konnte z.T. sogar etwas früher als es die Nachrichtenagenturen es über Twitter selbst veröffentlicht haben. Auch in, für die SWH-Software, ungünstigen Situationen, wie z.B. dem Boston-Marathon Anschlag mit einem großem Presseaufgebot vor Ort und dem für die SWH-Analyse ungünstigen Zeitpunkten, konnte das Ereignis dennoch in einem sehr frühem Stadium detektiert werden. Dies war schon möglich mit einem implementierten 10-Minuten-Analyseraster, welches die Software momentan nutzt. Mit einem deutlich engeren Analyseraster von einer Minute, konnte gezeigt werden, dass sich hier die Zeitpunkte der Detektion, bei zwei beispielhaften Ereignissen, noch einmal verbessern konnten. Man konnte in diesem Fall mit den etablierten Medien, auch bei einem großen Presseaufgebot vor Ort, wie im Falle des Boston-Marathons, mithalten. Auch im Vergleich mit speziell implementierter Software wie Quakebot konnte man sogar schneller reagieren. Neben der eigentlichen Detektion des Ereignisses bzw. des Terms liegen zu diesem Zeitpunkt, durch die SWH-Analyse, bereits Beispiel-Tweets vor, die das Ereignis näher beschreiben oder schon entsprechendes Bildmaterial, welches das Ereignis zeigt (z.B. Abbildung 54). Ein Ziel ist es zukünftig komplett ohne einem Analyseraster auszukommen, um die Ereignisse so früh wie möglich detektieren zu können. Dazu ist es notwendig, dass die Differenzanalyse kontinuierlich durchgeführt

wird. Jeder neue eintreffende Tweet triggert dann die Neuberechnung der Ereigniserkennung an und kann dann potentiell die Ergebnismenge beeinflussen. Das momentane 10 Minuten Raster ist eine willkürliche Festlegung. Die Idee dazu war die, dass der Zeitabstand so groß sein sollte das die vorangegangene Analyse auf jeden Fall beendet ist. In den ersten Versionen der Software dauerte die Berechnung der Ereignisse noch einige Minuten. Somit wurde als „runder" Wert ein 10 Minuten Raster gewählt um auf aktuelle Ereignisse noch schnell reagieren zu können.

5.1.4 Eingestellte Parameter

Die Ergebnisse der gezeigten Analysen sind natürlich abhängig von den eingestellten Parametern. In Kapitel 3.3 wurden die Parameter bzw. die Stellschrauben der Algorithmen genau erklärt und wie sie die Algorithmen beeinflussen. So bestimmt man mit dem Effektradius z.B. wie viele Tweets in die Berechnung mit einfließen, indem man über den Effektradius den Wirkbereich jedes einzelnen Tweets verändert. Weitere Parameter sind z.B. das maximale Alter des Referenzkorpus, was angibt nach welcher Zeit der Korpus aus den aktuellen Daten neu erstellt werden muss.

	Deutschland	**Boswash**	**New York**	**San Francisco**
max. Alter des Referenzkorpus	24h	24h	24h	24h
max. Alter des Analysekorpus	10min	10min	10min	10min
zeitl. Breite des Referenzkorpus	15 Tage[1]	4 Tage[2]	12 Tage[3]	30 Tage[4]
zeitl. Breite des Analysekorpus	1h[5]	30min[6]	1h[7]	1h[8]
Effektradius	100km	140km	4km	20km
Nutzerblockzeit nach Tweet	30min	30min	30min	30min
min. Termfrequenz	3	6	6	6
Termanstiegsschwelle	550%	550%	550%	550%
signifikantes Ereignisschwelle	3	3	3	3

Tabelle 12: SWH-Parameter für die Ereignisdetektionsanalyse in den jeweiligen Regionen

In Tabelle 12 sind diese Parameter (Stand Januar 2016) für die einzelnen untersuchten Regionen aufgelistet werden, um die Unterschiede der Einstellungen der Prototypen in den einzelnen Regionen besser zu verstehen. Die meisten Einstellungen sind in den verschiedenen Regionen gleich. Z.B. wird in allen Regionen der Referenzkorpus alle 24h neu gebaut und es startet alle 10 Minuten ein neuer Analyseschritt, der zuerst einen neuen Analysekorpus für die jeweilige Region bildet. Bei der Breite der Korpora unterscheiten sich die Werte aber z.T. drastisch. So ist z.B. die zeitliche Breite des Referenzkorpus in Deutschland mit 15 Tagen relativ groß im Gegensatz zu den Referenzkorpora in den USA. Dies liegt daran, dass die Tweet-Konzentration in Deutschland viel geringer ist. Die zeitliche Breite des Analysekorpus ist i.d.R. immer 1h,

[1] ca. 1,3 Mio Tweets im Referenzkorpus
[2] ca. 1,5 Mio. Tweets im Referenzkorpus
[3] ca. 1,6 Mio. Tweets im Referenzkorpus
[4] ca. 1,6 Mio. Tweets im Referenzkorpus
[5] ca. 3 600 Tweets im Analysekorpus
[6] ca. 7 800 Tweets im Analysekorpus
[7] ca. 5 500 Tweets im Analysekorpus
[8] ca. 2 200 Tweets im Analysekorpus

außer in der Boswash-Region, da dort die Tweet-Konzentration so hoch ist, dass man hier schon in 30 Minuten genügend Tweets für eine Analyse hat. Zur Bestimmung der zeitlichen Breite muss man im Hinterkopf behalten, dass für die Terme, die in den Tweets enthalten sind, die Termkonzentration für den jeweiligen Ort errechnet werden müssen. D.h. es sollte für jeden Ort der Term in diesem Zeitraum häufiger aufgetreten sein als die Mindesttermhäufigkeit Schwelle (siehe Kapitel 3.3.4.3) vorgibt. Je häufiger der Term an einen Ort vorkommt umso genauer lässt sich die Termkonzentration für diesen Ort bestimmen. Jedoch erhöht sich mit zunehmender zeitlicher Breite des Analysekorpus die Trägheit des Systems. Mit zunehmender Breite des Analysekorpus dauert es umso länger bis ein stattfindendes Ereignis und die einhergehende Termkonzentrationänderung, sich im Analysekorpus niederschlägt hat.
Die Termanstiegsschwelle ist bei allen Regionen ebenfalls gleich eingestellt. Das ist die Schwelle, ab der der hochgerechnete Termkonzentration als ungewöhnlich hoch detektiert wird und auf der Website angezeigt wird. Ein Term muss im Analysekorpus hochgerechnet mehr als 5,5 mal häufiger auftreten als im Referenzkorpus, damit er von der SWH-Software gemeldet wird. Diese Schwelle kann man anpassen, um die Empfindlichkeit der Analyse einzustellen. Senkt man die Schwelle ab, so bekommt man mehr Ergebnisse. Doch besteht natürlich hier die Gefahr, dass die Ergebnisse kein Ereignis repräsentieren, sondern dass es sich um natürliche Termhäufigkeitsschwankung handelt. Setzt man dagegen die Schwelle höher, so mindert man zwar die Gefahr, solche zufälligen Schwankungen zu detektieren, aber man verschluckt kleinere Ereignisse bzw. detektiert die Ereignisse erst mit einer Verspätung. Möchte man also sehr frühzeitig Ereignisse detektieren, so muss man die Schwelle absenken, um schon sich anbahnende Termhäufigkeitsänderungen frühzeitig zu erkennen.
Die signifikante Ereignisschwelle ist dagegen die Schwelle ab der ein Ereignis als ein signifikantes Ereignis bzw. Großereignis detektiert wird. Dass bestimmte Terme als signifikant erhöht detektiert werden, ist i.d.R. normal. Gibt es davon aber ungewöhnlich viele Ergebnisse so liegt wohl ein großes Ereignis vor. Rechnet man den Vierwochendurschnitt aus und vergleicht ihn mit der aktuellen Anzahl von Termen, die man detektiert hat, so sieht man, ob die Ergebnismenge erhöht ist. Gibt es dreimal mehr Ergebnisse als sonst (signifikante Ereignisschwelle), so wird von einem großen Ereignis ausgegangen und die Nutzer ggf. gesondert benachrichtigt (vgl. Kapitel 4.4.3).

5.2 Stimmungsdetektion

Um zu zeigen, dass die entworfenen Algorithmen auch für andere Aufgabenstellungen genutzt werden können, wurde zusätzlich eine Stimmungsdetektion realisiert, die Teile der Algorithmen der Ereignisdetektion nutzt. Die Algorithmen zur Berechnung der Stimmung eines Tweets oder eines Terms wurde in Kapitel 3.4 detailliert erläutert.
Die Stimmungsdetektion findet in der SWH-Software parallel zur Ereignisdetektion statt und läuft im gleichen Analyseraster wie diese, also alle 10 Minuten. Das Ziel der Stimmungsdetektion soll es sein, Stimmungsschwankungen zu detektieren und, wie bei der Ereignisdetektion, den Ort der Schwankung zu benennen.

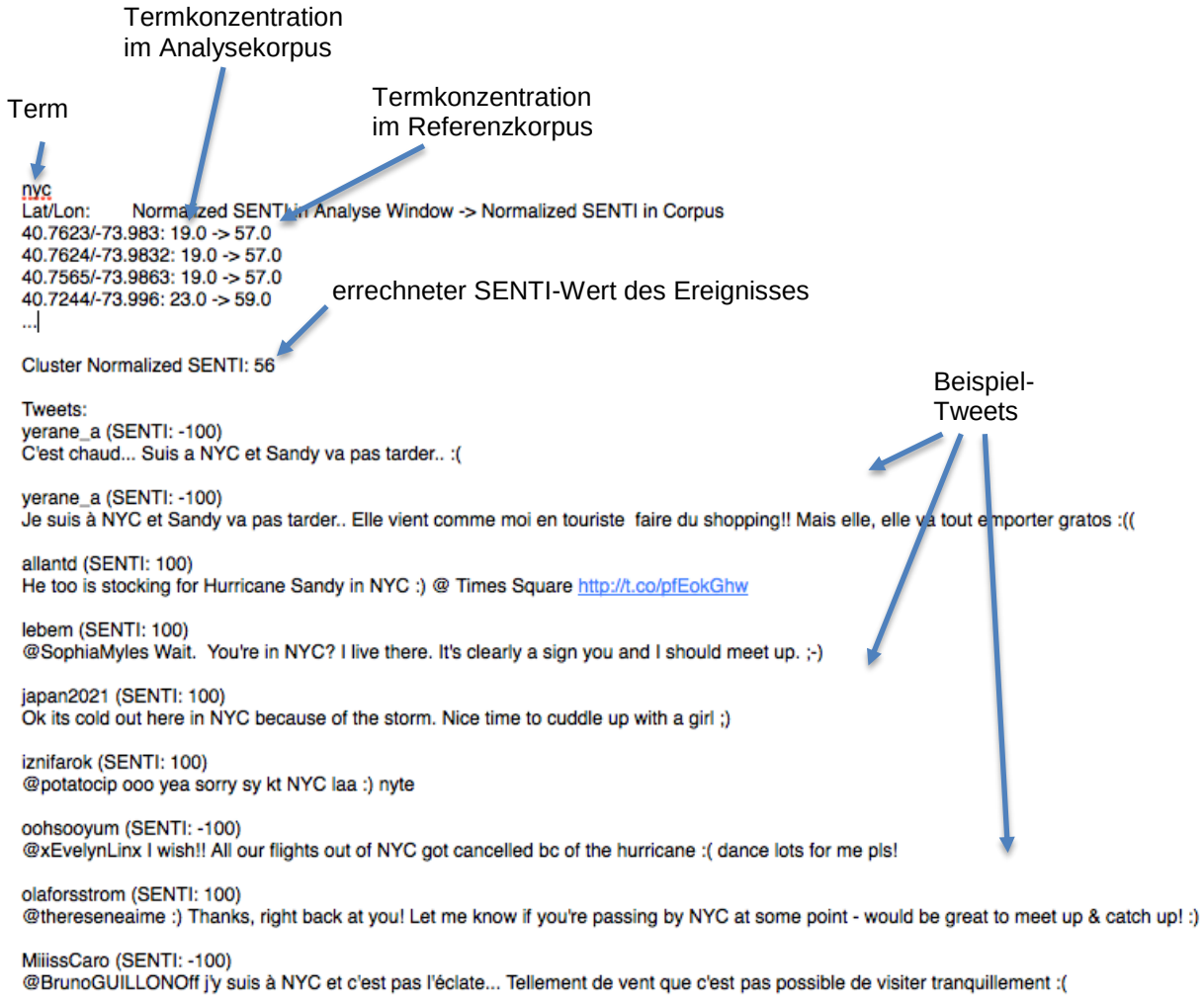

Abbildung 63: Stimmungsänderung nach dem Hurrikan Sandy [145] in New York, Auszug aus der Warnbenachrichtigung des SWH-Systems

Wie bei der Ereigniserkennung konnten auch hier Schwankungen des errechneten SENTI-Wertes beobachtet werden. Im Gegensatz zur Ereigniserkennung können diese Schwankungen meist nicht einer gemeinsamen Ursache bzw. eines bestimmten Ereignisses zugeordnet werden. Hier ist zu beobachten, dass der SENTI-Wert zwar schwankt aber die extrahierten Tweets dazu kein gemeinsames Ereignis zeigen und es somit eine natürliche Schwankung darstellt, so wie z.B. die Termhäufigkeit auch schwankt. Doch können bestimmte Schwankungen sehr wohl einem Ereignis bzw. einer Ursache zugeordnet werden. Im Gegensatz zur Ereignisdetektion, bei der die Ereignisse in gewisser Weise mit Nachrichten auf einer News-Seite verglichen werden können,

handelt es sich bei den detektierten Ereignissen der Stimmungsanalyse eher um emotionale Ereignisse. Dies soll an ein paar Beispielen verdeutlicht werden.
Bei dem ersten Beispiel, in Abbildung 63, geht es um einen Abfall des ermittelten Stimmungswertes (SENTI-Werte) für den Term „nyc“ (New York City) in der Stimmungsanalyse für den Bereich New York am 01.11.2012. Dort fiel der SENTI-Wert von ca. 57 auf ca. 19 ab, wurde also negativer aufgefasst. Blickt man in die Beispiel-Tweets dann sieht man was geschehen ist. Der Hurrikan Sandy [145], ein tropischer Wirbelsturm der erhebliche Schäden anrichtete und in der Nähe von New York am 29.10.2012 auf Land traf und in der Stadt für Stromausfälle und Überschwemmungen verursachte, war gerade abgezogen und die Schäden wurden sichtbar. Die Nutzer twitterten daher von Flugausfällen und anderen Unannehmlichkeiten die der Sturm verursachte.
Dies führte dazu, dass die Stimmung in Zusammenhang mit New York und somit zu dem Term nyc negativer wurde.

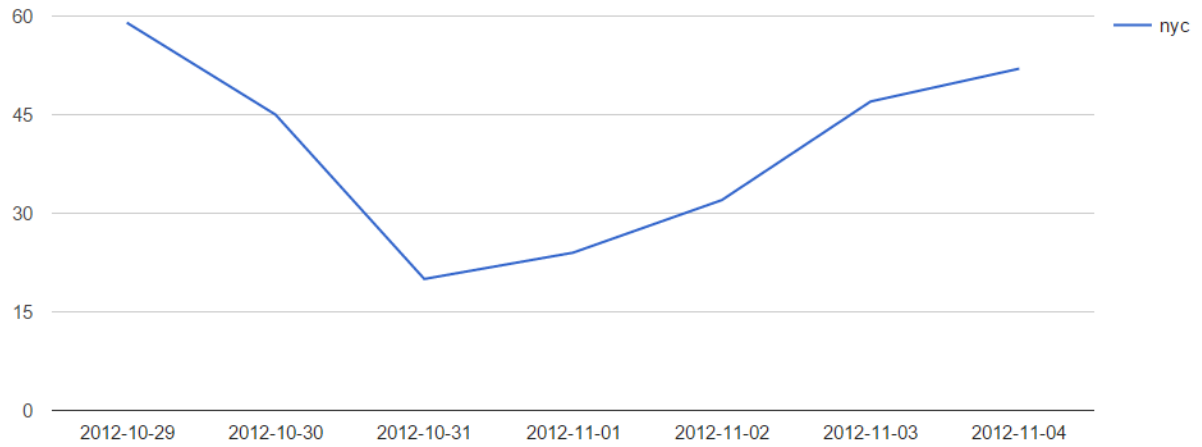

Abbildung 64: Stimmungskurve des Terms "nyc" für New York in der Woche des Hurrikan Sandy

In Abbildung 64 sieht man dazu den SENTI Verlauf dieser Tage, des Terms „nyc“, und den Abfall des Wertes um den 31.10.2012-01.11.2012.
In dem nächsten Beispiel in Abbildung 65 wird deutlich, dass diese detektierten emotionalen Ereignisse sich stark von den detektierten Termen der Ereignisdetektion unterscheiden. In einem Beispiel sank die Stimmung in der Bay Area für die Terme „San Francisco“ und „back“. Detektiert wurde dies am 24.04.2014, also in der Woche nach Ostern. In den Beispiel-Tweets sieht man, dass die Nutzer darüber schreiben dass ihr Urlaub vorbei ist und man wieder zurück (back) muss. Man ist also traurig, dass die Zeit vorbei ist, was dazu führt, dass die Stimmung gedrückter bzw. negativer wird. Man sieht in diesem Beispiel, wie unterschiedlich die detektierten Ereignistypen sind.

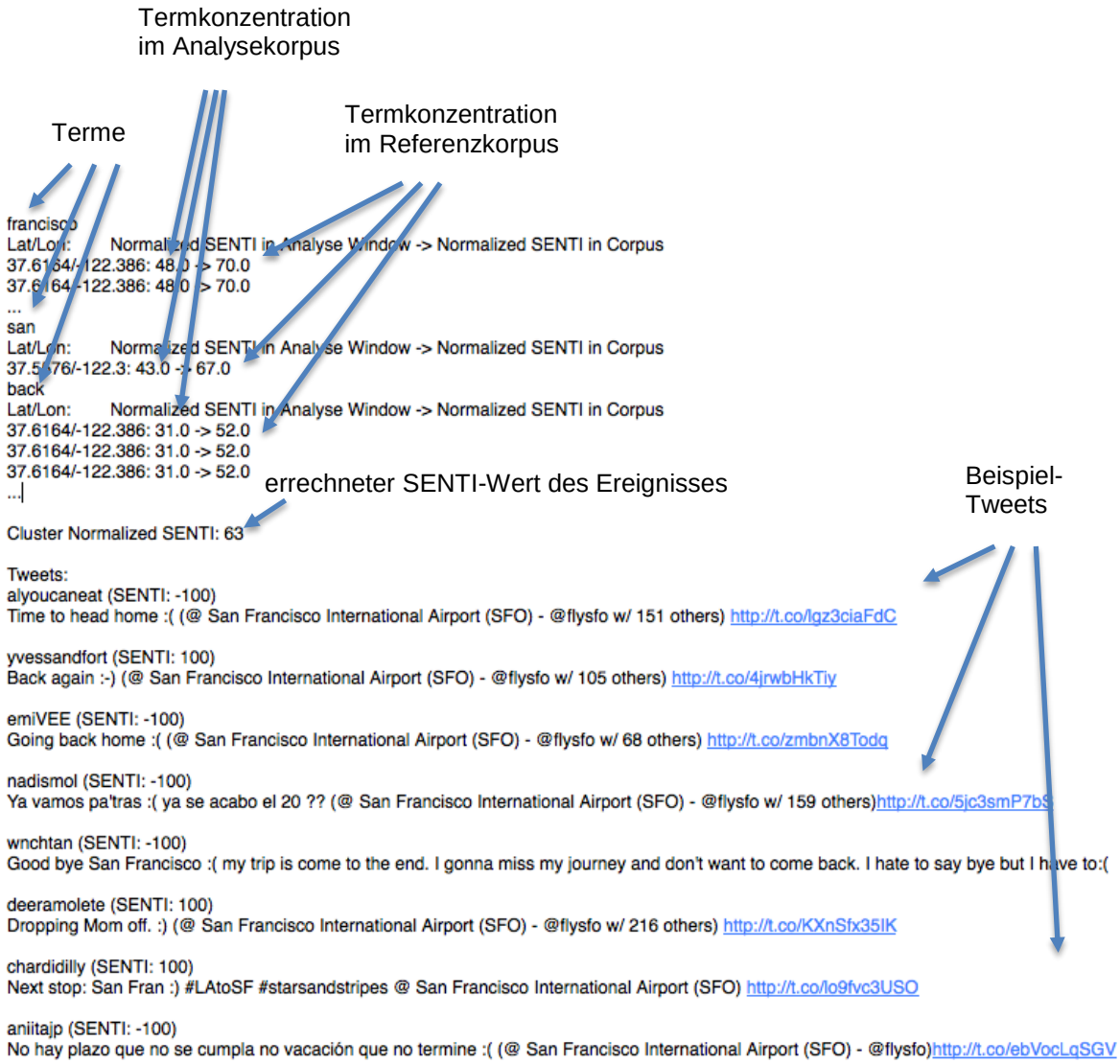

Abbildung 65: Stimmungsänderung in der Bay Area nach Ostern 2014, Auszug aus der Warnbenachrichtigung des SWH-Systems

Im nächstem Beispiel in Abbildung 66 geht es um ein ähnliches emotionales Ereignis. Hier sinkt der SENTI-Wert für den Term „school“. Auch hier zeigen wieder die extrahierten Beispiel-Tweets was eigentlich los ist. Der erste Schultag nach den Ferien hat begonnen und die Nutzer, hier die twitternden Schüler, sind in ihren Tweets davon nicht begeistert und somit sinkt der schon zuvor niedrige SENTI-Wert für den Term „school“ noch weiter ab.

An den ausgewählten Beispielen konnte man sehen, dass die Detektion von Stimmungsschwankungen ebenfalls funktioniert. Zu beachten ist, dass die Detektion der Stimmungsschwankungen hier sprachunabhängig funktioniert. Durch diese Analyse detektiert man komplett eine andere Klasse von Ereignissen, die man wohl eher als emotionale Ereignisse beschreiben könnte. Somit ist es auch möglich solche emotionalen Ereignisse zu detektieren, obwohl eine schlechte Stimmung am ersten Schultag unter den Schülern vielleicht nicht ganz überraschend ist.

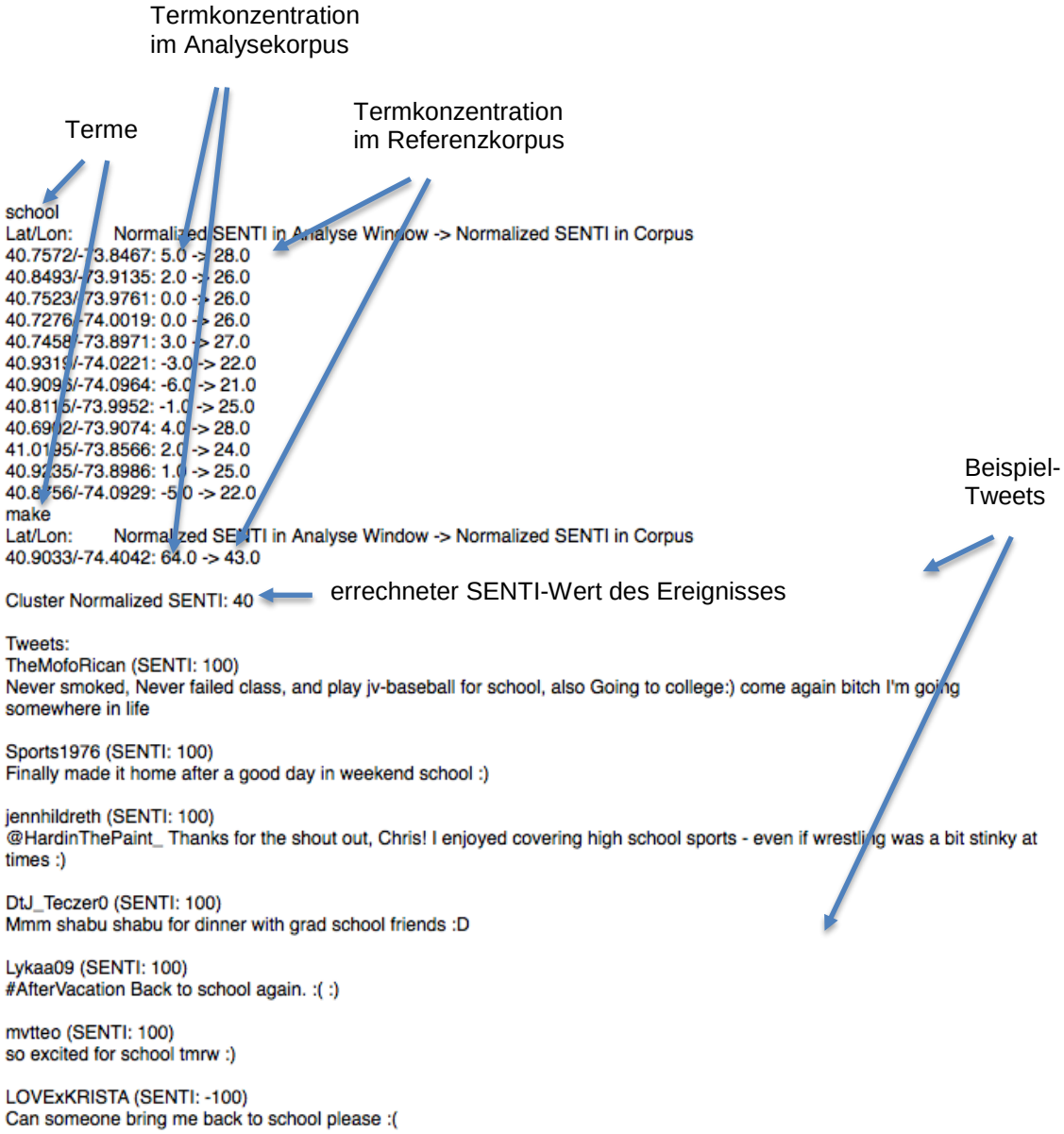

Abbildung 66: Stimmungseinbruch für den Term "school" nach den Ferien an der Ostküste der USA

5.2.1 Eingestellte Parameter

Aus Performance Gründen nutzt die Stimmungsanalyse dieselben Referenzkorpora wie die Ereignisdetektion. Die Analysekorpora werden dagegen neu berechnet, da diese zeitlich breiter sind, denn es sind nun nur Tweets interessant, die ein Emoticon beinhalten, welches zur Bewertung des Tweets hinzugezogen wird. Da dies nur in einer Teilmenge der Tweets der Fall ist, müssen Tweets aus einem größeren Zeitraum für die Analyse genutzt werden.

Die folgende Tabelle zeigt die momentan genutzten Parameter (Stand Januar 2016).

	Deutschland	**Boswash**	**New York**	**San Francisco**
max. Alter des Referenzkorpus	24h	24h	24h	24h
max. Alter des Analysekorpus	10min	10min	10min	10min
zeitl. Breite des Referenzkorpus	30 Tage[9]	4 Tage[10]	12 Tage[11]	30 Tage[12]
zeitl. Breite des Analysekorpus	5 Tage	3 Tage	3 Tage	5 Tage
Effektradius	200km	140km	10km	50km
Nutzerblockzeit nach Tweet	30min	30min	30min	30min
min. Tweets mit Emoticon	20	30	30	30
SENTI-Änderungsschwelle	20	20	20	20

Tabelle 13: SWH-Parameter für die Stimmungsdetektionsanalyse in den jeweiligen Regionen

Im Vergleich zu den SWH-Parametern der Ereignisdetektion sieht man, dass das maximale Alter der Korpora dasselbe ist, wie in der Ereignisdetektion, da die Neuberechnung der Korpora in demselben Zyklus stattfindet, wie bei der Ereignisdetektion. Die zeitliche Breite der Analysekorpora wurde breiter gewählt und liegt nun im Zeitraum von Tagen statt Stunden. Für die Analyse gibt es zwei Schwellen, die genutzt werden. Einmal die minimalen Tweets mit Emoticon Schwelle, die angibt, in wie vielen Tweets der Term mindestens an dem jeweiligen Ort aufgetreten sein muss, damit man diesen Wert nehmen kann. Die andere Schwelle ist die SENTI-Änderungsschwelle. Die SENTI-Änderungsschwelle gibt an, wie groß die Änderung des SENTI-Wertes zwischen Analysekorpus und Referenzkorpus sein muss, damit der Term und der dazugehörige Wert als Stimmungsänderung angezeigt wird (siehe Kapitel 3.4.2 für die detaillierte Erläuterung der Schwellen und die Funktionsweise der Stimmungsdetektionsalgorithmen). D.h. hat ein Term im Referenzkorpus einen errechneten SENTI-Wert von 90, also sehr positiv, und beträgt der SENTI-Wert im Analysekorpus nur noch 60, so besteht eine Differenz von >20 SENTI-Einheiten und folglich sollte diese Änderung mitgeteilt werden.
Mit Hilfe dieser gezeigten Parameter lässt sich die Detektion von Stimmungsänderungen den eigenen Wünschen anpassen. Je nach Menge der Tweets mit Emoticon in der zu analysierenden Region, können die zeitlichen Breiten der Korpora angepasst werden. Möchte man die Ergebnismenge verändern, z.B. wenn man noch kleinere Schwankungen detektieren möchte, so können die Schwellen weiter abgesenkt werden oder erhöht werden, um nur noch große Schwankungen zu detektieren.

5.2.2 Weitere Analysen zur Stimmungsdetektion

Bei der Beobachtung der SENTI-Werte über einen längeren Zeitraum lassen sich weitere Erkenntnisse gewinnen. So sieht man, dass die SENTI-Werte bestimmter Terme im Laufe der Zeit schwanken [12].

[9] ca. 420 000 Tweets im Referenzkorpus
[10] ca. 2 Mio. Tweets im Referenzkorpus
[11] ca. 2 Mio. Tweets im Referenzkorpus
[12] ca. 1,47 Mio. Tweets im Referenzkorpus

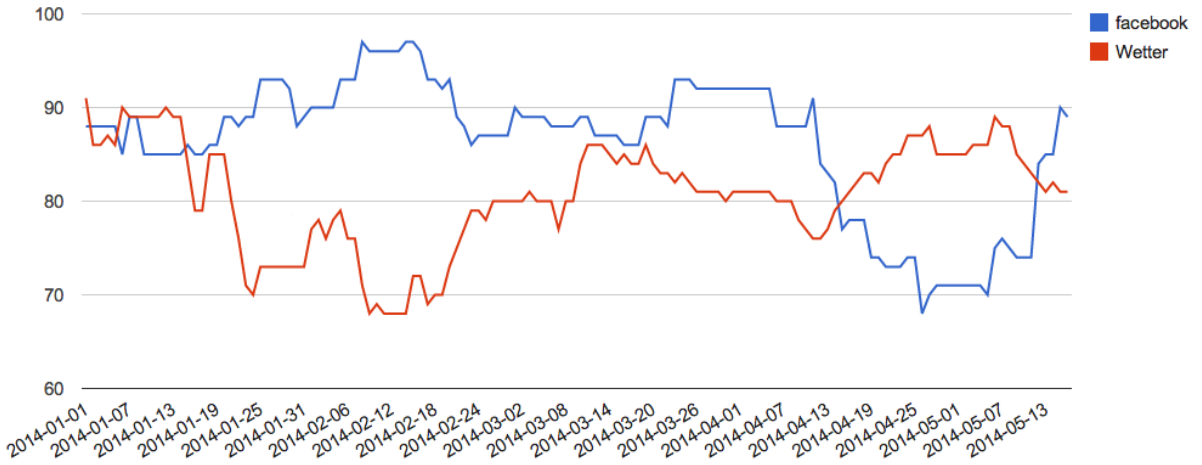

Abbildung 67: Schwankende SENTI-Werte bei den Termen "facebook" und "Wetter" in Deutschland im Zeitraum von 01.01.2014 - 16.05.2014

In Abbildung 67 sieht man dies an den Beispiel-Termen „Facebook“ und „Wetter“ in Deutschland im Zeitraum vom 01.01.2014 bis 16.05.2014. Die SENTI-Schwankungen können auf Ereignisse hindeuten, die mit den Termen im Zusammenhang stehen. Gibt es z.B. Probleme mit einem Dienstleister oder wird eine Aktion von den Nutzern für schlecht gefunden, so wird die Stimmung, also der SENTI-Wert, absinken. Dies ist z.B. zuerkennen an der Facebook Kurve Ende Februar. Ab dem 20.02.2014 ist hier ein absacken der SENTI-Kurve zu erkennen. Die könnte durch den Whatsapp-Kauf [146] von Facebook verursacht sein, der durch Datenschutzbedenken bei vielen Nutzern zuerst eher negativ aufgenommen wurde. Am Verlauf der Wetter-SENTI-Kurve kann man erkennen, dass sie ebenfalls stark schwankt. Herrscht für die jeweilige Zeit „gutes Wetter“, z.B. ist es warm und sonnig, so sind die SENTI-Werte erhöht. Zu sehen z.B. am Jahresanfang als Ende Dezember 2013 sehr sonniges und relativ warmes Wetter herrschte was zu einem hohen SENTI-Wert führte.
Es lassen sich noch weitere zeitliche Betrachtungen des SENTI-Wertes vornehmen, so wurde beobachtet, dass Terme unterschiedlich stark schwanken. Dazu wurden für verschiedene Terme, in Tabelle 14, der SENTI-Durchschnitt und die Standardabweichung im Zeitraum von April 2012 bis Mitte August 2012 aus der Region New York errechnet.
Es ist hier zu erkennen, dass die Standardabweichung bei manchen Termen relativ gering ist, d.h. die SENTI-Werte schwanken kaum, aber es gibt auch Terme mit großen Schwankungen. Damit ließen sich Terme sprachunabhängig in Terme deren Stimmung kaum schwanken und in Terme deren Stimmung stark schwanken klassifizieren. Gerade letztere Gruppe wäre dann eine interessant Gruppe, die man über längere Zeit beobachten könnte. Diese Daten wären z.B. interessant für Meinungsforschungsinstitute und deren Kunden (z.B. Politiker), um die Stimmung der Bevölkerung genau zu analysieren, ganz ohne Befragungen.

Term	Region	Mittlerer SENTI-Wert	Standardabweichung des SENTI-Wertes
love	New York	92,18	0,91
amazing	New York	92,49	2,68
good	New York	88,29	1,33
sad	New York	-46,61	7,69
sick	New York	-28,95	8,79
weather	New York	65,03	10,08
weather	Bay Area	81,10	2,39
weather	Boswash	58,29	5,62
wetter	Germany	76,27	5,45
vacation	New York	67,43	18,41
sun	New York	73,01	12,26
rain	New York	48,07	11,85
tv	New York	71,29	12,41
bahn	Germany	76,37	9,99
mood	New York	72,02	9,44
money	New York	65,92	10,37
apple	New York	75,43	9,02
facebook	New York	75,06	11,70
instagram	New York	77,55	14,69
ipad	New York	71,21	19,84
iphone	New York	55,32	12,11
jfk	New York	54,90	8,28
macdonalds	New York	75,87	12,74

Tabelle 14: Ausgewählte Terme (April 2012 – Mitte August 2012) aus der Region New York mit deren Durchschnitts-SENTI-Wert (s(w)) und der Standardabweichung [12]

Es ist auch zu beobachten, dass die Schwankungen mancher Terme sich nach Region unterscheiden können. Der Term „weather“ aus Tabelle 14 z.B. schwankt sehr unterschiedlich in den Regionen. Bei großen Schwankungen bedeutet dies, dass der Term stark unterschiedlich stimmungsmäßig bewertet wird. Bei dem Term Wetter bzw. hier „weather“ ist zu beobachten, dass man über den Verlauf der Schwankung ziemlich gut das dort vorherrschende Wetter ablesen kann. So könnte man interpretieren, dass das Wetter wohl an der Ostküste der USA stark schwankt (Standardabweichung: 5,62) und nicht so positiv bewertet wird wie an der Westküste der USA wo der SENTI-Wert wesentlich höher ist und auch nur sehr gering schwankt (Standardabweichung: 2,39).
Die Betrachtung des SENTI-Wertes über die Zeit, lässt Raum für weitere Analysen und Forschung. So könnten die Schwankungen der Terme in den Regionen miteinander verglichen werden und näher untersucht werden, welche Ereignisse welche Schwankungen ausgelöst haben.

6 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde gezeigt, dass mit Hilfe der neu entwickelten Algorithmen aus einem Echtzeitdatenstrom von georeferenzierten Kurzmitteilungen, eines sozialen Netzwerkes, neue Ereignisse und deren Ort detektiert werden können. So kann die Frage: „Was geschieht gerade?“ durch die Analyse von Daten der sozialen Netzwerke, mit Hilfe der in dieser Arbeit entwickelten Algorithmen, beantwortet werden.
Diese Algorithmen sind so entworfen, dass sie verschiedene Arten von Ereignissen detektieren können. Dabei reicht die Spanne der erkannten Ereignissen von Sportereignissen, Kongressen, kulturellen Events und signifikanten Wetterereignissen bis hin zu Katastrophen (Naturkatastrophen, Unfällen oder ähnlichen katastrophalen Ereignissen). Eine Spezialisierung der Algorithmen auf spezifische Ereignisse, wie es oft in ähnlichen Systemen zu finden ist, die in Kapitel 2 vorgestellt wurden, ist hier nicht nötig.
Mit der Erweiterung eines Burst-basierten Ereignisdetektionssystems mit Hilfe der Einführung eines adaptiven Referenzkorpus bzw. von ortsabhängigen Korpora konnte die Detektionsleistung verbessert werden. Der Ort der zu untersuchenden Kurzmitteilungen fließt direkt in die Ereignisdetektion ein und verbessert somit die Erkennungsleistung, da ortsabhängige Termkonzentrationen mit berücksichtigt werden. Zu jedem detektierten Ereignis kann zudem später der Ort des Ereignisses präzise benannt werden.
Wie in den vorgestellten Arbeiten in Kapitel 2 zu sehen war, besteht ein großer Teil der Aufgaben bisheriger Ereignisdetektionssysteme darin, die eintreffenden Daten zu filtern (z.B. wird nur eine bestimmte Sprache unterstützt oder es werden bestimmte Terme wie z.B. Emoticons oder Hashtags entfernt) und aufzubereiten. Diese Vorverarbeitung und Aufbereitung der Eingangsdaten ist häufig sehr komplex und ressourcenhungrig und kann die Detektion von Ereignissen stark einschränken, so dass nur noch ein ganz bestimmter Ereignistyp in einer ganz bestimmten Sprache detektiert werden kann.
Diese Vorverarbeitung der Eingangsdaten ist mit der im Rahmen dieser Arbeit realisierten Ereignisdetektion nicht mehr nötig. Es benötigt keine Trennung der Eingangsdaten in „News“ und „Noise“. Ganz im Gegenteil wäre eine solche Filterung sogar kontraproduktiv da sonst die Algorithmen die ortsabhängigen Termkonzentrationen nicht genau ermitteln können wenn Eingangsdaten zuvor schon herausgefiltert werden würden. Auch eine Filterung der Daten nach bestimmten Sprachen ist nicht nötig, denn die Algorithmen funktionieren, wie in Kapitel 3.3 beschrieben, sprachunabhängig.
Um diese Ansätze zu testen und zu evaluieren, wurde ein Ereignisdetektionssystem namens „See What Happens“ (SWH) implementiert, welches Ereignisse ausgewählter Regionen (Deutschland, Boswash (Ostküste der USA) und Bay Area (Umgebung um San Francisco)) aus den Daten des sozialen Netzwerkes Twitter detektiert. Die Architektur dieses Systems wurde so entworfen, dass unabhängige Analysen parallel ausgeführt werden können, um verschiedene Algorithmen gleichzeitig zu testen bzw. unterschiedliche Analysen auf den Daten durchzuführen. Die Analysesoftware konnte bereits in den letzten Jahren demonstrieren, dass dieses Ziel erreicht werden konnte und die Ereignisdetektion in den unterschiedlichen Regionen und Sprachen funktioniert. Dies wurde in den Kapiteln 5.1.2 und 5.1.3 beschrieben.
SWH detektiert die Ereignisse, benennt deren Ort, erzeugt Heatmaps um den Ort des Ereignisses zu visualisieren, extrahiert Beispiel-Tweets, die das Ereignis näher beschreiben, extrahiert ggf. vorhandenes Bildmaterial und präsentiert es dem Nutzer

auf einer Website. So steht bereits schon zu einem sehr frühen Zeitpunkt (z.T. nach ca. 10 Minuten) eine große Auswahl an Informationen zur Verfügung, die das detektierte Ereignis näher beschreiben.
Die aufgestellten Hypothesen, dass mit einem dynamischen Referenzkorpus eine bessere Differenzanalyse möglich ist sowie, dass mit ortsabhängigen Korpora sich die Ergebnisse noch einmal verbessern lassen, konnten theoretisch sowie praktisch in Kapitel 5.1 mit realen Daten bestätigt werden.
Weiterhin konnte, bei ausgewählten großen Ereignissen, gezeigt werden, dass die Geschwindigkeit der Ereignisdetektion mit der von Nachrichtenagenturen wie z.B. Reuters (Bewertung anhand den veröffentlichten Reuter-Tweets) mithalten kann. Selbst bei einem Vergleich mit einer speziell entwickelten Software, die Nachrichtenartikel selbstständig schreibt, nach einem aufgetretenen Erdbeben im Raum Los Angeles (vgl. Kapitel 5.1.3.3), konnten die eigenen Algorithmen zeitlich mithalten und erkannten das Ereignis gerade einmal 2 Minuten später gegenüber dieser spezialisierten Software.
Dabei ist zu beachten, dass der realisierte SWH-Prototyp noch nicht auf Ablaufgeschwindigkeit optimiert wurde. So führt die SWH-Software die Analysen in den ausgewählten Regionen in einem 10-Minuten-Raster durch und pausiert danach. Dieses Vorgehen wurde deshalb so implementiert um die Ergebnisse der Analysen (Durchläufen mit unterschiedlichen Parametern) direkt miteinander zu vergleichen. Durch die spezifizierten Intervalle konnte genau definiert werden welche Nachrichten für die Analyse ausgewählt wurden und es konnte eine bestimmte Analyse mit einem veränderten Algorithmus bzw. Parametern neu durchgerechnet werden und die Ergebnisse direkt miteinander verglichen werden. Durch Umstellung der Software auf ein 1-Minuten-Raster Analyseintervall konnte gezeigt werden, dass die Detektionszeiten verbessert werden konnten. Durch die Erweiterung des Prototypens hin zu einer kontinuierlichen Analyse, würde sich die Detektionsgeschwindigkeit von Ereignissen noch weiter erhöhen lassen.
Durch Modifikationen der Parameter der SWH-Software lässt sich die Software den Anforderungen anpassen. Auch eine Ad-hoc Benachrichtigung der Nutzer bei signifikanten Ereignissen wurde implementiert und funktioniert.
Es ist geplant die SWH-Software weiter zu entwickeln zu einem Dienst der durch Nutzer im Internet, ähnlich einer Nachrichtenseite, genutzt werden kann. Vergleichbar einer Nachrichtenseite werden die durch Mediendaten angereicherten Ereignisse präsentiert. Im Gegensatz zu den jetzigen Nachrichtenseiten bietet die neue Seite einen Überblick über die gerade passierenden Ereignissen weltweit und es ist immer ersichtlich wo die Ereignisse gerade stattfinden bzw. wo der Hotspot des Ereignisses liegt. Für professionelle Nutzer des Dienstes, wie z.B. Journalisten, wäre auch eine kostenpflichtige Version vorstellbar, die den Nutzern mehr Einstellmöglichkeiten zur Verfügung stellen könnte. So könnten dann die Detektionsschwellen oder andere Parameter durch den Nutzer veränderbar sein, um die Ergebnismenge auf die eigenen Wünsche anzupassen. Durch die größeren zu verarbeiteten Datenmengen und der kontinuierlichen Analyse muss die Software hier natürlich weiter entwickelt werden. Auch die Differenzanalyse selbst könnte erweitert werden. Momentan werden nur die Termkonzentrationen von einzelnen Termen (Uni-Gramme) überprüft. Dies könnte erweitert werden auf die Überprüfung auf N-Gramme. Z.B. würden dann die Terme „San“ und „Francisco“ als eine Einheit betrachtet werden können und nicht getrennt zumal sie ja immer zusammen auftreten wenn die Stadt gemeint ist. Dabei ist zu überprüfen, ob sich dadurch die Ergebnisse noch weiter verbessern lassen.
Mit einer Erweiterung der Ereignisdetektionsalgorithmen ließen sich nicht nur aktuelle Ereignisse detektieren sondern es wäre auch denkbar bevorstehende Ereignisse aus den Daten vorherzusagen. Einige Ereignisse werden z.B. durch vorherige Ereignisse

ausgelöst wie z.B. ein länger andauernder Regen in einem Gebiet zu einem Hochwasser führen kann. Detektiert man solche Vorläuferereignisse in bestimmten Regionen, die schon einmal zu einem bestimmten Nachfolgeereignis in dieser Region geführt haben, so kann man diese Nachfolgeereignisse, also in dem Beispiel das Hochwasser, schon vorher voraussagen (zum jetzigem Zeitpunkt noch nicht veröffentlichte Arbeiten der Universität Leipzig).
Dass der Ansatz der Ereignisdetektionsalgorithmen sich auch auf andere Problemfelder anwenden lässt, wurde mit einer zusätzlichen Stimmungsdetektion gezeigt. Dazu wurde eine sprachunabhängige Stimmungsdetektion konzipiert, die die Bewertung der Terme auf Grundlage von Emoticons vornimmt. Auch hier konnte eine sprachunabhängige Stimmungsdetektion implementiert werden, die ohne aufwendige Vorverarbeitung der Daten auskommt. Mit Hilfe der SWH-Software können nun zusätzlich Terme bewertet werden, ob diese eher positiv oder negativ bewertet werden. Auch Schwankungen von diesem errechneten Stimmungswert (SENTI) können detektiert werden und der Ort der Schwankung genannt werden. So lässt sich eine andere Klasse von Ereignissen detektieren, die als emotionale Ereignisse bezeichnet werden könnte. So sind Schwankungen der Stimmung z.B. nach Naturkatastrophen oder aber auch das Ende von Ferien hiermit erkennbar. Um in Zukunft die Ergebnisse der Stimmungsschwankungsdetektion weiter zu verbessern, sind hier noch weitere Anpassungen der Parameter und Optimierungen notwendig.
Doch nicht nur zur Detektion von Stimmungsschwankungen kann man den SENTI-Wert nutzen. Man kann darüber auch weitere Analysen durchführen und so auch die zurzeit positivsten (Top) bzw. negativsten (Flop) bewerteten Terme einer Region erkennen. Dabei ist zu erkennen, dass es einerseits Terme in diesen Top- bzw. Flop-Tabellen gibt die immer eindeutig als positiv (z.B. „happy“ und „gut“) bzw. negativ (z.B. „traurig“, „sad“ und „krank“) bewertet werden. Aber es treten auch Terme auf, die auf ein signifikantes emotionale Ereignisse hindeuten können die eine Region beschäftigt (z.B. der Tod eines bekannten Künstlers). Vergleicht man diese Top- und Flop-Tabellen der unterschiedlichen Regionen miteinander so fällt auf, dass gleichbedeutende Terme, in ihrer jeweiligen Sprache, emotional sehr ähnlich bewertet werden z.B. „traurig“ und „sad“.
In [12] wurde zudem, mit Hilfe der Errechnung der Standardabweichung des SENTI-Wertes, gezeigt, dass der SENTI-Wert von Termen ganz unterschiedlich stark schwanken kann. Es gibt Terme deren SENTI-Werte kaum schwanken, d.h. deren SENTI-Bewertung gefestigt ist. Dagegen gibt es auch Terme, die relativ stark schwanken z.B. Firmen- bzw. Produktnamen. Beobachtet man diese Werte über die Zeit, so können diese Schwankungen der Werte sichtbar gemacht werden. Diese Schwankungen können wiederum auf Ereignisse zu diesen Zeiten hindeuten, die in Zusammenhang zu diesen Termen stehen. Auch hier besteht noch genügend Raum für weitere Untersuchungen und Entwicklungen.

7 Literaturverzeichnis

[1]. Wikipedia - Check-in. [Online] 23. 08 2013. [Zitat vom: 06. 09 2015.] http://de.wikipedia.org/w/index.php?title=Check-in&oldid=121816221.

[2]. Wikipedia - Media Sharing. [Online] 05. 09 2013. [Zitat vom: 06. 09 2015.] http://de.wikipedia.org/w/index.php?title=Media_Sharing&oldid=122241417.

[3]. Solsman, Joan E. YouTube's Music Key: Can paid streaming finally hook the masses? *cnet.* [Online] 12. 11 2014. [Zitat vom: 17. 11 2015.] http://www.cnet.com/news/youtube-music-key-googles-stab-at-taking-paid-streaming-songs-mainstream/.

[4]. Facebook Reports First Quarter 2015 Results. *Facebook.* [Online] 22. 04 2015. [Zitat vom: 17. 10 2015.] http://investor.fb.com/releasedetail.cfm?ReleaseID=908022.

[5]. Facebook - Like. [Online] [Zitat vom: 06. 09 2015.] https://www.facebook.com/help/452446998120360.

[6]. Facebook. Facebook's Growth In The Past Year. *Facebook.* [Online] 17. 5 2013. [Zitat vom: 06. 09 2015.] https://www.facebook.com/media/set/?set=a.10151908376636729.1073741825.20531316728&type=1.

[7]. Lafferty, Justin. Infographic: What happens in a Facebook minute? *SocialTimes.* [Online] 09. 06 2014. [Zitat vom: 17. 10 2015.] http://www.adweek.com/socialtimes/infographic-what-happens-in-a-facebook-minute-2/299587?red=if.

[8]. Foto von codylagrow. [Online] [Zitat vom: 06. 09 2015.] http://instagram.com/p/issY92nWgo/.

[9]. SLÁNDÁIL - Empowering Emergency Response Systems Using Social Media. [Online] [Zitat vom: 06. 09 2015.] http://slandail.eu.

[10]. Jürgen Nützel, Frank Zimmermann. Improved Burst Based Real-time Event Detection using Adaptive Reference Corpora. *IEEE International Conference on Future Internet of Things and Cloud (FiCloud-2015).* Rom, Italien : s.n., 24. 08 2015. S. 512-518.

[11]. Jürgen Nützel, Frank Zimmermann. Improved Burst Based Real-time Event Detection using Location Dependent Corpora. *3rd International Conference on Future Internet of Things and Cloud (FiCloud 2015).* Rom, Italien : s.n., 24. 08 2015. S. 681-686.

[12]. Jürgen Nützel, Frank Zimmermann. Real-time Language Independent Sentiment Analysis in Social Network. [Hrsg.] Virtual Goods 2012. Namur, Belgien : s.n., 24. 09 2012.

[13]. Xueliang Liu, Raphaël Troncy , Benoit Huet. Using Social Media to Identify Events. *WSM '11 Proceedings of the 3rd ACM SIGMM international workshop on Social media.* Scottsdale, USA : s.n., 30. 11 2011. S. 3-8.

[14]. Phelan, Owen, McCarthy, Kevin und Smyth, Barry. Using Twitter to recommend real-time topical news . [Hrsg.] ACM. *Proceedings of the third ACM conference on Recommender systems .* 10 2009.

[15]. Wikipedia - Information Retrieval. [Online] 11. 08 2013. [Zitat vom: 06. 09 2015.] http://de.wikipedia.org/w/index.php?title=Information_Retrieval&oldid=121416438.

[16]. Kubek, Mario. Dezentrale, kontextbasierte Steuerung der Suche im Internet . [Dissertation zur Erlangung des akademischen Grades Doktor-Ingenieur der Fakultät für Mathematik und Informatik der FernUniversität in Hagen]. 2012.

[17]. Stock, Wolfgang G. Themenentdeckung und-verfolgung und ihr Einsatz bei Informationsdiensten für Nachrichten. *Information – Wissenschaft und Praxis.* 2007, 58, S. 41-46.

[18]. Allan, James. Detection as multi-topic tracking. . [Hrsg.] MA, USA Kluwer Academic Publishers Hingham. *Information Retrieval.* 2002, Bd. 5, 2-3, S. 139-157.

[19]. James Allan, Jaime G. Garbonell, George Doddington, Jonathan Yamron, Yiming Yang. Topic Detection and Tracking Pilot Study Final Report. [Hrsg.] Computer Science Department. In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop : Carnegie Mellon University, 02. 01 1998. S. 341.

[20]. Jeffrey Dean, Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters . *OSDI'04: Sixth Symposium on Operating System Design and Implementation.* San Francisco, CA, USA : s.n., 12 2004.

[21]. Duden - Ereignis. [Online] [Zitat vom: 06. 09 2015.] http://www.duden.de/rechtschreibung/Ereignis.

[22]. Allan, James. Introduction to topic detection and tracking. [Hrsg.] Kluwer Academic Publishers Norwell. *Topic detection and tracking.* s.l. : Boston: Kluwer, S. 1-16.

[23]. Dragomir R. Radev, Hongyan Jing, Małgorzata Stys, Daniel Tam. Centroid-based summarization of multiple documents. *Information Processing & Management.* 2003. 40, S. 919-938.

[24]. Giridhar Kumaran, James Allan. Using Names and Topics for New Event Detection. *Proceeding HLT '05 Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing.* s.l. : Center for Intelligent Information Retrieval Department of Computer Science University of Massachusetts Amherst, 2005. S. 121-128.

[25]. James Allan, Ron Papka, Victor Lavrenko. On-line new event detection and tracking. *Proceeding SIGIR '98 Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval.* 1998. S. 37-45.

[26]. Wikipedia - Cluster (Informatik). [Online] 10. 07 2013. [Zitat vom: 06. 09 2015.] http://de.wikipedia.org/w/index.php?title=Cluster_(Informatik)&oldid=120403235.

[27]. Yiming Yang, Tom Pierce, Jaime Carbonell. A Study on Retrosp ective and On-Line Event Detection. *Proceeding SIGIR '98 Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval.* S. 28-36.

[28]. Gang Luo, Chunqiang Tang, Philip S. Yu. Resource-Adaptive Real-Time New Event Detection. *Proceeding SIGMOD '07 Proceedings of the 2007 ACM SIGMOD international conference on Management of data.* 2007. S. 497-508.

[29]. O'Reilly, Tom. Radar - Web 2.0 Compact Definition: Trying Again. [Online] O'Reilly, 10. 12 2006. [Zitat vom: 06. 09 2015.] http://radar.oreilly.com/2006/12/web-20-compact-definition-tryi.html.

[30]. Richard M. C. Mccreadie, Craig Macdonald , Iadh Ounis. Insights on the Horizon of News Search. [Hrsg.] University of Glasgow. 2010.

[31]. Jannik Strötgen, Michael Gertz. HeidelTime: High Quality Rule-based Extraction and Normalization of Temporal Expressions. [Hrsg.] Association for Computational Linguistics. *Proceedings of the 5th International Workshop on Semantic Evaluation.* 15. 07 2010. S. 321-324.

[32]. Jannik Strötgen, Michael Gertz. Multilingual and cross-domain temporal tagging . *Lang Resources & Evaluation .* s.l. : Springer Science+Business Media B.V. , 08. 05 2012.

[33]. Jannik Strötgen, Michael Gertz. Temporal Tagging on Different Domains: Challenges, Strategies, and Gold Standards . *Proceedings of the Eigth International Conference on Language Resources and Evaluation.* Istanbul, Türkei : s.n., 2012. S. 3746-3753.

[34]. Albert Angel, Nick Koudas, Nikos Sarkas, Divesh Srivastava. What's on the Grapevine ? *SIGMOD'09.* 02. 06 2009.

[35]. Nilesh Bansal, Nick Koudas. BlogScope: A System for Online Analysis of High Volume Text Streams. *Proceeding VLDB '07 Proceedings of the 33rd international conference on Very large data bases.* Wien, Österreich : VLDB Endowment, 23. 09 2007. S. 1410-1413.

[36]. Jagan Sankaranarayanan, Hanan Samet, Benjamin E. Teitler, Michael D. Lieberman, Jon Sperling. TwitterStand: News in Tweets. *Proceeding GIS '09 Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.* 2009. S. 42-51.

[37]. Alan Jackoway, Hanan Samet, Jagan Sankaranarayanan. Identification of Live News Events using Twitter. *Proceeding LBSN '11 Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social.* S. 25-32.

[38]. Thomas Ertl, Junghoon Chae, Ross Maciejewski, Harald Bosch, Dennis Thom, Yun Jang, David S. Ebert. Spatiotemporal Social Media Analytics for Abnormal Event Detection and Examination using Seasonal-Trend Decomposition. *Proceeding VAST '12 Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology (VAST).* 2012. 143-152.

[39]. Ryong Lee, Shoko Wakamiya, Kazutoshi Sumiya. Discovery of unusual regional social activities using geo-tagged microblogs. *Journal World Wide Web archive.* 07 2011. Bd. 14, 4, S. 321-349.

[40]. Alexei Pozdnoukhov, Christian Kaiser. Space-Time Dynamics of Topics in Streaming Text. *Proceeding LBSN '11 Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social.* 2011. S. 1-8.

[41]. Takeshi Sakaki, Makoto Okazaki, Yutaka Matsuo. Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. *Proceedings of the Nineteenth International WWW Conference (WWW2010).* 2010.

[42]. Sungjun Lee, Sangjin Lee, Kwanho Kim, Jonghun Park. Bursty Event Detection from Text Streams for Disaster Management. *Proceeding WWW '12 Companion Proceedings of the 21st international conference companion on World Wide Web.* 2012. S. 697-682.

[43]. Bertrand De Longueville, Robin S. Smith, Gianluca Luraschi. "OMG, from here, I can see the flames!": a use case of mining Location Based Social Networks to acquire spatio-temporal data on forest fires. *Proceeding LBSN '09 Proceedings of the 2009 International Workshop on Location Based Social.* 2009. S. 73-80.

[44]. Rui LI, Kin Hou Lei, Ravi Khadiwala, Kevin Chen-Chuan Chang. TEDAS: a Twitter Based Event Detection and Analysis System. [Hrsg.] IEEE Computer Society. *ICDE.* 2012. S. 1273-1276.

[45]. Dennis Thom, Harald Bosch, Steffen Koch, Michael Wörner, Thomas Ertl. Spatiotemporal Anomaly Detection through Visual Analysis of Geolocated Twitter Messages . [Hrsg.] 2012 IEEE Pacific Visualization Symposium (PacificVis). 28. 02 2012. S. 41-48.

[46]. Brett Meyer, Kevin Bryan, Yamara Santos, Beomjin Kim. TwitterReporter: Breaking News Detection and Visualization through the Geo-Tagged Twitter Network . *CATA : ISCA.* 2011. S. 84-89.

[47]. Stemming and lemmatization. [Online] 2009. [Zitat vom: 06. 09 2015.] http://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html.

[48]. Wikipedia - Bayesscher Filter. [Online] 30. 11 2014. [Zitat vom: 06. 09 2015.] http://de.wikipedia.org/w/index.php?title=Bayesscher_Filter&oldid=136329691.

[49]. Hila Becker, Mor Naaman, Luis Gravano. Beyond Trending Topics: Real-World Event Identification on Twitter . [Hrsg.] 5. ICWSM 2011. Barcelona, Spanien : s.n., 2011.

[50]. Swit Phuvipadawat, Tsuyoshi Murata. Breaking News Detection and Tracking in Twitter. *Proceeding WI-IAT '10 Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology.* 2010. Bd. 3, S. 120-123.

[51]. M. Osborne, S. Petrovic, R. McCreadie, C. Macdonald, I. Ounis. Bieber no more: First Story Detection using Twitter and Wikipedia. *SIGIR 2012 Workshop on Time-aware Information Access.* 2012.

[52]. Hassan Syyadi, Matthew Hurst, Alexey Maykov. Event Detection and Tracking in Social Streams. *Proceedings of International Conference on Weblogs and Social Media (ICWSM).* 2009.

[53]. KeyGraph: A Graph Analytical Approach For Fast Topic Detection. *CodePlex.* [Online] [Zitat vom: 06. 09 2015.] http://keygraph.codeplex.com/.

[54]. Nominalphrase. *ProGr@mm.* [Online] 13. 11 2009. [Zitat vom: 06. 09 2015.] http://hypermedia.ids-mannheim.de/call/public/gruwi.ansicht?v_id=1625.

[55]. Saša Petrović, Miles Osborne, Victor Lavrenko. Streaming First Story Detection with application to Twitter. *Proceeding HLT '10 Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics.* 2010. S. 181-189.

[56]. Chien Chin Chen, Yao-Tsung Chen, Yeali Sun, Meng Chang Chen. Life Cycle Modeling of News Events Using Aging Theory . [Hrsg.] Cavtat-Dubrovnik, Croatia, September 22-26, 2003. Proceedings 14th European Conference on Machine Learning. *Machine Learning: ECML 2003.* 2003. S. 47-59.

[57]. David M. Blei, Andrew Y. Ng, Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research 3.* 02 2002. S. 993-1022.

[58]. Blei, David M. Probabilistic topic models . *communicationS of the acm .* 04 2012, Bd. 55, 4, S. 77-84.

[59]. Kirill Kireyev, Leysia Palen, Kenneth M. Anderson. Applications of Topics Models to Analysis of Disaster-Related Twitter Data. *NIPS Workshop on Applications for Topic Models: Text and Beyond.* 11. 12 2009.

[60]. Jianshu Weng, Bu-Sung Lee. Event Detection in Twitter. [Hrsg.] HP Laboratories. 06. 07 2011.

[61]. Benhardus, James. Streaming Trend Detection in Twitter . [Hrsg.] Inderscience Publishers. *Journal International Journal of Web Based Communities.* 01 2013. S. 122-139.

[62]. Sasa Petrovic, Miles Osborne, Victor Lavrenko. The Edinburgh Twitter Corpus. *Proceeding WSA '10 Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media.* 2010. S. 25-26.

[63]. Michael Mathioudakis, Nick Koudas. TwitterMonitor: Trend Detection over the Twitter Stream . *Proceeding SIGMOD '10 Proceedings of the 2010 ACM SIGMOD International Conference on Management of data.* 2010. S. 1155-1158.

[64]. Wikipedia - Kookkurrenz. [Online] 04. 04 2013. [Zitat vom: 06. 09 2015.] http://de.wikipedia.org/w/index.php?title=Kookkurrenz&oldid=116993496.

[65]. Mario Cataldi, Luigi Di Caro, Claudio Schifanella. Emerging Topic Detection on Twitter based on Temporal and Social Terms Evaluation . *Proceeding MDMKDD '10 Proceedings of the Tenth International Workshop on Multimedia Data Mining.* 2010.

[66]. Kaufmann, Max. Syntactic Normalization of Twitter Messages . *International conference on natural language processing.* 29. 07 2010.

[67]. Kazufumi Watanabe, Masanao Ochi, Makoto Okabe, Rikio Onai. Jasmine: A Real-time Local-event Detection System based on Geolocation Information Propagated to

Microblogs. *Proceeding CIKM '11 Proceedings of the 20th ACM international conference on Information and knowledge management.* 2011. S. 2541-2544.

[68]. Foursquare. [Online] [Zitat vom: 06. 09 2015.] https://de.foursquare.com/.

[69]. Ryong Lee, Kazutoshi Sumiya. Measuring Geographical Regularities of Crowd Behaviors for Twitter-based Geo-social Event Detection . *Proceeding LBSN '10 Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks.* 2010. S. 1-10.

[70]. Weisstein, Eric W. Voronoi Diagram. [Online] [Zitat vom: 06. 09 2015.] http://mathworld.wolfram.com/VoronoiDiagram.html.

[71]. Augusto Dias Pereira dos Santos, Leandro Krug Wives, Luis Otavio Alvares. Location-Based Events Detection on Micro-Blogs . [Hrsg.] Cornell University Library. 15. 10 2012. arXiv:1210.4008.

[72]. Adam Bermingham, Alan F. Smeaton. Crowdsourced Real-world Sensing: Sentiment Analysis and the Real-Time Web. *AICS 2010 - Sentiment Analysis Workshop at Artificial Intelligence and Cognitive Science.* 30. 08 2010.

[73]. Read, Jonathon. Using Emoticons to reduce Dependency in Machine Learning Techniques for Sentiment Classification . *Proceeding ACLstudent '05 Proceedings of the ACL Student Research Workshop.* 2005. S. 43-48.

[74]. Nicholas A. Diakopoulos, David A. Shamma. Characterizing Debate Performance via Aggregated Twitter Sentiment. *Proceeding CHI '10 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* 2010. S. 1195-1198.

[75]. Amazon. Amazon Web Services. [Online] [Zitat vom: 06. 09 2015.] http://aws.amazon.com/de/.

[76]. Dmitry Davidov, Oren Tsur, Ari Rappoport. Enhanced Sentiment Learning Using Twitter Hashtags and Smileys. *Proceeding COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics: Posters.* 2010. S. 241-249.

[77]. Johan Bollen, Alberto Pepe, Huina Mao. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. . [Hrsg.] AAAI Press. *The Fifth International AAAI Conference on Weblogs and Social Media (ICWSM-11).* Barcelona, Spanien : s.n., 17. 07 2011.

[78]. Wikipedia - Emoticon. [Online] 21. 12 2013. [Zitat vom: 06. 09 2015.] http://de.wikipedia.org/w/index.php?title=Emoticon&oldid=125653040.

[79]. Akshi Kumar, Teeja Mary Sebastian. Sentiment Analysis on Twitter . *IJCSI International Journal of Computer Science Issues.* 07 2012.

[80]. Akshat Bakliwal, Piyush Arora, Senthil Madhappan, Nikhil Kapre, Mukesh Singh, Vasudeva Varma. Mining Sentiments from Tweets. *3rd Workshop on Sentiment and Subjectivity Analysis (WASSA) in Conjunction with 50th annual meeting of Association for Computational Linguistics (ACL) 2012.* 07 2012.

[81]. Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Own Rambow, Rebecca Passonneau. Sentiment Analysis of Twitter Data . *Proceeding LSM '11 Proceedings of the Workshop on Languages in Social Media.* 2011. S. 30-38.

[82]. Wikipedia - Twitter. [Online] 28. 06 2013. [Zitat vom: 06. 09 2015.] http://de.wikipedia.org/w/index.php?title=Twitter&oldid=120020039.

[83]. Wickre, Karen. Celebrating #Twitter7. [Online] 21. 03 2013. [Zitat vom: 06. 09 2015.] https://blog.twitter.com/2013/celebrating-twitter7.

[84]. Akshay Java, Tim Finin, Xiaodan Song, Belle Tseng. Why We Twitter: Understanding Microblogging Usage and Communities. *Procedings of the Joint 9th WEBKDD and 1st SNA-KDD Workshop.* 12. 08 2007.

[85]. Andreas M. Kaplan, Michael Haenlein. The early bird catches the news: Nine things you should know about micro-blogging. [Hrsg.] Business Horizons. 2011. Bd. 54, S. 105-113.

[86]. Parker, Ashley. The New York Times - Twitter's Secret Handshake. [Online] 10. 06 2011. [Zitat vom: 06. 09 2015.] http://www.nytimes.com/2011/06/12/fashion/hashtags-a-new-way-for-tweets-cultural-studies.html?pagewanted=all&_r=1&.

[87]. Pfeiffer, Thomas. #PB21 Web 2.0 in der politischen Bildung - Live-Feedback einbinden mit einer Twitterwall. [Online] 31. 05 2011. [Zitat vom: 06. 09 2015.] http://pb21.de/2011/05/live-feedback-einbinden-mit-einer-twitterwall/.

[88]. Windows - E-Mail und soziale Netzwerke. [Online] [Zitat vom: 06. 09 2015.] http://windows.microsoft.com/de-de/windows-8/people-app#1TC=t1.

[89]. How-to People + email. [Online] [Zitat vom: 06. 09 2015.] http://www.windowsphone.com/en-us/how-to/wp7/people/people-hub.

[90]. Wikipedia - OS X 10.8. [Online] 22. 06 2015. [Zitat vom: 06. 09 2015.] https://de.wikipedia.org/w/index.php?title=OS_X_10.8&oldid=143463298.

[91]. Viticci, Federico. iOS 5: Twitter Integration. *Mac Stories.* [Online] 13. 10 2011. [Zitat vom: 06. 09 2015.] http://www.macstories.net/stories/ios-5-twitter-integration/.

[92]. Stone, Biz. Changes for Some SMS Users—Good and Bad News. [Online] 13. 08 2008. [Zitat vom: 06. 09 2015.] https://blog.twitter.com/2008/changes-some-sms-users%E2%80%94good-and-bad-news.

[93]. Balachander Krishnamurthy, Phillipa Gill, Martin Arlitt. A few chirps about twitter. [Hrsg.] NY, USA ©2008 ACM New York. *WOSN '08 Proceedings of the first workshop on Online social networks.* 2008. S. 19-24.

[94]. The Twitter REST API. [Online] [Zitat vom: 06. 09 2015.] https://dev.twitter.com/docs/api.

[95]. Twitter - The Streaming APIs. [Online] 24. 09 2012. [Zitat vom: 06. 09 2015.] https://dev.twitter.com/docs/streaming-apis.

[96]. Walcher, Stephan. Bing und Twitter verlängern ihre Zusammenarbeit über Twitter. [Online] 07. 09 2011. [Zitat vom: 06. 09 2015.] http://www.prometeo.de/2011/09/bing-und-twitter-verlaengern-ihre-zusammenarbeit-ueber-twitter/.

[97]. *Twitter Storm: Verteiltes Rechnen in Echtzeit.* [Online] 28. 09 2011. [Zitat vom: 06. 09 2015.] http://www.heise.de/open/meldung/Twitter-Storm-Verteiltes-Rechnen-in-Echtzeit-1351107.html.

[98]. *hadoop.* [Online] [Zitat vom: 06. 09 2015.] http://hadoop.apache.org/.

[99]. republica. re:publica. [Online] [Zitat vom: 06. 09 2015.] http://www.re-publica.de/.

[100]. *A Storm is coming: more details and plans for release.* [Online] 04. 08 2011. [Zitat vom: 06. 09 2015.] https://blog.twitter.com/2011/storm-coming-more-details-and-plans-release.

[101]. S4 distributed stream computing platform. [Online] [Zitat vom: 06. 09 2015.] http://incubator.apache.org/s4/.

[102]. Spark - Lightning-Fast Cluster Computing. [Online] [Zitat vom: 06. 09 2015.] http://spark-project.org/.

[103]. Das, Tathagata. Spark Streaming - Large-scale near-real-time stream processing. [Online] 26. 02 2013. [Zitat vom: 06. 09 2015.] http://spark-project.org/talks/strata_spark_streaming.pdf.

[104]. Matei Zaharia, Tathagata Das, Haoyuan Li, Scott Shenker, Ion Stoica. *Discretized Streams: An Efficient and Fault-Tolerant Model for Stream Processing on Large Clusters.* University of California, Berkeley. s.l. : HotCloud 2012, 2012.

[105]. Lubbadeh, Jens. Seismograf der Welt. *Technlogy Review.* 02 2013, S. 52-55.

[106]. Google Earth. [Online] Google. [Zitat vom: 06. 09 2015.] http://www.google.de/intl/de/earth/.

[107]. Kuwait Has Most Twitter Users Per Capita. *The Realtime Report.* [Online] 01 2014. [Zitat vom: 14. 11 2015.] http://therealtimereport.com/2014/01/14/kuwait-has-most-twitter-users-per-capita/.

[108]. Brian McClanahan, Swapna S. Gokhale. Location Inference of Social Media Posts at Hyper-Local Scale. *2015 3rd International Conference on Future Internet of Things and Cloud.* Rom : s.n., 24. 08 2015.

[109]. Aktuelles Korpusarchiv. *Institut für deutsche Sprache.* [Online] [Zitat vom: 14. 11 2015.] http://www1.ids-mannheim.de/kl/projekte/korpora/archiv.html.

[110]. Oracle und Java. [Online] [Zitat vom: 06. 09 2015.] http://www.oracle.com/de/technologies/java/overview/index.html.

[111]. MySQL - Die populärste Open-Source Datenbank der Welt. [Online] [Zitat vom: 06. 09 2015.] http://www.mysql.de/.

[112]. Crockford, D. The application/json Media Type for JavaScript Object Notation (JSON). [Online] [Zitat vom: 06. 09 2015.] http://tools.ietf.org/html/rfc4627.

[113]. Twitter4J - A Java library for the Twitter API. [Online] [Zitat vom: 06. 09 2015.] http://twitter4j.org/en/index.html.

[114]. *Wikipedia - Boswash.* [Online] 29. 12 2013. [Zitat vom: 06. 09 2015.] http://de.wikipedia.org/w/index.php?title=Boswash&oldid=125908194.

[115]. FAQs about the Tweet location feature. [Online] [Zitat vom: 06. 09 2015.] https://support.twitter.com/articles/78525-about-the-tweet-location-feature.

[116]. Tweets. *Twitter Developers.* [Online] [Zitat vom: 30. 11 2015.] https://dev.twitter.com/overview/api/tweets.

[117]. ARCHIVED: What is a troll? [Online] 03. 01 2013. [Zitat vom: 06. 09 2015.] https://kb.iu.edu/d/afhc.

[118]. Open Data Structures. [Online] [Zitat vom: 06. 09 2015.] http://opendatastructures.org/versions/edition-0.1e/ods-java/5_Hash_Tables.html.

[119]. KML Documentation Introduction. [Online] [Zitat vom: 06. 09 2015.] https://developers.google.com/kml/documentation/.

[120]. Bootstrap. [Online] [Zitat vom: 06. 09 2015.] http://getbootstrap.com/.

[121]. jQuery. [Online] [Zitat vom: 06. 09 2015.] http://jquery.com/.

[122]. Farzindar Atefeh, Wael Khreich. A Survey of Techniques for Event Detection in Twitter. *Computational Intelligence.* 2012.

[123]. Wikipedia - Beurteilung eines Klassifikators. [Online] 23. 09 2014. [Zitat vom: 06. 09 2015.] http://de.wikipedia.org/w/index.php?title=Beurteilung_eines_Klassifikators&oldid=134284242.

[124]. CBS New York. [Online] 12. 03 2014. [Zitat vom: 06. 09 2015.] http://newyork.cbslocal.com/2014/03/12/explosion-reported-at-harlem-building/.

[125]. USA Today. [Online] 12. 03 2014. [Zitat vom: 06. 09 2015.] http://www.usatoday.com/story/news/nation/2014/03/12/fire-san-francisco/6315437/.

[126]. L. Burks, M. Miller, R. Zadeh. Rapid esitimate of ground shaking intensity by combining simple earthquake characteristics with tweets. *Proceedings of the 10th National Conference in Earthquake Engineering.* 21. 07 2014.

[127]. Wikipedia - Amoklauf an der Sandy Hook Elementary School. [Online] 04. 01 2014. [Zitat vom: 06. 09 2015.] http://de.wikipedia.org/w/index.php?title=Amoklauf_an_der_Sandy_Hook_Elementary_School&oldid=126105999.

[128]. Tweet von Rickbryce an NBCConnecticut. [Online] 14. 12 2012. [Zitat vom: 06. 09 2015.] https://twitter.com/Rickbryce/status/279596403444031489.

[129]. Tweet von NBCConnecticut - Polizei unterwegs. [Online] 14. 12 2012. [Zitat vom: 06. 09 2015.] https://twitter.com/NBCConnecticut/status/279600560364220417.

[130]. Tweet von NBCConnecticut - Polizei berichtet von Schießerei. [Online] 14. 12 2012. [Zitat vom: 06. 09 2015.] https://twitter.com/NBCConnecticut/status/279604931042889728.

[131]. Tweet von Reuters - Schießerei an Schule. [Online] 14. 12 2012. [Zitat vom: 06. 09 2015.] https://twitter.com/Reuters/status/279618497997332481.

[132]. Spiegel Online - Connecticut: 27 Menschen sterben bei Schießerei an US-Grundschule. [Online] 14. 12 2012. [Zitat vom: 06. 09 2015.] http://www.spiegel.de/panorama/justiz/schiesserei-an-grundschule-in-connecticut-mit-vielen-toten-a-873054.html.

[133]. stern.de - Unfassbare Tat in US-Grundschule. [Online] 14. 12 2012. [Zitat vom: 06. 09 2015.] http://www.stern.de/panorama/dutzende-tote-unfassbare-tat-in-us-grundschule-1942416.html.

[134]. Focus Online - Szenen des Schreckens an der Schule von Newtown. [Online] [Zitat vom: 06. 09 2015.] http://www.focus.de/panorama/welt/amoklauf-an-grundschule-in-connecticut-szenen-des-schreckens-an-der-schule-von-newtown_aid_882395.html.

[135]. Wikipedia - Anschlag auf den Boston-Marathon. [Online] 21. 04 2014. [Zitat vom: 06. 09 2015.] http://de.wikipedia.org/w/index.php?title=Anschlag_auf_den_Boston-Marathon&oldid=129707848.

[136]. Wikipedia - Boston-Marathon. [Online] 21. 04 2014. [Zitat vom: 06. 09 2015.] http://de.wikipedia.org/w/index.php?title=Boston-Marathon&oldid=129713910.

[137]. Spiegel Online - USA: Explosionen beim Boston-Marathon - drei Tote, hundert Verletzte. [Online] 15. 04 2013. [Zitat vom: 06. 09 2015.] http://www.spiegel.de/panorama/explosionen-beim-boston-marathon-a-894525.html.

[138]. Tweet - ReutersUS berichtet Sperrung des Boston-Marathon Hauptquartiers. [Online] 15. 04 2013. [Zitat vom: 06. 09 2015.] https://twitter.com/ReutersUS/status/323872460195909632.

[139]. Tweet - WCVB berichtet von Explosion nahe der Ziellinie. [Online] 15. 04 2013. [Zitat vom: 06. 09 2015.] https://twitter.com/WCVB/status/323873688124522497.

[140]. Tweet - cnnbrk berichtet über Explosion an der Ziellinie. [Online] 15. 04 2013. [Zitat vom: 06. 09 2015.] https://twitter.com/cnnbrk/status/323875187563053056.

[141]. stern.de - Explosionen bei Marathon in Boston: Verletzte. [Online] 15. 04 2013. [Zitat vom: 06. 09 2015.] http://www.stern.de/politik/ausland/anschlagsserie-in-boston-tote-und-verletzte-nach-explosionen-beim-marathon-3022540.html.

[142]. Focus online - Blutbad beim Boston-Marathon offenbar durch Bomben verursacht. [Online] 15. 04 2013. [Zitat vom: 06. 09 2015.]

http://www.focus.de/panorama/welt/usa-blutbad-beim-boston-marathon-offenbar-durch-bomben-verursacht_aid_961146.html.

[143]. Holland, Martin. heise.de - Quakebot schreibt erste Meldung zum Erdbeben in Los Angeles. [Online] 18. 03 2014. [Zitat vom: 06. 09 2015.] http://www.heise.de/newsticker/meldung/Quakebot-schreibt-erste-Meldung-zum-Erdbeben-in-Los-Angeles-2149156.html.

[144]. Oremus, Will. slate.com - The First News Report on the L.A. Earthquake Was Written by a Robot. [Online] [Zitat vom: 06. 09 2015.] http://www.slate.com/blogs/future_tense/2014/03/17/quakebot_los_angeles_times_robot_journalist_writes_article_on_la_earthquake.html.

[145]. Wikipedia - Hurrikan Sandy. [Online] 12. 05 2014. [Zitat vom: 06. 09 2015.] http://de.wikipedia.org/w/index.php?title=Hurrikan_Sandy&oldid=130326940.

[146]. Stefan Porteck, Volker Briegleb, Herbert Braun, Jan-Keno Janssen. Facebook kauft WhatsApp. [Online] 19. 02 2014. [Zitat vom: 06. 09 2015.] http://heise.de/-2118920.

[147]. Jeffrey Dean, Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *OSDI'04: Sixth Symposium on Operating System Design and Implementation.* San Francisco, CA, USA : s.n., 12 2004.

[148]. Certified Products. [Online] 17. 06 2013. [Zitat vom: 06. 09 2015.] https://dev.twitter.com/programs/twitter-certified-products/products.

[149]. *Open Source at Twitter.* [Online] [Zitat vom: 06. 09 2015.] https://engineering.twitter.com/opensource.

[150]. Twitter. Twitter. [Online] [Zitat vom: 06. 09 2015.] https://www.twitter.com.

[151]. What is Twitter? *Twitter.* [Online] [Zitat vom: 06. 09 2015.] https://business.twitter.com/de/basics/what-is-twitter/.

[152]. Wikipedia - Textkorpus. [Online] 09. 04 2014. [Zitat vom: 06. 09 2015.] http://de.wikipedia.org/w/index.php?title=Textkorpus&oldid=129344352.

[153]. Ritterbusch, S. Die Mathematik des Bayes Spamfilters. *Karlsruher Institut für Technologie - Fakultät für Mathematik.* [Online] [Zitat vom: 06. 09 2015.] http://www.math.kit.edu/ianm4/~ritterbusch/seite/spam/de.

Anlage 1

8 Erklärung

Ich versichere, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Weitere Personen waren an der inhaltlich-materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs bzw. Beratungsdiensten (Promotionsberater oder anderer Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer Prüfungsbehörde vorgelegt.

Ich bin darauf hingewiesen worden, dass die Unrichtigkeit der vorstehenden Erklärung als Täuschungsversuch bewertet wird und gemäß § 7 Abs. 10 der Promotionsordnung den Abbruch des Promotionsverfahrens zur Folge hat.

Weinbergen, den 01.03.2016