Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit Nr. 161

Visuelle Analyse von Social Media Informationsdiffusion für den Zivilschutz

Heiko Zimmermann

Studiengang:	Informatik
Prüfer/in:	Prof. Dr. Thomas Ertl
Betreuer/in:	Dipl.-Inf. Dennis Thom, Robert Krüger, M.Sc.
Beginn am:	23. Juli 2014
Beendet am:	29. Januar 2014
CR-Nummer:	H.5.1, H.5.2, I.3.6

Kurzfassung

In den letzten Jahren haben vor allem soziale Medien zunehmend Einzug in das Leben unserer Internetgesellschaft gehalten. Dabei generieren Benutzer sogenannter sozialer Netzwerke eine Vielzahl an Daten, die untereinander ausgetauscht und der Öffentlichkeit zugänglich gemacht werden. Richtig gefiltert und ausgewertet stellen diese Daten wichtige Informationsquellen dar, welche in der Marktforschung, der Bekämpfung von Kriminalität und nicht zuletzt beim Katastrophenschutz Verwendung finden. Gerade bei Letzterem ist es oftmals notwendig, die gewonnenen Informationen auf einer Karte zu verorten. Dies geschieht meist über GPS-Koordinaten, die den Nachrichten angehängt sind. Von besonderem Interesse sind dabei neben den Inhalten der Nachrichten auch die Verbreitungswege der enthaltenen Informationen. In dieser Bachelorarbeit werden, ausgehend von einem Workshop mit Domäneexperten, Konzepte zur Gewinnung und Visualisierung von Kommunikationsnetzwerken, am Beispiel von Twitter, entwickelt. Dabei dienen als Datengrundlage sowohl online verfügbare Twitterdaten, als auch ein zuvor gesammelter, lokal gespeicherter Datensatz, der zuvor auf enthaltene Kommunikationsnetzwerke untersucht wird. Zur Visualisierung der Daten auf einer Karte wird ein Knoten-Kanten-Diagramm und eine, aus den Kommunikationsnetzwerken erzeugte, Heatmap verwendet. Bei einem abschließenden Anwendungsfall wird ein Datensatz, mit Hilfe der Heatmap, auf dessen Verbreitungswege untersucht.

Abstract

In recent years, especially social media have increasingly found their way into the live of our online community. During this process users of so-called social networks generate a variety of data which is shared and published to the public. Filtered and evaluated the right way this data is an important source of information that is used in market research, crime fighting and finally disaster management. Especially for the latter, it is often necessary to locate the information obtained on a map. This is usually done through GPS coordinates that are attached to the messages. Beside the content of the messages, especially the way of distribution of the information contained in the messages are of particular interest. In this thesis, starting from a workshop with domain experts, concepts for the acquisition and visualization of communication networks, exemplified by Twitter, are developed. As a data source online available Twitter data, as well as previously collected local data that is studied beforehand, is used. For visualizing the data on a map a node-edge diagram and a heat map which is generated from communication networks is used. In a final use case a data set is analyzed for its diffusion using the heat map.

Inhaltsverzeichnis

Abbildungsverzeichnis

Verzeichnis der Listings

1 Einleitung

Seit dem Aufkommen des Internets befindet sich dieses im ständigen Wandel. Dabei entwickelte es sich von einem eher passiven Medium, das von wenigen Experten genutzt wurde, zu einem interaktiven Medium, das fest in der Gesellschaft verankert ist. Heute findet ein Großteil der weltweiten Kommunikation über das Internet statt. Die meisten Menschen nutzen es täglich um Informationen einzuholen, mit anderen Menschen zu kommunizieren und Daten auszutauschen. Dabei haben sich in den letzten Jahren besonders sogenannte soziale Netzwerke hervorgetan. In einer Arbeit zu sozialen Medien verweisen Kaplan und Haenlein [KH10] auf Angaben von Forrester Research, nach denen 75% aller Internetnutzer in der zweiten Jahreshälfte 2008 soziale Medien benutzten, 2007 waren es noch 56%. Soziale Medien erlauben es Benutzern untereinander in Kontakt zu treten und Informationen und Inhalte aller Art auszutauschen. Mit der zunehmenden Digitalisierung unseres Lebens sinkt dabei auch die Hemmschwelle der Nutzer, private Daten preiszugeben und Anderen zugänglich zu machen.

In den letzten Jahren hat diese Flut an Daten enorme Ausmaße angenommen. Um so mehr ist es daher eine wichtige Aufgabe, diese Daten zugänglich und verwertbar zu machen. Aus sozialen Medien gewonnene Informationen sind neben der Wirtschafts- und Marketingbranche auch für den Zivil- und Katastrophenschutz von großem Interesse. Eine Studie von Qu et al. [QHZZ11] belegt, dass Nutzer während Krisensituationen verstärkt relevante Informationen und Lageeinschätzungen veröffentlichen und weiterverbreiten. Innerhalb der Nachrichten werden zudem verstärkt Angaben zur geografischen Position getätigt. Informationen wie diese können richtig verwertet, entscheidend zur Entscheidungsfindung, Koordination der Einsatzkräfte und damit letzlich zur Sicherheit der Menschen beitragen. Neben dem Finden und Analysieren einzelner relevanter Nachrichten sind jedoch auch deren Verbreitungswege, bzw. die Diffusion der enthaltenen Informationen, von großem Interesse. Das Wissen über die Verbreitung von Informationen kann unter anderem Aufschluss über die Informationslage in bestimmten Regionen gewähren. Umgekehrt können jedoch auch Einsichten über die Ausbreitung von Fehlinformationen erlangt und deren Quellen ausfindig gemacht werden.

1.1 Aufgabenstellung und Lösungsansatz

Diese Arbeit beschäftigt sich mit der Analyse der Informationsdiffusion in sozialen Netzwerken im Zuge des Zivil- und Katastrophenschutzes. Anhand von Nachrichten des sozialen Netzwerks Twitter, sollen hierbei Kommunikationsnetzwerke gewonnen und Techniken zur Visualisierung der Nachrichtenverbreitung und Informationsdiffusion entwickelt werden. Dabei interessiert neben der topologischen und zeitlichen Ausbreitung von Nachrichten vor allem deren geografische Verbreitung. Die entwickelten Konzepte sollen anschließend prototypisch in einer Anwendung implementiert werden. Des Weiteren sollen bereits vorhandene Twitterdaten auf Konversationsnetzwerke und deren

geografischen Verbreitung analysiert werden. Ausgangspunkt stellt ein Workshop mit Domäneexperten, aus verschiedenen Bereichen eines Krisenstabs, dar.

Um die Informationsverbreitung von Nachrichten des Microblogging-Diensts Twitter untersuchen zu können, müssen die zugrundeliegenden Kommunikationsnetzwerke zunächst extrahiert werden. Diese Kommunikationsnetzwerke enthalten neben den ursprünglichen Nachricht alle Nachrichten, die sich direkt oder indirekt über andere Nachrichten auf diese beziehen. Sie werden daher in dieser Arbeit als Konversationen bezeichnet. Einzelne Konversationen werden anschließend in einer Baumstruktur dargestellt. Die als Konversationsbäume bezeichnet werden. Zur Visualisierung dieser Konversationsbäume auf einer Karte wurde ein Knoten-Kanten-Diagramm in Kombination mit einer Heatmap verwendet. Dabei besteht die Heatmap aus gerichteten Splats, die durch einen zugrundeliegenden Gaußkernel erzeugt werden. Bei einem abschließend betrachteten Anwendungsfall stellte sich dieser Ansatz als nützlich heraus, um Richtungen auf der Karte zu visualisieren. Allerdings weisen die zugrundeliegenden Daten nur bedingt Muster in ihrer geografischen Verbreitung auf.

1.2 Struktur der Arbeit

Dieses Dokument ist in sieben Kapitel aufgeteilt, welche jeweils verschiedene Aspekte der Arbeit behandeln. Ausgehend von dieser Einleitung werden in Kapitel 2 **verwandte Arbeiten** ähnlicher Thematik vorgestellt. Kapitel 3 behandelt anschließend die allgemeinen **Grundlagen**, die im späteren Verlauf der Arbeit Verwendung finden. Im Verlauf des 4. Kapitels werden die **Konzepte**, Vorgehensweisen und Überlegungen, die von der Gewinnung der Daten bis hin zu deren Visualisierung angestellt wurden, näher erläutert. Kapitel 5 behandelt die **Umsetzung** dieser Konzepte. Im Zuge dessen wird die Benutzeroberfläche und Funktionsweise der Applikation sowie einzelne Aspekte der Implementierung erläutert. Des Weiteren werden, die im Rahmen der Datenfilterung gewonnenen Daten und Erkenntnisse, vorgestellt. Zuletzt wird ein möglicher **Anwendungsfall** vorgestellt und in einem abschließenden Kapitel eine **Zusammenfassung** gegeben.

2 Verwandte Arbeiten

In den letzten Jahren haben sich viele Arbeiten mit sozialen Medien und deren Einsatz in Krisensituationen beschäftigt. Hierzu durchgeführte Studien weisen darauf hin, dass Nachrichten sozialer Netzwerke, besonders sogenannter Mikroblogging-Dienste, zur besseren Einschätzung der Situation vor Ort beitragen können. Eine Studie von Qu et al. [QHZZ11] untersuchte Nachrichten des in China beliebten Mikroblogging-Dienstes Sina-Weibo, die während des Yushu Erdbebens 2010 abgesetzt wurden. Die Studie zeigte, dass große Mengen an Nachrichten direkt nach den Beben abgesetzt wurden, welche Einschätzungen und Beschreibungen zur aktuellen Situation vor Ort beinhalteten. Darüber hinaus wurde festgestellt, dass sich diese Nachrichten schneller innerhalb des sozialen Netzwerks ausbreiteten, als andere sich ebenfalls auf die Katastrophe beziehenden Nachrichten. Yin et al. [YLC+12] zeigten deutliche Korrelationen zwischen dem Tweetaufkommen und einzelnen Beben in Christchurch, Neuseeland, auf. Tweets dieser Region enthielten ebenfalls verstärkt Lage- und Schadenseinschätzungen sowie Hilferufe. Weitere Untersuchungen von Vieweg et al.[VHSP10] widmeten sich den Nachrichten, die während einer Flut am Red River und Buschfeuern in Oklahoma auf Twitter abgesetzt wurden. Demnach setzten 78% der Nutzer des Flut-Datensatzes und 86% der

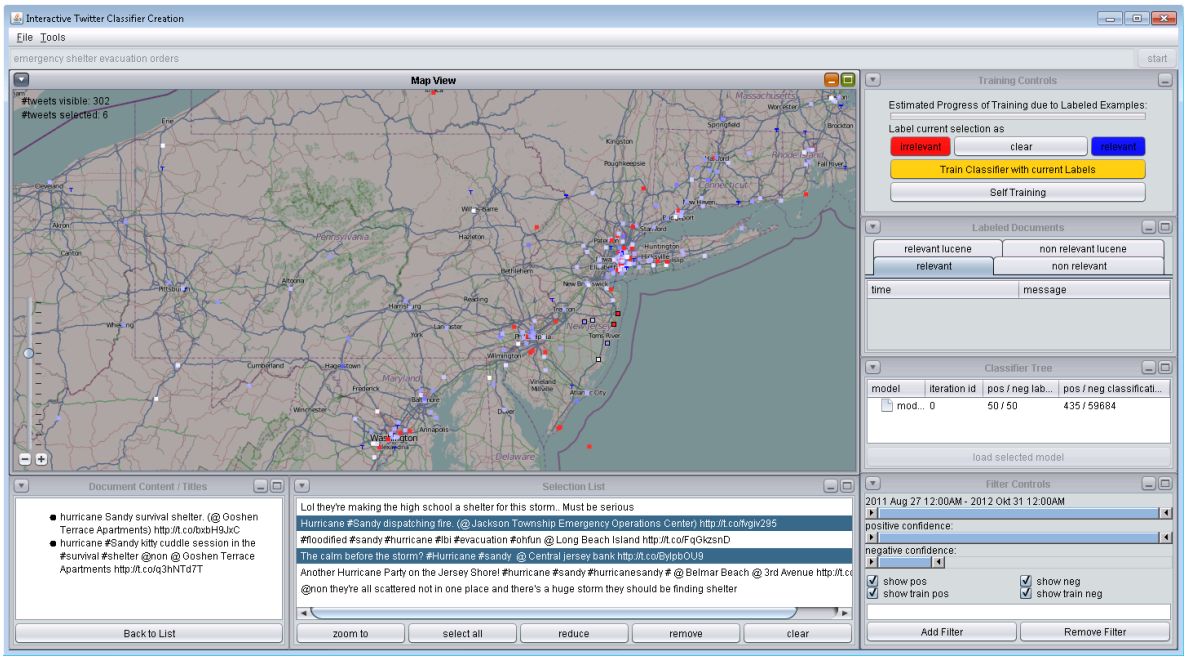


Abbildung 2.1: ScatterBlogs-Umgebung zum Erstellen und Trainieren von Klassifikatoren. Bild von Thom et al. [BTH+13]

Nutzer des Buschfeuer-Datensatzes mindestens einen Tweet mit einer Angabe zur geografischen Position ab. Lokale Benutzer scheinen demnach Informationen zur Position in Notfallsituationen als wichtig zu erachten und dementsprechend öfter anzuhängen.

Nach Erkenntnissen solcher Studien, speziell Qu et al.[QHZZ11], lassen sich relevante situationsbezogene Informationen von Inhalt und Aufbau, sowie dem Kontext in welchem diese auftreten meist von der großen Menge der täglich abgesetzten Nachrichten unterscheiden und weiter klassifizieren. Man spricht daher oft von der Detektion von Anomalien. Aufbauend auf genannte Erkenntnisse wurden in verschiedenen Arbeiten bereits Werkzeuge und Anwendungen zur Detektion solcher Anomalien innerhalb sozialer Netzwerke entwickelt. Dabei wird meist auf Techniken des maschinellen Lernens zurückgegriffen. Thom et al. [BTH+13, BTW+11] stellten ScatterBlogs, eine Applikation zur Detektion von Anomalien innerhalb des sozialen Netzwerks Twitter, vor. Dieses verwendet trainierbare Klassifikatoren, um relevante Tweets, innerhalb des enormen Kommunikationsaufkommens zu erkennen und herausfiltern. Das System erlaubt zudem die Kombination verschiedener Klassifikatoren und Filter. Somit können komplexe Filter aus bereits bestehenden Filtern und trainierten Klassifikatoren erstellt und an das jeweilige Anwendungsgebiet angepasst werden. Abbildung 2.1 zeigt die ScatterBlogs-Umgebung zum Erstellen und Trainieren von Klassifikatoren. Eine Fallstudie zu

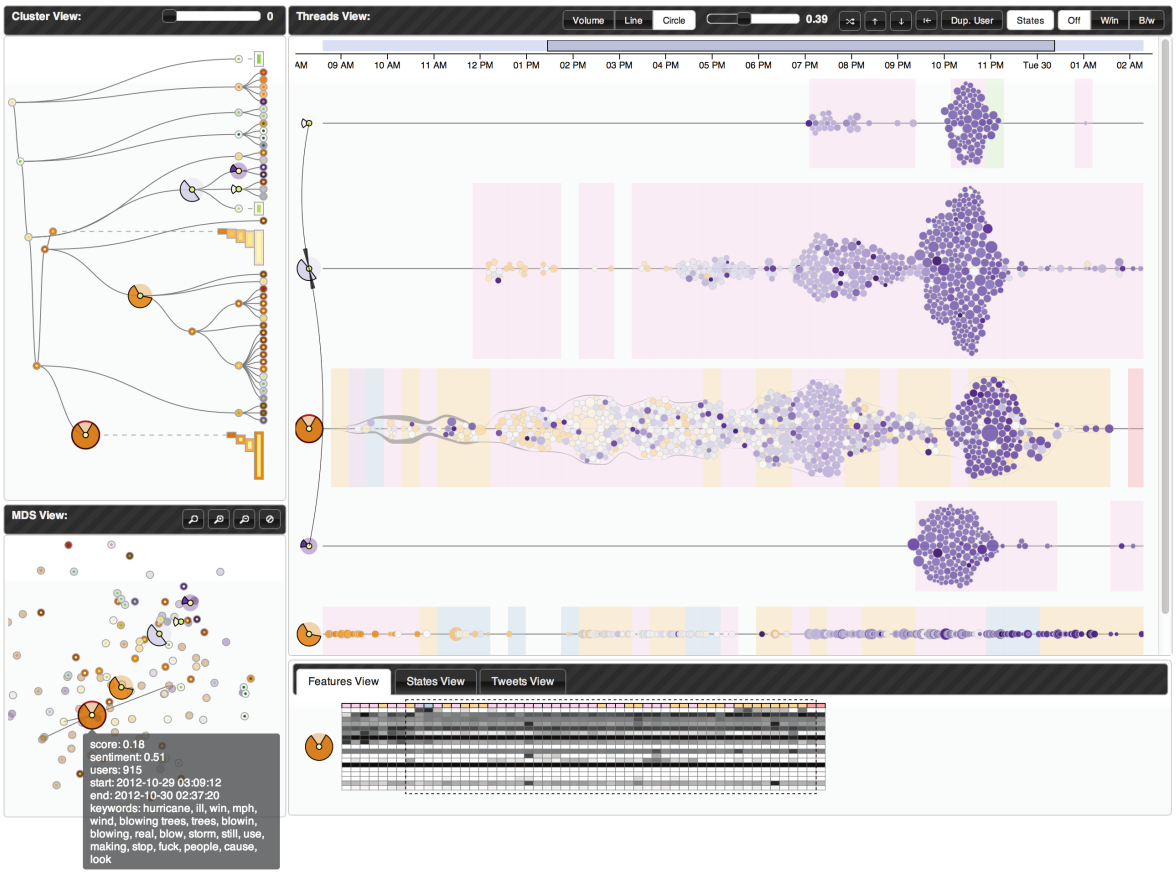


Abbildung 2.2: Benutzeroberfläche von FluxFlow. Bild von Zhao et al. [ZCW+14]

ScatterBlogs, in der Twitterdaten des Hurrikans Irene und den Unruhen in London 2011 untersucht wurden, bestätigte, dass viele Nutzer wichtige Informationen bezüglich ihres Standpunkts teilen. Dabei konnten Informationen über vom Hurrikan verursachte Stromausfälle gewonnen werden. Bei den Londoner Unruhen konnten unter Anderem die Distrikte aktueller Brennpunkte identifiziert werden. FluxFlow, von Zhao et al. 2.2, betrachtet ebenfalls in Twitter aufkommende Anomalien, um relevante Ereignisse aus dem Rauschen des täglichen Datenaufkommen herauszufiltern. Dabei liegt der Fokus jedoch auf der Detektion anomaler Informationsverbreitung innerhalb von Kommunikationsnetzwerken. Zur sequentiellen Detektion von Anomalien und ihren zeitlichen Abhängigkeiten wurde ein spezieller Algorithmus entwickelt, der zuvor mit Hilfe eines Datensatzes trainiert wird. Als Indikator für Anomalien dient dabei sowohl das Retweet-Aufkommen, als auch die in Tweets enthaltenen Schlagworte und Stimmungen. Eine Evaluation sowie Gespräche mit Domäneexperten ergaben, dass mit Hilfe von FluxFlow unter anderem effizient die Verbreitung von Falschinformationen identifiziert werden kann.

Ein Weiterer wichtiger Aspekt ist die Darstellung der Daten, diese wird von verschiedenen Arbeiten je nach Anwendungsfeld unterschiedlich realisiert. ScatterBlogs verortet die gewonnenen Daten auf einer interaktiven Karte. Mit Hilfe der Karte und einer Zeitleiste können die Daten anschließend in Abhängigkeit von Raum und Zeit analysiert werden. Zur übersichtlichen Darstellung

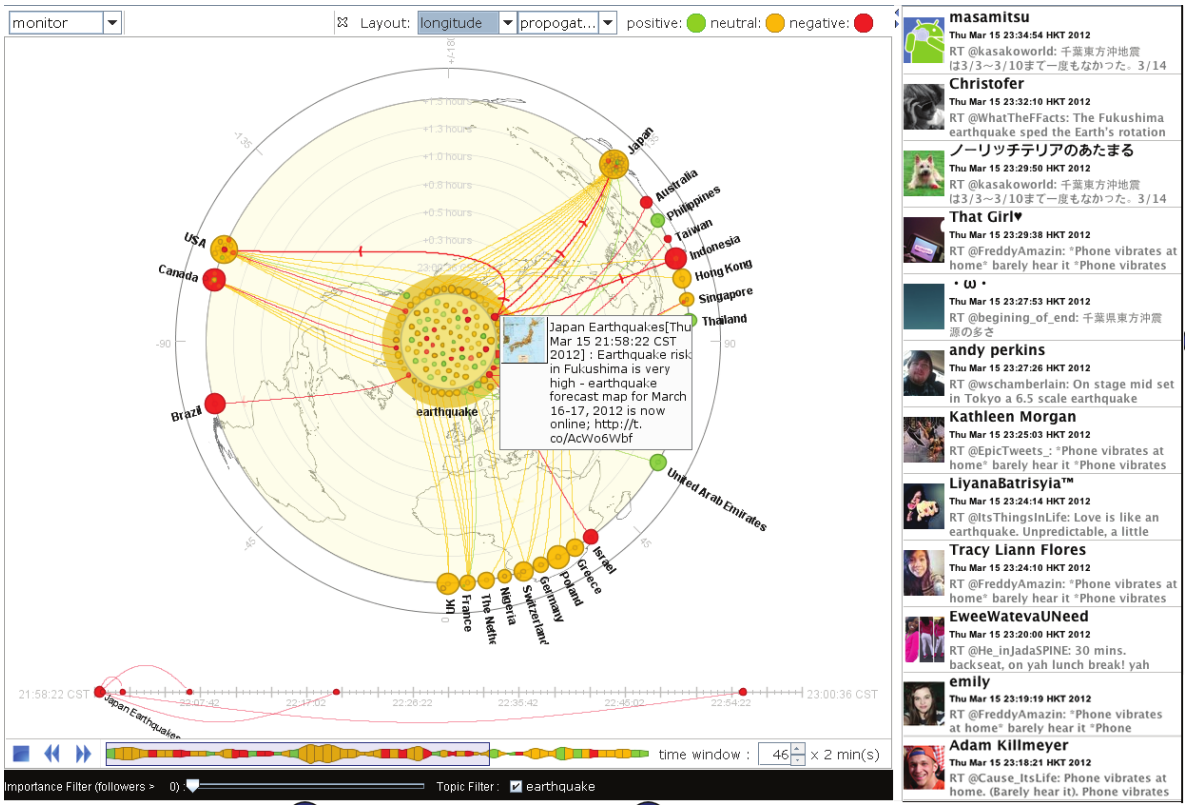


Abbildung 2.3: Whisper nützt zur Visualisierung die Metapher einer Sonnenblume. Bild von Cao et al. [CLS+12]

relevanter Themen auf der Karte wird eine Tag-Cloud verwendet. Ein visueller Filter, in Form einer Linse, wird verwendet, um Themen bestimmter Gebiete genauer aufzufächern. Da ScatterBlogs besonders auf den Einsatz in Krisensituationen, z.B. während Naturkatastrophen, abzielt, ist die Kartendarstellung von größer Wichtigkeit. Sie kann genutzt werden, um schnell einen geografischen Bezug herstellen zu können und wichtige, auf der geografischen Umgebung beruhende, Entscheidungen zu treffen. In anderen Anwendungsbereichen werden jedoch häufig abstraktere Darstellungen gewählt. Cao et al. [CLS+12] stellten mit Whisper eine weitere Anwendung zur Analyse und Darstellung der Informationsverbreitung auf Twitter vor. Diese nutzt zur Visualisierung die Metapher einer Sonnenblume, deren Samen über weite Strecken transportiert werden, um dort neue Wurzeln zu schlagen. Dies gleicht Informationen auf Twitter, die von anderen Nutzern aufgeschnappt und weiterverbreitet werden. Abbildung 2.3 zeigt ein Bild der Anwendung. Innerhalb der gelben Kreisfläche in der Mitte befinden sich die aktuell beobachteten Tweets, dargestellt als farbige Punkte analog zu den Kernen einer Sonnenblume. Wird ein solcher Tweet von einer Gruppe *geretweetet* wandert er nach außen in den jeweiligen Kreis dieser Gruppe. Dabei werden Verbindungen zu den jeweiligen Gruppen gezeichnet, welche die Blütenblätter der Pflanzen darstellen sollen. Die Farbe der einzelnen Tweets entspricht deren Stimmung. Bereits erwähntes FluxFlow, Abbildung 2.2, teilt die Visualisierung des Kommunikationsnetzwerks in mehrere Fenster auf. Während im oberen linken Fenster ein Übersichtsgraph im Form eines Knoten-Kanten-Diagramms angezeigt wird, können im Hauptfenster die zeitlich angeordneten Tweets, bzw. deren Absender, dargestellt als kleine Kreise, betrachtet werden. Größe und Farbe des Kreises sind Indikatoren für die Anzahl der Follower bzw. der Stärke der Anomaly. Durch die Aufteilung in mehrere Visualisierungen kann die topologische Übersicht zusammen mit einer Detailansicht eines ausgewählten Zeitintervalls, parallel zur Analyse genutzt werde. OpinionFlow, von Wu et al. [WLY+14], ist ein weiteres visuelles Analysesystem, um die Verbreitung von Meinungsbildern innerhalb des sozialen Netzwerks Twitter darzustellen. Zur Visualisierung der Meinungsdiffusion wurde ein sogenanntes Sankey- bzw. Flussdiagramm in Kombination mit einer speziell entwickelten Heatmap verwendet, welche den Meinungsfluss zwischen verschiedenen Nutzern darstellt. Zur Darstellung der Heatmap werden gerichtete Splats verwendet, die auf einem Gaußkernel basieren. Abbildung 2.4 zeigt den durch Splats erzeugten Meinungsfluss.

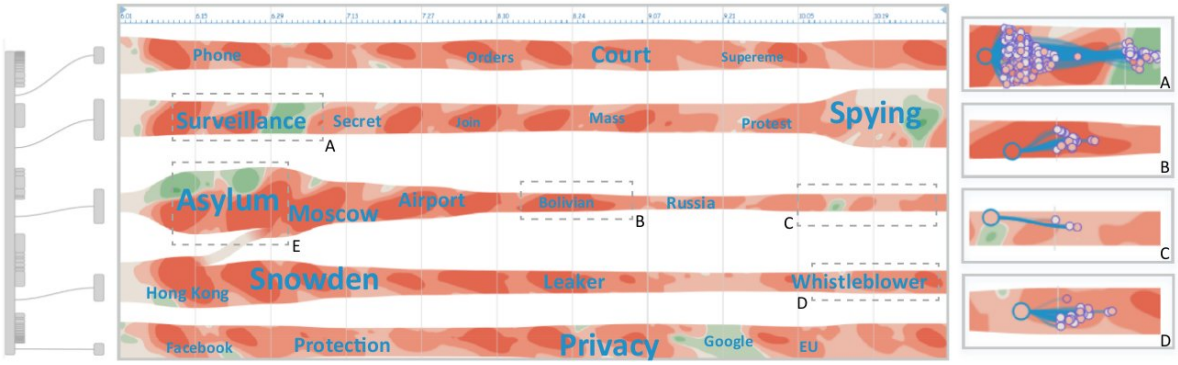


Abbildung 2.4: OpinionFlow verwendet gerichtete Splats zur Visualisierung der Meinungsdiffusion. Die Analyse der Meinungsbilder zu Thema PRISM zeigt fünf große Themenstränge. Bild von Wu et al. [WLY+14]

Existierende Arbeiten haben bereits erheblichen Aufwand zur Detektion von relevanten Nachrichten, bzw. Anomalien, betrieben. Diese Arbeit geht daher davon aus, dass relevante Nachrichten bereits vorliegen und verfolgt deren weitere Ausbreitung in Raum und Zeit. Dabei wird auf verschiedene bereits erwähnte Techniken zurückgegriffen. So wird aufgrund des Aufgabenfeldes ebenfalls eine Karte zur Darstellung der Nachrichten verwendet. Auf dieser werden die Nachrichten verortet und die Kommunikationsnetzwerke als Knoten-Kantendiagramme dargestellt. Zur zeitlichen Einordnung der Nachrichten wird auf eine Zeitleiste zurückgegriffen. Die auf der Karte verorteten Nachrichten stellen ein spatiales Histogramm dar. Fallen viele Nachrichten bzw. Punkte übereinander, können in einem solchen Histogramm nur schwer Aussagen über die Anzahl dieser Punkte getroffen werden. Durch die Darstellung in einem Knoten-Kanten-Diagramm und die daraus resultierenden Kanten kann die Lesbarkeit zusätzlich negativ beeinflusst werden. Maciejewski et al. [MRH+10] verwendeten Heatmaps zur Visualisierung spatialer Histogramme auf einer Karte. Dabei ließen sich sogenannte Hotspots, Bereiche mit besonders vielen Datenpunkten, deutlich besser erkennen als in den ursprünglichen Histogrammen. In dieser Arbeit wurden neben dem Knoten-Kanten-Diagramm gerichtete Splats ähnlich zu denen in OpinionFlow genutzt. Diese werden jedoch zur Visualisierung der geografischen Verbreitungsrichtungen auf einer Karte verwendet. Durch die Aggregation dieser Splats entsteht eine Heatmap, welche die Verbreitungsrichtungen der Nachrichten visualisiert. Diese kann durch interaktive Veränderung der Farbskalierung auch zur Detektion schwächerer Verbreitungsrichtungen genutzt werden.

3 Grundlagen

Dieses Kapitel dient der Einführung grundlegender Themen, die zum Verständnis der Arbeit relevant sind. Weiter wird angestrebt dem Leser vorab einen einheitlichen Stand zu vermitteln. Zunächst wird eine kurze Einführung in Visual Analytics und die in diesem Gebiet verwendeten Techniken gegeben. Anschließend werden Graphen und Bäume zur Repräsentation von Netzwerken und der Quadtree als Indexstruktur vorgestellt. Zuletzt wird das soziale Netzwerk Twitter und die Twitter-APIs eingeführt und deren wichtigste Funktionen erläutert.

3.1 Visual Analytics

Die enormen und stetig wachsenden Mengen an Daten, die heute tagtäglich produziert werden, bergen großes Potential. Sie enthalten wertvolle Informationen, die tiefgreifende Einsichten in verschiedenste Bereiche gewähren können. Diese Informationen zu extrahieren, stellt in Anbetracht der Ausmaße und Komplexität der Daten, jedoch eine wesentliche Hürde dar. Vor diesem Hintergrund ist in den letzten Jahren ein junges interdisziplinäres Forschungsgebiet entstanden, Visual Analytics. Eines der ersten Bücher zu diesem Thema, Illuminating the Path: The Research and Development Agenda for Visual Analytics von Thomas und Cook. [TC05], beschreibt Visual Analytics als die Wissenschaft des analytischen Urteilens, unterstützt durch interaktive, visuelle Schnittstellen zwischen Mensch und Computer.

Dabei kombiniert Visual Analytics Konzepte der automatischen Datengewinnung und Analyse, z.B. aus den Bereichen des Data Minings und maschinellen Lernens, mit Methoden der Visualisierung und Mensch-Computer-Interaktion, um Einsicht in große und komplexe Datenmengen zu erlangen. Gegenüber traditionellen Ansätzen zur Datenanalyse werden vor allem die Fähigkeiten des Menschen, Muster und Anomalien schnell erkennen zu können, adressiert. Eine zentrale Aufgabestellung ist daher, die vorliegenden Daten in eine geeignete visuelle Repräsentation zu überführen, um diese leicht zugänglich und verständlich zu machen. Zur weiteren Exploration werden interaktive Techniken zur Manipulation der Daten und deren Repräsentation verwendet, um die visuelle Analyse weiter zu unterstützen.

Keim et al. [KKEM10] haben in ihrem Buch, Mastering the Information Age - Solving Problems with Visual Analytics, einen iterativen Prozess festgehalten, in dem aus heterogenen Daten durch Anwendung automatischer und visueller Methoden Einsichten erlangt werden. Dieser ist in Abbildung 3.1 zu sehen. Ziel ist es, Einsichten bzw. Wissen aus den zugrundeliegenden Daten zu extrahieren. Hierzu werden die Daten zunächst in eine geeignete Form überführt. Anschließend können visuelle oder automatisierte Methoden zur Datenanalyse angewandt werden, die sich von Benutzer festlegen

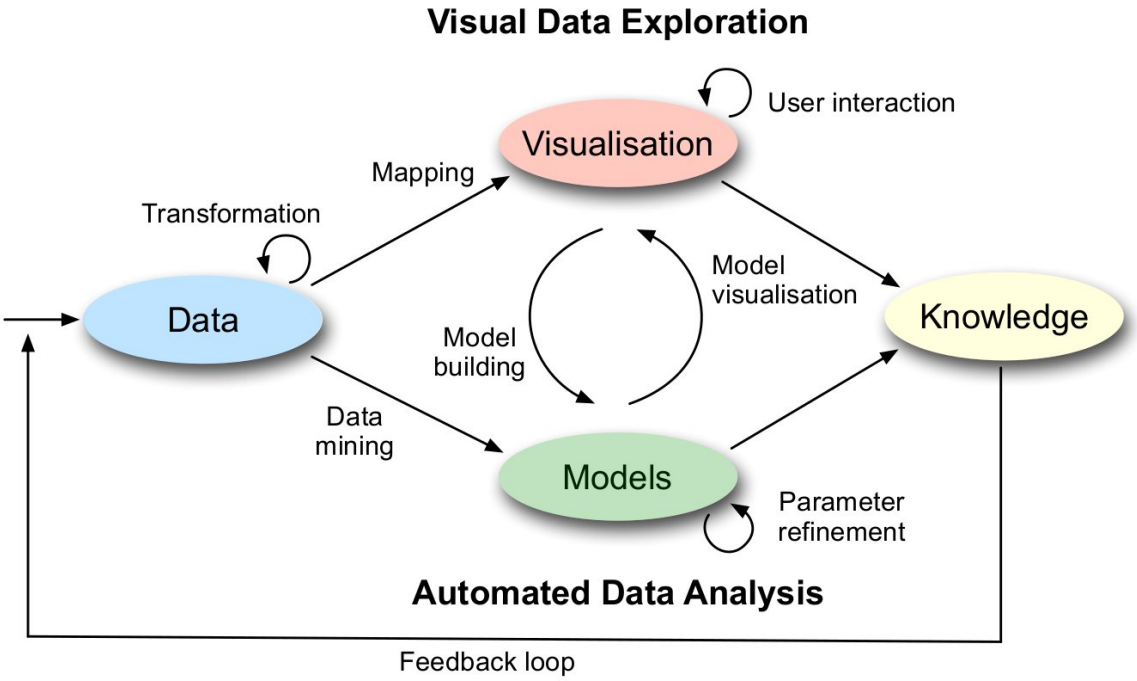


Abbildung 3.1: Visual Analytics Prozess. Bild von Keim et. al [KKEM10]

und beeinflussen lassen. Einsichten können entweder direkt auf Grund der Visualisierung und automatischen Datenanalyse oder, in einem weiteren Schritt, durch eine automatisierte bzw. visuelle Überprüfung zuvor aufgestellter Hypothesen, gewonnen werden. Zur weiteren Analyse kann dieser Prozess beliebig oft iteriert werde.

Gerade bei der Exploration großer und komplexer Informationsräume können Visual Analytics Konzepte vielversprechende Resultate liefern. So sind potenzielle Einsatzbereiche die Physik, Astronomie, Wirtschaft und Marketing, sowie Krisenmanagement und Verbrechensbekämpfung. Viele zu analysierende Daten, z.B. Daten aus sozialen Netzwerken, enthalten dabei Netzwerkstrukturen, weshalb die Netzwerkanalyse und Visualisierung ein wichtiges Teilgebiet der Informationsvisualisierung und Visual Analytics darstellt.

3.2 Repräsentation von Netzwerken

Einzelne Objekte einer Menge, die untereinander durch Relationen zusammenhängen, bilden ein Netzwerk. Sind diese Objekte beispielsweise Computer und die Relation ist das Bestehen einer physikalischen Verbindung zum Zweck der Datenübertragung, spricht man von einem Computernetzwerk. Im Zusammenhang mit sozialen Medien sind die Objekte Personen, die soziale Verbindungen miteinander eingehen, und somit ein soziales Netzwerk bilden. Verschiedene Netzwerke haben unterschiedliche Eigenschaften und lassen sich allgemein als Graphen darstellen. Mit Graphen und deren Eigenschaften und Repräsentationsmöglichkeiten beschäftigt sich das Feld der Graphentheorie.

3.2.1 Graphen

Ein Graph besteht aus einer Menge von Elementen und einer Menge von Verbindungen zwischen diesen Elementen. Die Elemente eines Graphen werden dabei als Knoten, die Verbindungen zwischen den Knoten als Kanten bezeichnet. Ein allgemeiner Graph stellt keine Anforderungen an die Endpunkte einer Kante, diese können sich auch auf den selben Knoten beziehen. Abbildung 3.2 zeigt einen gerichteten und einen ungerichteten Graph.

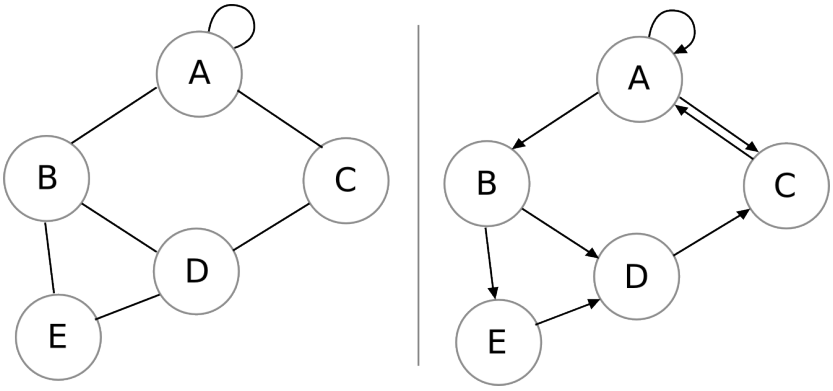


Abbildung 3.2: Links: Ungerichteter Graph; Rechts: Gerichteter Graph

Definition 3.2.1
Ein Graph ist ein Tupel $G = (V, E)$, wobei V die Menge aller Knoten und E die Menge aller Kanten darstellt. Die Kantenmenge $E \subseteq \{\{u, v\} : u, v \in V\}$ ist eine Teilmenge aller 2-elementigen Teilmengen von V.

Ein Graph kann somit beispielsweise zur Darstellung eines Rechnernetzes, wie oben beschrieben, genutzt werden, da die Relation der Rechner zueinander in beide Richtungen gilt, also symmetrisch ist. Hat ein Computer A eine physikalische Verbindung zu einem anderen Computer B, so hat B auch eine Verbindung zu A. Um auch einseitige, nicht symmetrische Relationen aus Netzwerken darstellen zu können, bedarf es eines gerichteten Graphen. Dazu muss lediglich die Menge der Kanten E neu definiert werden.

Definition 3.2.2
Die Kantenmenge $E = ((u, v) : u, v \in V)$ eines gerichteten Graphen $G = (V, E)$ ist eine Teilmenge aller Tupel (u, v) mit Elementen aus V. Dabei wird u als Start- und v als Zielknoten der Kante bezeichnet.

Durch diese Definition der Kantenmenge sind nun auch gerichtete Kanten von einem Startknoten zu einem Zielknoten möglich. So können z.B. bestimmte Relationen in sozialen Netzwerken modelliert werden. Abonniert ein Benutzer A die Nachrichten eines Benutzers B, kann dies durch eine gerichtete Kante zwischen den zugehörigen Knoten modelliert werden. Diese Relation ist nicht symmetrisch, sodass der Benutzer B nicht zwangsläufig die Nachrichten von Benutzer A abonniert haben muss. Der eingehende bzw. ausgehende Grad eines Knotens ist dabei die Anzahl der eingehenden bzw.

ausgehenden Kanten. Wird ein Graph grafisch, wie in Abbildung 3.3 dargestellt, so spricht man auch von einem Knoten-Kanten-Diagramm.

3.2.2 Bäume

Bäume werden oft benutzt um hierarchische Netzwerke zu modellieren. Ein Baum ist ein zusammenhängender Graph, der bei ungerichteter Betrachtung der Kanten keine Zyklen aufweist [Wika]. Ein gewurzelter Baum weist dabei genau einen Knoten als Wurzelknoten aus. Bei gerichteten Bäumen ist die Wurzel ein Knoten, der keine eingehenden Kanten besitzt. Im Folgenden dieser Arbeit ist mit dem Begriff Baum ein gewurzelter, gerichteter Baum im Sinne der obigen Definition gemeint.

Des Weiteren wird ein Knoten eines Baumes A als Kind eines anderen Knoten B bezeichnet, wenn eine Kante von B nach A existiert. Umgekehrt wird ein Knoten A als Elternknoten eines Knoten B bezeichnet, wenn eine Kante von A nach B existiert. Ein Knoten der keine ausgehenden Kanten aufweist wird als Blatt bezeichnet.

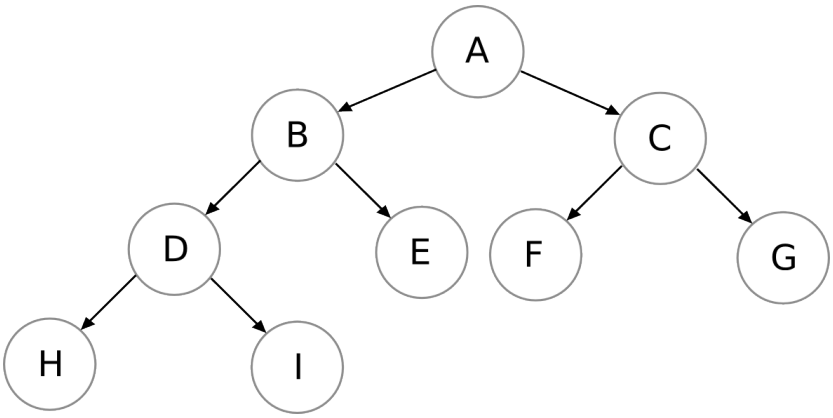


Abbildung 3.3: Baum mit gerichteten Kanten

Quadtree

Gerade im Visual Analytics Bereich wird meist mit sehr großen, komplexen Datenmengen gearbeitet. Wichtig ist daher, im Zuge einer schnellen und interaktiven Analyse, auch die Wahl geeigneter Indexstrukturen. Hierzu können bestimmte Bäume, sogenannte Quadtrees, verwendet werden. Ein Quadtree stellt eine spezielle Baumstruktur dar, bei welcher jeder Elternknoten genau 4 Kinder besitzt. Sie werden oft benutzt um zweidimensionale Rechteckflächen in feinere Rechteckstrukturen aufzuteilen. Dem Wurzelknoten wird dabei eine initiale Fläche zugewiesen. Anschließend wird diese rekursiv in vier gleichgroße, untergeordnete Rechteckflächen, eine je Kind, unterteilt. Dies geschieht so lange, bis die gewünschte Granularität bzw. Größe der Blattknoten erreicht ist. Abbildung 3.4 stellt die Aufteilung einer Fläche innerhalb des Quadtrees dar.

Ein weiteres häufiges Einsatzgebiet vom Quadtrees stellt die Computergrafik dar. Dort werden

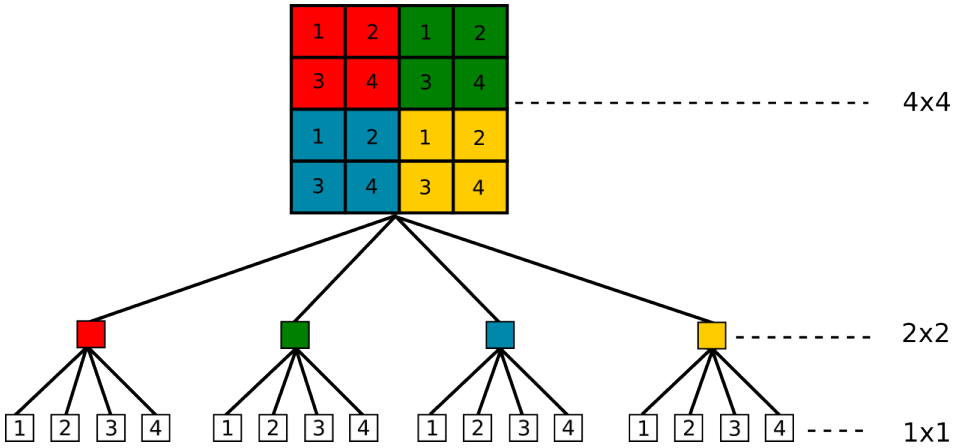


Abbildung 3.4: Die Fläche wird auf die Blattknoten des Quadtrees aufgeteilt. Jeder Knoten entspricht einem bestimmten Teil der Gesamtfläche.

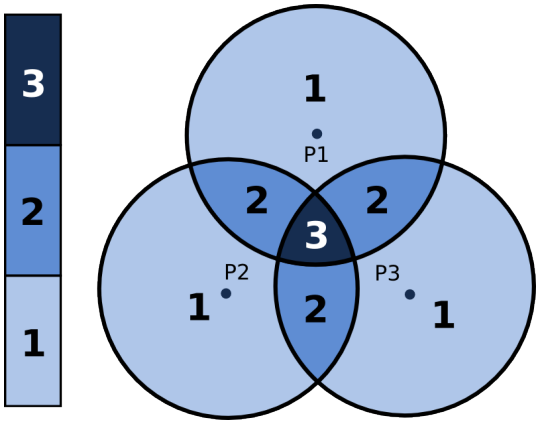
sie zur Organisation zweidimensionaler Daten verwendet. Da die Rechteckflächen der Blätter die komplette Fläche des Quadtrees abdecken, können Punkte darauf eindeutig einem Blattknoten des Baumes zugeordnet werden. Die Datenstruktur kann nun effizient feststellen, welche Punkte sich in bestimmten Bereichen des Quadtrees befinden, indem die Blattknoten, die diesen Ausschnitt abdecken, betrachtet werden. Dies ist z.B. hilfreich, um Punkte abzufragen, welche sich innerhalb des aktuellen Viewports einer grafischen Anwendung befinden, und somit gezeichnet werden müssen.

3.3 Heatmaps

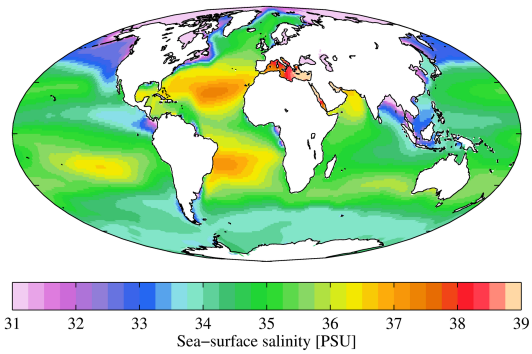
Heatmap können auf vielfältige Weise zur Visualisierung von Datensätzen auf einer zweidimensionalen Ebene verwendet werden. Eine wichtige Rolle nehmen diese vor allem bei der Visualisierung sogenannter Dichtekarten ein. Diese geben Auskunft über die Verteilung der zugrundeliegenden Daten, an bestimmten Stellen der Karte. Dabei gibt es verschiedene Möglichkeiten zur Erzeugung solcher Dichtekarten.

Bei einer Kerndichteschätzung werden die Daten des zugrundeliegenden Datensatzes als statistische Stichprobe betrachtet. Ausgehend von diesen lokal entnommenen Stichproben wird versucht auf die globale Verteilung der Daten zu schließen. Jedem Punkt auf der Karte kann so eine bestimmter Wert zugewiesen werden.

Ein weiterer Ansatz zur Generierung von Heatmaps stellt das sogenannte Splatting dar. Dem Bereich um einem Datenpunkt wird dabei ein gewisser, meist skalarer Wert, zugewiesen. Der mit einem Wert versehene Bereich eines einzelnen Datenpunktes wird als Splat bezeichnet. In überlappenden Bereichen werden die einzelnen Werte miteinander verrechnet. Bereiche auf der Karte, an denen sich viele Datenpunkte ballen, überlagern sich dementsprechend oft und sind daher meist stärker ausgeprägt. Abbildung 3.5a zeigt drei Datenpunkte P_1, P_2 und P_3, deren Splats sich gegenseitig überlappen,



(a) Überlagerung einzelner Splats



(b) HeatMap zum Darstellung des prozentualen Salzgehalts der Meere. Bild von Wikipedia [Wikc]

Abbildung 3.5: Darstellung Heatmap

dabei addieren sich die Werte der Splats auf. Splats liegen dabei oft bestimmte Kernelfunktionen, z.B. Gaußkernel, zu Grunde.

3.3.1 Farbcodierung

Heatmaps eignen sich besonders gut, um markante Stellen und Trends innerhalb großer Datenmengen zu detektieren, wenn die einzelnen Datenpunkte sich sinnvoll auf einer Ebene verorten lassen. Zunächst muss jedoch eine geeignete Farbcodierung gefunden werden.

Die Werte der Dichtekarte werden hierzu auf eine Farbskala übertragen, siehe Abbildung 3.5a. Die daraus entstehende Visualisierung wird als Heatmap bezeichnet. Ein Beispiel einer solchen Heatmap ist in Abbildung 3.5b zu sehen. Die Wahl der Farbskala bzw. der Farbcodierung kann dabei von entscheidender Bedeutung bei der Detektion von Mustern innerhalb einer Datenmenge sein. Zur Farbcodierung werden oftmals die Farben einer Temperaturskala verwendet, welche namensgebend für die Heatmap ist.

3.4 Twitter

Zur automatisierten Verarbeitung und Analyse von Nachrichten in sozialen Netzwerken, eigenen sich vor allem bereits erwähnte Microblogging-Dienste, deren Nachrichten meist einer gewissen Struktur folgen und von kurzer Länge sind. Diese Arbeit befasst sich exemplarisch mit Twitter, einem der weltweit größten Microblogging-Dienste.

Twitter ist ein soziales Netzwerk, genuer Microblogging-Dienst, der es Nutzern ermöglicht, Kurznachrichten, von maximal 140 Zeichen Länge, sogenannte Tweets, abzusetzen. Alle öffentlichen Tweets eines Nutzers sind anschließend auf dessen Profil einsehbar. Unter anderem haben Nutzer von

Twitter die Möglichkeit anderen Nutzern zu folgen. Das Folgen einer Person kann als Abonnement ihrer Tweets verstanden werden. Die folgenden Personen werden als Follower bezeichnet. Abonierte Tweets werden anschließend automatisch auf der Startseite der Follower, ihrer sogenannten Timeline, angezeigt. Abgesetzte Tweets können somit, je nach Anzahl der Follower eines Nutzers, in kurzer Zeit eine Vielzahl anderer Personen erreichen.

Die am häufigsten genutzten Möglichkeiten der Nutzerinteraktion auf Twitter sind sogenannte Retweets und Replies. Während das *Retweeten* einer Nachricht diese an die eigenen Follower weiterleitet, kann ein Reply als eine direkte Antwort auf eine Nachricht verstanden werden. Während Privatpersonen meist aktuelle Tätigkeiten, Beobachtungen und Meinungen teilen, wird Twitter von Medien und öffentlichen Stellen zur schnellen Verbreitung aktueller Nachrichten verwendet. Mit ca. 500 Millionen abgesetzten Tweets pro Tag ist Twitter eines der meistgenutzten sozialen Netzwerke weltweit [Twib].

3.4.1 Retweets

Abbildung 3.6: Links: Retweeten eines Tweets über den Retweet-Button; Rechts: Manueller Retweet mit eigenem Kommentar

Als Retweet wird ein Tweet eines anderen Benutzers bezeichnet, der mit den eigenen Followern geteilt wird. Somit dienen Retweets der Weiterverbreitung von Nachrichten Anderer. Dabei gibt es zwei Arten von Retweets. Zum einen die native Variante, die direkt vom User-Interface in Form eines Reetweet-Buttons bereitgestellt wird, zum anderen die Möglichkeit Retweets in Form eines normalen Tweets zu schreiben, der das Schlüsselwort $RT@ < Username >$ beinhaltet. Beide Varianten einen Retweet abzusetzten sind in Abbildung 3.6 zu sehen.

Native Retweets

Native Retweets sind die von Twitter implementierte Ein-Klick-Variante, um Tweets zu retweeten. Sie wurden erst nachträglich von Twitter eingeführt, um eine schnelle Möglichkeit des Retweetens anzubieten. Native Retweets enthalten keine zusätzlichen Kommentare und können nicht direkt über die Suchfunktion, als eigenständige Tweets, gefunden werden.

Erneute Retweets eines nativen Retweets verhalten sich wie ein Retweet des ursprünglichen Tweets. Solche Retweets verweisen also direkt auf den ursprünglich abgesetzten Tweet. Native Retweets können im Gegensatz zu manuellen Retweets direkt über die Twitter-API abgefragt werden.

Manuelle Retweets

Manuelle Retweets werden auch Classic-Retweets genannt, da sie die ursprüngliche Form des Retweet, vor Einführung der nativen Retweets, auf Twitter darstellten. Um einen manuellen Retweet zu erstellen, wird der Inhalt des ursprünglichen Tweets kopiert und in einen neuen Tweet eingefügt. Anschließend wird der Prefix $RT@ < Username >$ angefügt, um den Urheber zu referenzieren. Möchte man zusätzlich einen eigenen Kommentar abgeben, fügt man diesen, meist vor der Nutzer-Referenz, an. Ein manueller Retweet hat also die Form: $< Kommentar > RT@ < Username >< UrspruenglicherTweet >$. Es gilt weiterhin die Beschränkung auf 140 Zeichen, daher muss der ursprünglicher Tweet möglicherweise gekürzt werden.

Manuelle Retweets haben gegenüber nativen Retweets den Vorteil über die Suchfunktion gefunden zu werden, da sie im Grunde eigenständige Tweets sind. Aus dem selben Grund beziehen sich Retweets eines manuellen Retweets direkt auf diesen und nicht auf den ursprünglichen Tweet [Dif].

3.4.2 Replies

Während ein Retweet als Weiterleitung einer Nachricht an die eigenen Follower verstanden werden kann, stellt ein Reply eine direkte Antwort auf einen Tweet dar. Auch Replies können dabei auf zwei unterschiedliche Arten abgesetzt werden.

Direkte Nachrichten

Möchte man eine private Nachricht übermitteln, die nur für den Empfänger sichtbar ist, so kann man dies in Form einer direkten Nachricht, kurz DM, tun. DMs lassen sich direkt, über einen extra dafür vorgesehenen Button, absenden. Allerdings können direkte Nachrichten ausschließlich an die eigenen Follower versendet werden.

Manuelle Replies

Die zweite Möglichkeit stellt das manuelle versenden von Replies, ähnlich dem Versenden manueller Retweets, dar. Dem Tweet wird hierzu die Sequenz $< @username >$ beigefügt. Dabei wird unterschieden ob $< @username >$ den Präfix bildet oder inmitten des Tweets auftaucht.

Im ersten Fall ist der Tweet direkt an den Empfänger adressiert und somit ausschließlich für Benutzer sichtbar, die sowohl Absender als auch Empfänger folgen. Im letzteren Fall stellt der Tweet eine sogenannte Erwähnung dar und ist auch für Nutzer sichtbar, die dem Empfänger nicht folgen.

Nach dem Login kann der Empfänger alle, an ihn adressierten, manuellen Replies unter seinen Replies/Erwähnungen einsehen [Dif, Twid].

3.4.3 Geolokation von Tweets

Twitter ermöglicht es Nutzern ihren Tweets zusätzliche Informationen zu deren Standort anzuhängen. So wird einem Tweet aus Iowa beispielsweise der Standort „Iowa, USA“ angehängt. Nutzen die Absender moderne Mobilgeräte, so kann die Angabe noch genauer ausfallen. Mittels GPS-Sensoren kann der Aufenthaltsort des Absenders, teils bis auf wenige Meter genau, geortet und unter Angabe von Breiten- und Längengrad angezeigt werden.

Die Entscheidung ob und bei welchen Tweets Informationen zum Standort angehängt werden sollen, bleibt dabei dem Nutzer überlassen. Dieser kann die Standortangabe jeder Zeit deaktivieren und alle bisher verwendeten Standorte einsehen. Standortangaben können darüber hinaus auch nachträglich von abgesetzten Tweets entfernt werden [Twia].

3.5 Twitter-APIs

Entwickler haben die Möglichkeit, mittels von Twitter bereitgestellten APIs auf Twitterdaten zuzugreifen und diese in Applikationen einzubinden. Zu diesem Zweck werden verschiedene APIs angeboten, darunter die Twitter Rest- und Streaming-API. Während die Streaming-API einen Livestream aktueller Tweet bereitstellt, ist die Rest-API auf gezielte Anfragen und die Suche spezieller Tweets ausgelegt. Um die APIs nutzen zu können, müssen sich die Entwickler bzw. die Benutzer der Applikation zunächst gegenüber Twitter authentifizieren. Die im folgenden verwendeten Informationen sind im Detail in der Twitter-API Dokumentation [Twic] nachzulesen.

3.5.1 Authentifizierung

Die Authentifizierung gegenüber Twitter erfolgt mit Hilfe des OAuth-Protokolls. Dabei ist es möglich, sich gegenüber der API als Nutzer oder als Applikation zu authentifizieren. Letzteres wird als Application-only Authentication bezeichnet. Bei einer Application-only Authentication kann nicht auf nutzerspezifische Funktionen, wie das Versenden von Tweets, zurückgegriffen werden. Andererseits sieht die Application-only Authentication für andere Funktionen weniger strenge Abfragelimitierungen vor.

3.5.2 Rest-API

Die Twitter Rest-API bietet Entwicklern die Möglichkeit gezielt Tweets abzufragen und abzusetzen. Für jede Anfrage wird dabei eine neue Verbindung zu Twitter aufgebaut. Jede gesendete Anfrage wird einzeln bearbeitet und anschließend eine Antwort zurück gesendet. Abbildung 3.7 zeigt die Abfrage von Daten der Rest-API durch eine Webapplikation.

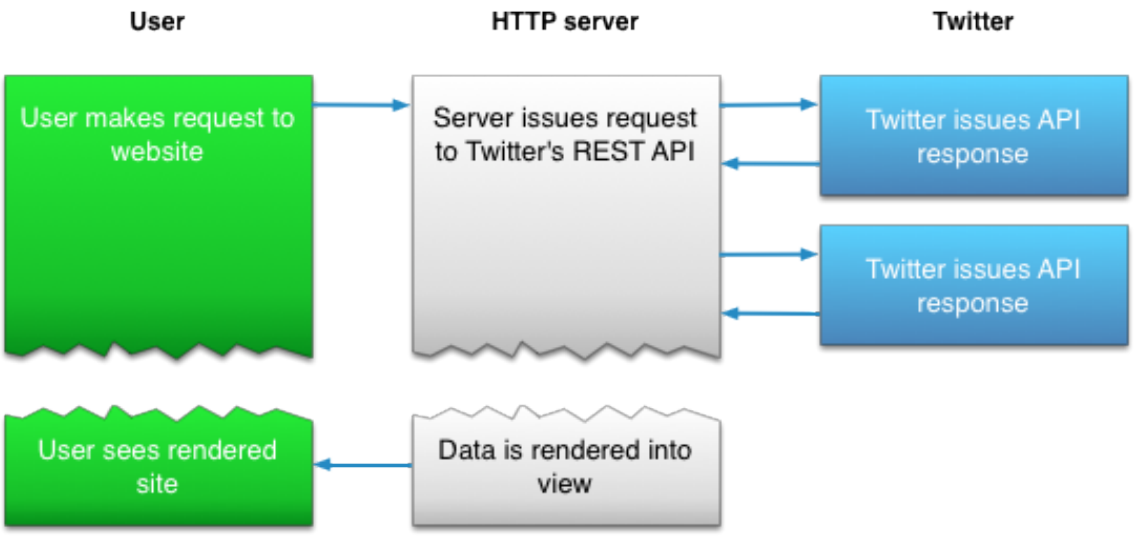


Abbildung 3.7: Durch einen Benutzer der Webapplikation ausgelöste Anfragen werden von Twitter bearbeitet und beantwortet. Die erhaltenen Daten werden anschließend dem Benutzer angezeigt. Dabei wird für jede Anfrage der Rest-API eine neue Verbindung aufgebaut. Bild aus Twitter-API Dokumentation [Twic]

Die Such- und Abfragefunktionen der Rest-API ermöglichen es, sehr genaue Anfragen zu formulieren. Zu den Suchoptionen gehört unter anderem die Suche nach bestimmten Inhalten, Absendern und Empfängern, Stimmungen oder Hashtags. So kann beispielsweise eine Anfrage formuliert werden, welche Tweets zurückgibt, die das Wort „Restaurant“, nicht jedoch das Wort „Italiener“ beinhalten und eine positive Stimmung widerspiegeln. Eine entsprechende Anfrage würde wie folgt aussehen: $Restaurant - Italiener :)$. Dabei ist zu beachten, dass die Suchfunktion der Rest-API keinen Anspruch auf Vollständigkeit erhebt und sich vielmehr auf das Durchsuchen aktuell relevanter Tweets bezieht. Des Weiteren können mit der Rest-API auch Tweets abgesetzt und Informationen zu Nutzer, Beziehungen zwischen Nutzern oder Einstellungen abgerufen und geändert werden, sofern man als Benutzer authentifiziert ist.

3.5.3 Streaming-API

Anstatt einzelne Anfragen an Twitter zu senden, baut die Streaming-API eine persistente HTTP-Verbindung zu den Twitter-Servern auf. Einmal verbunden wird so ein stetiger Fluss an aktuellen Tweets empfangen. Die Streaming-API ist damit gut für Echtzeitanwendungen und dem Sammeln von Twitterdaten, beispielsweise zu statistischen Zwecken, geeignet. Abbildung 3.8 zeigt eine Webapplikation, welche mit der Streaming-API arbeitet. Anfragen der Nutzer werden mit den bereits gesammelten Daten bearbeitet.

Die Streaming-API unterstützt das Empfangen von verschiedenen Streams. Public Streams beinhalten einen Auszug aller öffentlichen Tweets, die auf Twitter geteilt werden, während User Streams alle für einen bestimmten Benutzer relevanten Tweets beinhalten. Außerdem gibt es sogenannte Side Streams, welche die relevanten Tweets einer Vielzahl von ausgewählter Benutzer zurückliefern können. Die Streaming-API bietet darüber hinaus auch das Vorfiltern von Streams an. So ist es möglich einen Stream zu empfangen, dessen Tweets ausschließlich bestimmte Schlagworte enthalten oder von bestimmten Verfassern stammen.

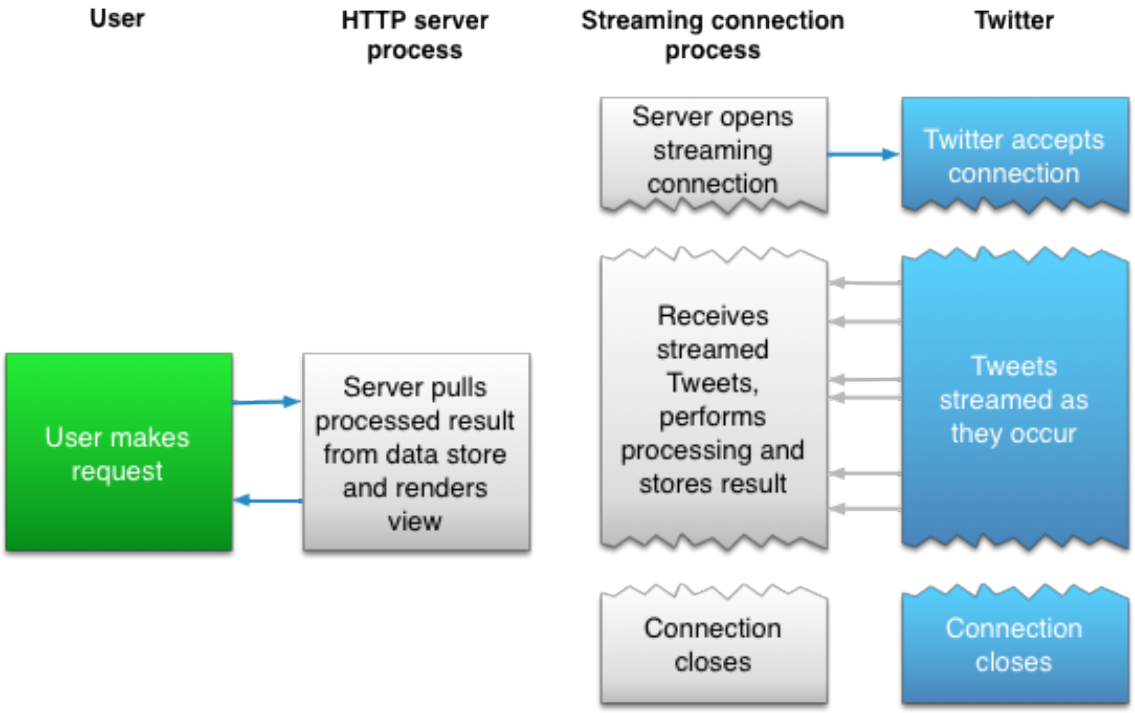


Abbildung 3.8: Die Verbindung zur Streaming-API wird zu Beginn der Anwendung aufgebaut. Anschließend wird ein Livestream aktueller Tweets empfangen. Diese werden lokal auf dem Server gehalten. Anfragen des Nutzers werden mit Hilfe der gesammelten, lokal gespeicherten Daten beantwortet. Bild aus Twitter-API Dokumentation [Twic]

3.5.4 Aufbau eines Tweets

Die APIs arbeiten mit vier grundlegenden Objekten: Tweets, Users, Entities und Places. API intern werden Tweets auch als Status-Updates oder Statuses bezeichnet, da sie den aktuellen Status eines Users wiedergeben. Als Rückgabeformat wird JSON verwendet. Tweets, Users und Places sind weitgehend selbsterklärend, sie enthalten die grundlegenden Informationen eines Tweets, Benutzers bzw. Ortes. Entity-Objekte beinhalten hingegen zusätzliche Metainformationen zu Inhalten, die auf Twitter veröffentlicht wurden. Dabei treten Entities meist innerhalb anderer Objekte auf. So enthalten die Entities eines Tweets beispielsweise Informationen zu beinhalteten Medien, sowie eine Liste der darin

vorkommenden Hashtags und URLs. Das Entity-Objekt eines nativen Retweets enthält zudem den referenzierten, ursprünglichen Tweet.

```
1  {
2    "coordinates": null,
3    "favorited": false,
4    "truncated": false,
5    "created_at": "Wed Jun 0620:07:10+0000 2012",
6    "id_str": "210462857140252672",
7    "entities": {
8      ...
9    },
10   "in_reply_to_user_id_str": null,
11   "contributors": [
12     14927800
13   ],
14   "text": "Along with ...",
15   "retweet_count": 66,
16   "in_reply_to_status_id_str": null,
17   "id": 210462857140252672,
18   "geo": null,
19   "retweeted": true,
20   "possibly_sensitive": false,
21   "in_reply_to_user_id": null,
22   "place": null,
23   "user": {
24     ...
25   }
26 }
```

Listing 3.1: Tweet bzw. Status in JSON

Listing 3.1 zeigt einen Tweet im JSON-Format, der die gängigsten Felder beinhaltet. In Zeile 7 und 23 ist zu erkennen, dass der Tweet Informationen über Entities und den Benutzer direkt in einem integrierten Objekt mitliefert. Je nach Anfrage können die enthaltenen Felder leicht variieren.

3.5.5 Limitierungen

Twitter limitiert die Anzahl der Anfragen, die in einem gewissen Zeitfenster, welches momentan 15 Minuten beträgt, von der Rest-API bearbeitet werden. Dabei wird unterschieden, ob man gegenüber Twitter als Nutzer oder Applikation authentifiziert ist. Weiterhin gibt es je nach Anfragetyp unterschiedliche Obergrenzen wie oft diese pro Zeitfenster abgesetzt werden dürfen. Die Suche nach Tweets ist beispielsweise auf 180 Tweets pro Nutzer bzw. 450 Tweets pro Applikation, je 15 Minuten beschränkt. Für die Streaming-API existiert eine Limitierung bezüglich der Anzahl der Neuverbindungen eines Streams. Eine Neuverbindung ist beispielsweise bei einer Änderung der Streamparameter notwendig. Genaue Angaben zur Anzahl der Neuverbindungen oder einem Zeitfenster macht Twitter jedoch nicht.

4 Konzepte

Im Folgenden werden die Konzepte und einzelnen Aspekte der Arbeit erlautert. Ausgangspunkt ist ein Workshop mit Domäneexperten, aus dem verschiedene Anforderungen Abgeleitet werden. Anschließend wird das Beziehen von Twitterdaten und deren Überführung in eine geeignete Baumstruktur für die Visualisierung erläutert. Zuletzt wird auf den Aufbau einer Heatmap mit Hilfe gerichteter Splats und mögliche Vorgehensweisen hierzu eingegangen.

4.1 Workshop mit Domäneexperten

Bei einem Workshop mit Domäneexperten des LK Viersen, im Zuge des BMBF Forschungsprojekts VASA ergab sich die Möglichkeit, mit Personen aus verschiedenen Bereichen eines Krisenstabs zu sprechen. Nachfolgend sind maßgebliche Kommentare und Erkenntnisse zum Thema aufgeführt, die in diesem Rahmen gewonnen wurden.

Großes Interesse zeigten die Teilnehmer an der Verfolgung der Verbreitungswege von herausgegebenen Warnungen und Pressemitteilungen. Werden diese schnell und oft von anderen Benutzern aufgegriffen und weiterverbreitet, lässt dies tendenziell auf eine hohe Akzeptanz schließen. Umgekehrt könnten bei geringer Verbreitung erneut Mitteilungen herausgegeben werden, um den Prozess zu beschleunigen. Neben der Verbreitungsgeschwindigkeit sei laut Experten zudem das Ausbreitungsgebiet interessant. So stimmten die Experten zu, dass Wissen über das Gebiet und die Richtung der Ausbreitung von Nachrichten bzw. Informationen, zur besseren Einschätzung der Informationslage und deren Entwicklung beitragen kann. So könnten beispielsweise Gebiete identifiziert werden, in welcher wichtige Meldungen noch nicht verbreitet sind.

Im Zuge des Workshops kam zudem ein Vorfall während einer kontrollierten Bombensprengung einer Fliegerbombe aus dem zweiten Weltkrieg in Viersen zur Sprache. Über soziale Medien wurde dabei zu einer sogenannten „Bombenparty“ aufgerufen, aufgrund deren sich Feuerwehrleute in gefährliches Gebiet begeben mussten. In einem solchem Szenario sei gemäß der Experten sowohl interessant, wie und an welche Gruppen sich solche potenziell gefährlichen Nachrichten verbreiten, als auch die Rückverfolgung der Nachrichten zu deren Urhebern.

Darüber hinaus wurde Interesse an Möglichkeiten zu Einflussnahme bzw. Kontaktaufnahme zu Personen, die an der Informationsverbreitung beteiligt sind, betont. Oftmals würden sich freiwillige Helfer in Eigeninitiative über soziale Medien organisieren. Könne man diese ausmachen, wäre eine Kommunikation bzw. Koordination mit diesen Helfern denkbar. Gleichzeitig wurden hierzu jedoch auch datenschutzrechtliche Bedenken geäußert.

Zur Darstellung von Verbreitungsnetzen in einem Knoten-Kanten-Diagramm zeigten sich einige der Teilnehmer skeptisch und verwiesen auf die Unübersichtlichkeit bei großen bzw. dichten Graphen. Auch die gleichzeitige Darstellung mehrerer Verbreitungs-Graphen könne zu Problemen führen. Karten, auf denen zusätzliche Informationen dargestellt sind, könnten die Lesbarkeit des Knoten-Kanten-Diagramms zusätzlich einschränken.

Weiterhin wurde auf die Ausbildung der Mitarbeiter des Krisenstabs hingewiesen. Da diese keine geschulten Analysten seien, müsse bei möglichen Umsetzungen vor allem auf eine leichte Zugänglichkeit geachtet werden.

Zusammengefasst lassen sich aus dem Workshop folgende, für die Teilnehmer wichtige Punkte, ableiten:

- Verfolgung der Verbreitung von eigenen Pressemitteilungen und herausgegebenen Warnungen
- Verfolgung von einzelnen Nachrichten zu deren Quelle
- Übersichtliche Darstellung der Verbreitungswege, beim Betrachten von mehreren oder dichten Graphen
- Intuitive Bedienbarkeit und niedrige Anforderungen an Analysten

4.1.1 Konzeptentwicklung ausgehend von Workshop

Um Informationen bezüglich der Nachrichten- bzw. Informationsdiffusion darstellen und analysieren zu können, müssen diese zunächst vorliegen. Ein wesentlicher Aspekt ist daher die Beziehung der zugrundeliegenden Daten von Twitter und deren Überführung in geeignete Datenstrukturen. Sind die zugrundeliegenden Daten geeignet repräsentiert, muss über deren Visualisierung, entschieden werden. Eine Orientierung stellen dabei die gewonnenen Einsichten des Workshops dar.

Ein Anliegen der Teilnehmer der Workshops war das gezielte Verfolgen von Nachrichten, zu deren Empfängern und Urhebern. Dies kann visuell nur durch eine sehr detaillierte Darstellung der einzelnen Nachrichten und deren Verbreitungswege ermöglicht werden. Hierzu eignet sich ein Knoten-Kanten-Diagramm. Es ermöglicht eine einfache und klare Darstellung einzelner Tweets und derer Relationen untereinander. Zudem haben die meisten Datenanalysten bereits mit Knoten-Kanten-Diagramm gearbeitet, sodass deren Semantik intuitiv erfasst werden kann.

Die Darstellung einer Übersicht der aktuellen Informationslage oder der Verbreitung der Nachrichten mittels eines Knoten-Kanten-Diagramm, kann jedoch zu Problemen führen. Wie bereits von den Domäneexperten angemerkt, neigen besonders große und dichte Graphen bei der Darstellung als Knoten-Kanten-Diagramm zu starker Unübersichtlichkeit und schlechter Lesbarkeit. Zur Darstellung einer Übersichtskarte muss daher eine alternative Visualisierung gefunden werden.

4.1.2 Überlegungen zur Visualisierung

Zur übersichtlichen Darstellung von Graphen existieren bereits eine Vielzahl an Techniken. Nicht alle lassen sich jedoch sinnvoll auf die gegebene Aufgabestellung anwenden.

Holten et al. [Hol06, HVW09] stellten mit Egde Bundling 2006 und Force-Directed Edge Bundling 2009 Techniken zur Reduzierung von Visual Clutter in Graphen vor. Während Edge Bundling sich ausschließlich auf hierarchische Graphen anwenden lässt, kann Force-Directed Edge Bundling auch auf nicht hierarchische Strukturen angewandt werden. Die hier betrachteten Kommunikationsnetzwerke stellen hierarchische Netzwerke dar, weshalb im Folgenden das ursprüngliche, hierarchische Edge Bundling betrachtet wird. Dabei fasst Egde Bundling benachbarte Kanten visuell zusammen, indem diese durch B-Splines entlang der vorgegebenen Kantenstruktur dargestellt werden. Abbildung 4.1 zeigt links den Verlauf des resultierenden B-Splines entlang der ursprünglichen Baumkanten. Rechts ist ein Graph vor und nach den Zusammenfassen der Kanten durch Edge Bundling zu sehen.

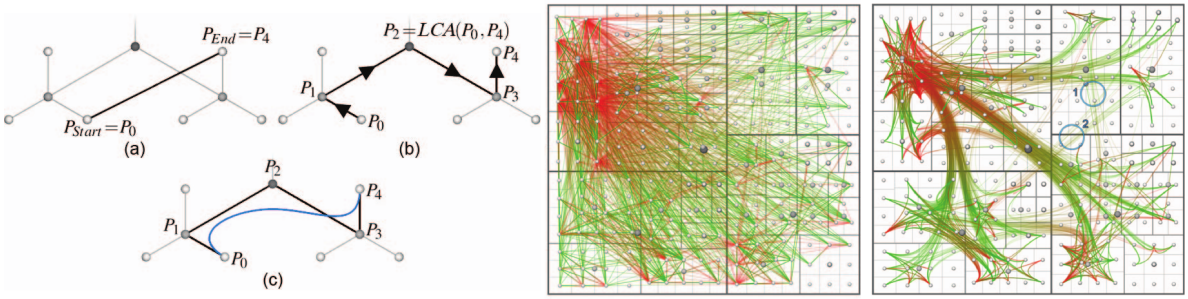


Abbildung 4.1: Links: Verlauf des resultierenden B-Splines entlang der Kanten des ursprünglichen Graphen; Rechts: Graph vor und nach Edge Bundling. Bilder von Holten et al. [Hol06]

Durch Edge Bundling wird Visual Clutter reduziert und das Resultat erscheint deutlich übersichtlicher und strukturierter. Allerdings werden die ursprünglichen Richtungen der Kanten teils stark verfälscht und das Verfolgen von Kanten zwischen Kind und Elternknoten ist meist nicht mehr möglich. Edge Bundling müsste daher zusätzlich zu einer Detail-Ansicht, z.B. dem Knoten-Kanten-Diagramm, angeboten werden. Da beide Darstellungen auf Kanten und Knoten beruhen ist eine gleichzeitige Darstellung jedoch problematisch. Außerdem können mittels Edge Bundling erzeugte Diagramme schlecht zur Darstellung mehrere Kommunikationsnetzwerke verwendet werden. Da der Algorithmus auf einem Graphen operiert, werden auch nur Kanten dieses Graphen berücksichtigt. Beim Überlagern mehrerer Diagramme würden daher erneut Überschneidungen und Verdeckungen auftreten. In der Praxis sollen jedoch möglicherweise viele, sich überdeckende Kommunikationsnetzwerke übersichtlich betrachten werden können.

Eine intuitive Alternative, bei der die gleichzeitige Darstellung mehrerer Netzwerke keine Probleme ergibt, stellen Heatmaps dar. Gerichtete Heatmaps wurden bei ähnlicher Aufgabenstellung bereits von Wu et al. [WLY+14] zur Darstellung der Diffusion von Meinungsbildern verwendet, zu sehen in Abbildung 2.4. Als Grundlage dienen hierzu gerichtete Splats. Auch eine parallele Darstellung zusammen mit dem Knoten-Kanten-Diagramm wäre möglich. Zudem könnten einzelnen Splats als zusätzliche Navigationshilfe dienen, um gezielt einzelne Kanten zu verfolgen. Eine Studie zu partiell

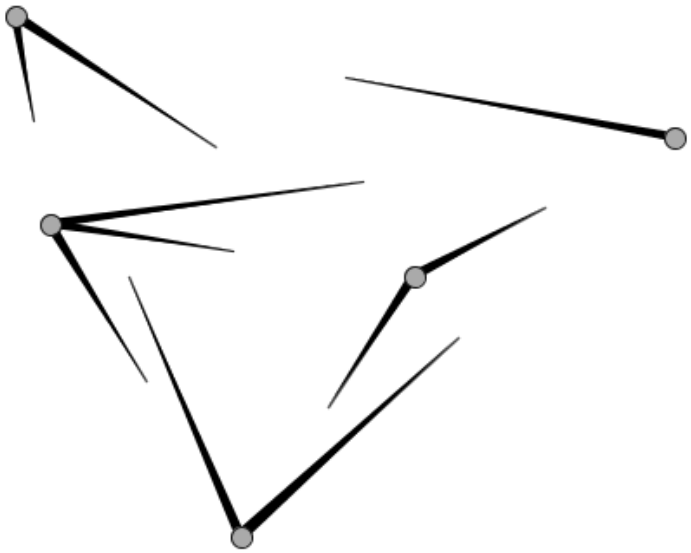


Abbildung 4.2: Graph mit partiell gezeichneten Kanten. Bild von Burch et al. [BVKW12]

gezeichneten Kanten, von Burch et. al [BVKW12], zeigte ,dass diese Visual Clutter reduzieren und sich positiv auf die Lesbarkeit und effiziente Kantenverfolgung auswirken können. Abbildung 4.2 zeigt einen solchen Graph mit partiell gezeichneten Kanten. Man kann sich die zugespitzten Kanten leicht als Splats vorstellen, die bei weniger dicht besiedelten Graphen eine Navigationshilfe darstellen, sich bei dicht besiedelten Graphen jedoch zu einer Heatmap ergänzen.

Aus eben genannten Überlegungen wird in dieser Arbeit zur Darstellung der Kommunikationsnetze ein Knoten-Kanten-Diagramm und eine Heatmap, basierend auf gerichteten Splats, verwendet.

4.2 Beziehen von Twitterdaten einer Konversation

Die Diffusion von Informationen erfolgt auf Twitter durch das *retweeten* und *replien* eines Tweets, der die Ursprungsinformation enthält. Dabei werden im Folgenden der ursprüngliche Tweet, sowie alle Reetweets oder Replies, die sich auf diesen Tweet beziehen, als eine Konversation bezeichnet. Diese Beziehung setzt sich transitiv fort, d.h. ein Reply eines Replys des Original-Tweets gehört ebenfalls zur Konversation. Abbildung 4.3 stellt eine solche Konversation dar. Wird ein neuer Tweet abgesetzt, der ein Reply oder Retweet eines Tweets der aktuellen Konversation ist, so wird dieser ebenfalls der Konversation hinzugefügt. Um eine möglichst vollständige Konversation zu erhalten ist es notwendig alle bereits abgesetzten Tweets abzufragen, um die Konversation bis zum jetzigen Zeitpunkt zu vervollständigen. Zum Anderen müssen jedoch auch neu verfasste Tweets möglichst schnell detektiert und der Konversation hinzugefügt werden.

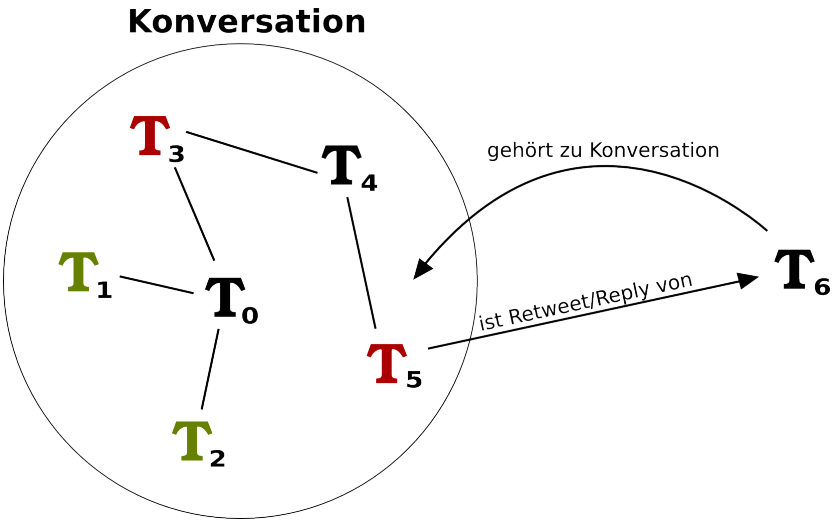


Abbildung 4.3: Eine Konversation bestehend aus verschiedenen Tweets $T_0 \dots T_5$, abgesetzt von drei verschiedenen Benutzern. Die Farben der Tweets stellen die einzelnen Benutzer dar. Ein Reply/Retweet T_6 eines Tweets der Konversation gehört ebenfalls der Konversation an.

4.2.1 Rest-API vs. Streaming-API

Während zur Abfrage bereits abgesetzter Tweets, die möglicherweise weit in der Vergangenheit liegen, nur die Rest-API in Frage kommt, scheint die Streaming-API für die Abfrage der neu hinzukommenden Tweets geeignet zu sein.

Die Streaming-API bietet zwei Möglichkeiten, Retweets und Replies von ausgewählten Tweets zu empfangen. Zum einen kann man über den Follow-Parameter einen Stream empfangen, der alle Replies und nativen Retweets von Tweets relevanter Benutzer enthält. Relevante Nutzer sind in diesem Szenario alle Verfasser von Tweets, die bereits in der Konversation enthalten sind. Als Beispiel kann Abbildung 4.3 betrachtet werden. Dort sind die Tweets, eines jeweiligen Benutzers, schwarz, rot, bzw. grün, makiert. Es nehmen also aktuell drei Benutzer an der Konversation teil. Alle Retweets und Replies auf Tweets dieser Benutzer sind potentielle Tweets der Konversation. In einem weiteren Schritt muss abschließend geprüft werden, ob die über den Stream empfangen Tweets sich tatsächlich auf einen Tweet aus der aktuellen Konversation beziehen oder auf einen anderen Tweet des Benutzers, der nicht zur Konversation gehört. Der Follow-Parameter ist jedoch auf maximal 5000 Benutzer beschränkt. Dies kann möglicherweise zu Problemen führen, wenn eine Vielzahl an Konversationen und somit möglicherweise eine Vielzahl an Benutzern gleichzeitig betrachtet werden sollen. Jedes mal, wenn ein neuer Benutzer zur Konversation hinzukommt, muss außerdem der Follow-Parameter um die jeweilige Benutzer-ID erweitert werden. Ein Update des Parameters hat jedoch eine Neuverbindung des Streams zur Folge. Bei zu vielen aufeinander folgenden Neuverbindungen wird der API-Nutzer für einige Minuten von Twitter gesperrt. Tritt diese Sperrung häufiger auf, so behält Twitter es sich vor, die verwendete IP-Adresse auf unbestimmte Zeit zu sperren. Über die Rest-API abgefragte Tweets können jedoch in großen Mengen und hoher Frequenz zur aktuellen Konversation

hinzustoßen. Enthalten diese Tweets viele, der Konversation fremde Benutzer, müsste der Stream möglicherweise ebenfalls in sehr hoher Frequenz einem Update unterzogen werden, was zu eben beschriebenen Problemen führen kann.

Die zweite Möglichkeit relevante Tweets über die Streaming-API abzufragen, bieten die von Twitter neu eingeführten Page-Streams, welche auf ähnliche Weise funktionieren. Allerdings befinden sich diese zum Zeitpunkt der Arbeit erst in der Bata-Phase und sind während dessen auf das Verfolgen von nur 100 Benutzern limitiert.

Unter diesen Gesichtspunkten bietet es sich an, die Rest-API zur Abfrage aller Tweets zu verwendet. Dazu werden die Retweets und Replies jedes Tweets aus der Konversation in regelmäßigen Zeitabständen abgefragt um auch Tweets, die seit der letzten Abfrage neu hinzugekommen sind, zu erhalten. Ein offensichtlicher Nachteil dieser Methode ist jedoch, dass Tweets nicht live, sondern leicht zeitversetzt, in regelmäßigen Intervallen, der Konversation hinzugefügt werden können. Außerdem muss auf die Abfragelimitierungen der Rest-API geachtet werden.

4.3 Aufbau eines Datenmodells und des Konversationsbaums

Sind die zugrundeliegenden Twitterdaten vorhanden, können diese, wie bereits angesprochen, einzelnen Konversationen zugeordnet werden. Das Netzwerk, dass die Tweets einer Konversation miteinander bilden, lasst sich in einem Baum darstellen. Die Darstellung als Baum folgt aus den Eigenschaften der Tweets einer Konversation. Jeder Tweet, der Teil der Konversation ist, bezieht sich auf genau einen anderen Tweet der Konversation. Weiter existiert genau ein Tweet innerhalb der Konversation, der sich auf keinen anderen Tweet bezieht. Dieser stellt somit die Wurzel des Baums dar. Der Graph der sich aus den Tweets der Konversation bilden lässt, ist daher immer einen Baum. Dieser Baum einer Konversation wird im Folgenden als Konversationsbaum bezeichnet.

4.3.1 Finden der Wurzel

Die Wurzel eines jeden Konversationsbaumes ist ein unabhängiger Tweet, der weder Retweet noch Reply eines anderen Tweets ist. Da nicht davon ausgegangen werden kann, dass der ursprüngliche Tweet, der die Wurzel bildet, bekannt ist, soll es möglich sein,einen Konversationsbaum aufzubauen ohne diesen zu kennen. Ausgangspunkt ist daher ein beliebiger Tweet der Konversation. Von diesem werden anschließend rekursiv dessen Elternknoten gesucht. Dies geschieht solange, bis ein Tweet erreicht wird, der keinen Elternknoten mehr besitzt. Dieser Tweet ist der Wurzelknoten der Konversation. Abbildung 4.4 zeigt den prinzipiellen Aufbau des Konversationsbaumes, wobei die neu hinzugefügten Tweets in jedem Schritt rot markiert sind. Knoten T_9 ist der Start- und Knoten T_1 der Wurzelknoten. Die Vervollständigung bis zur Wurzel findet in den Schritten 1 und 2 statt. Die Suche nach den Elternknoten gestaltet sich dabei relativ einfach. Man kann hierzu das $retweeted\_status$-Feld bzw. das $in\_reply\_to\_status\_id\_str$-Feld des entsprechenden Tweets benutzten. Ist der aktuelle Tweet ein Retweet so enthält das $retweetet\_status$-Feld den Tweet auf den sich dieser bezieht. Ist der aktuelle Tweet ein Reply, so liefert das $reply\_to\_status\_id\_str$-Feld die Tweet-ID des vorangegangenen Tweets als String zurück. Über die Tweet-ID lässt sich anschließend der entsprechende Tweet beziehen.

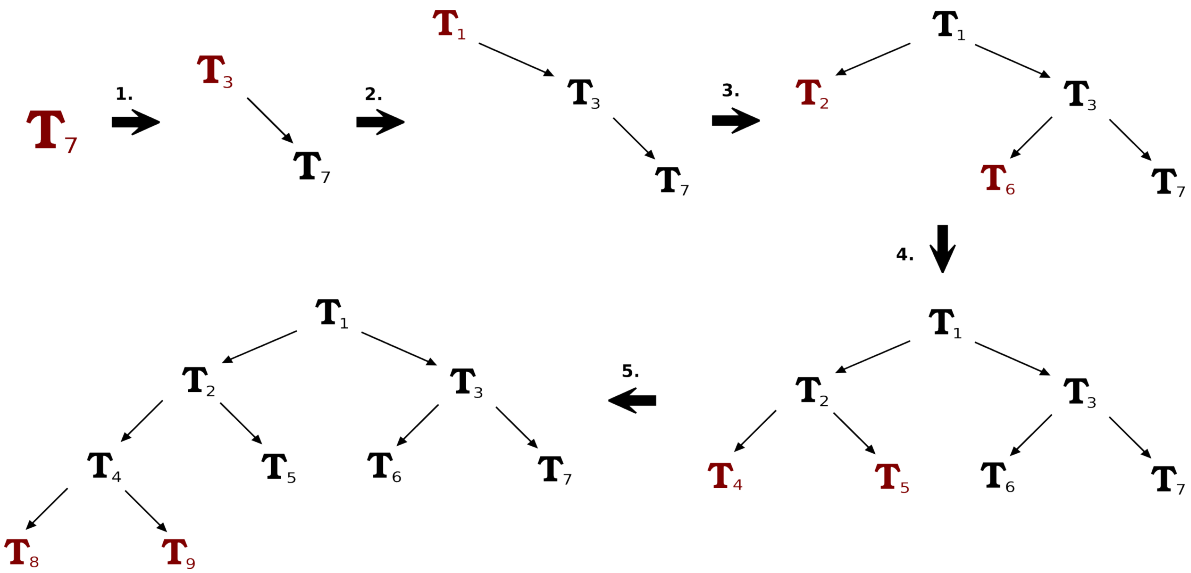


Abbildung 4.4: Aufbau des Konversationsbaumes

Eine Konversation kann durch jeden enthaltenen Tweet eindeutig identifiziert werden. Das Finden des Wurzel-Tweet einer Kommunikation ist dabei dennoch von besonderer Bedeutung. Betrachtet man zwei Tweets, von denen man nicht weiß ob diese der selben Konversation angehören, ist die einfachste Möglichkeit dies zu überprüfen, die rekursive Suche nach der Wurzel. Spätesten wenn die Wurzel gefunden ist, kann eindeutig bestimmt werden, ob die Tweets der selben Konversation angehören. Finden sich auf dem Weg zur Wurzel zwei übereinstimmende Knoten, so gehören die Tweets der selben Konversation an. Sind alle Knoten bist einschließlich der Wurzeln verschieden, so handelt es sich um zwei verschiedene Konversationen.

4.3.2 Rekursives Entfalten von Teilbäumen

Ist der Konversationsbaum, ausgehend von einem Startknoten, bis zur Wurzel vervollständigt, kann begonnen werden den Konversationsbaum, der im Moment aus einer linearen Liste besteht, weiter zu vervollständigen. Hierzu können die Kindknoten bereits vorhandener Knoten rekursiv erschlossen werden. Somit entfaltet sich der Baum, bis keine weiteren Tweets der Konversation gefunden werden können. Die Schritte 3, 4 und 5 in Abbildung 4.4 zeigen das rekursive Entfalten der Teilbäume.

4.4 Darstellung des Verbreitungsbaums auf der Karte

Nach dem das grundlegende Konzept zum Aufbau des Konversationsgraphen bekannt ist, muss über dessen Darstellung entschieden werden. Da Baumstrukturen weit verbreitet und gut untersucht sind, stehen für deren Repräsentation bereits eine Vielzahl an Möglichkeiten zur Verfügung. So können Bäume beispielsweise in hierarchischen und radialen Layouts, als Knoten-Kanten-Diagramm,

dargestellt werden. Bei der Darstellung auf einer Karte können jedoch bestimmte Einschränkungen auftreten.

4.4.1 Mapping von Eigenschaften auf visuelle Merkmale des Knoten-Kanten-Diagramm

Um den Konversationsbaum, als Knoten-Kanten-Diagramm, auf einer Karte darzustellen, werden Merkmale benötigt, die eine Entsprechung bezüglich beiden, dem Tweet und der Karte, haben. Eine logische Herangehensweise ist es, einen Bezug zwischen der geografischen Lokation eines Tweets und der entsprechenden Position auf der Karte herzustellen. Dabei wird ein Tweet genau auf die Position auf der Karte abgebildet, die seinem Längen- und Breitengrad entspricht. Da jeder Knoten bzw. Tweet genau eine vorgegebene geografische Lokation besitzt, ist dessen Position auf der Karte eindeutig festgelegt. Durch diese Einschränkungen können jedoch keine der bereits genannten Layout-Techniken zur Repräsentation des Baumes als Knoten-Kanten-Diagramm verwendet werden, da diese auf der freien, bzw eingeschränkten Möglichkeit zur Platzierung der Knoten beruhen.

Das Knoten-Kanten-Diagramm kann neben der Position eines Tweets auch zur Kodierung weiterer Eigenschaften des Konversationsbaums verwendet werden. So kann die Stimmung eines Tweets als Farbe des zugehörigen Knotens dargestellt werden. Dabei scheint es schlüssig eine negative Stimmung mit der Farbe rot und eine positive Stimmung mit der Farbe grün zu kennzeichnen, da dies mit der Assoziation der meisten Menschen übereinstimmt. Somit könnten Farbveränderungen innerhalb des Knoten-Kanten-Diagramms als Stimmungsänderungen interpretiert werden.

Ein weiteres Merkmal, das im Konversationsbaum dargestellt werden kann, ist die Art des Tweets, um den es sich handelt. Dies kann beispielsweise in der Form eines Knotens kodiert werden. Ein runder Knoten zeigt an, dass es sich bei dem Knoten um einen Reply handelt, während ein viereckiger Knoten signalisiert, dass es sich um einen Retweet handelt. Dadurch kann optisch unterschieden werden, in welcher Beziehung ein Kindknoten zu seinem Elternknoten steht.

4.4.2 Unterscheidung zwischen Tweets mit und ohne Geolokation

Die große Mehrheit aller Tweets wird ohne Angaben zur geografischen Lokation abgesetzt und kann somit nicht auf einer Karte verortet werden. Die Unterscheidung zwischen Tweets mit und Tweets ohne Geolokation spielt daher beim Aufbau der Konversationsgraphen, vor Allem in Zusammenhang mit dessen Repräsentation auf einer Karte, eine entscheidende Rolle. Zwar wurden bereits Möglichkeiten zur alternativen Identifizierung der geografischen Lokationen von Tweets ohne GPS-Koordinaten vorgestellt, diese sind jedoch teils ungenau und können nicht allen Tweets zweifelsfrei eine geografische Lokation zuordnen. Aufgrund dieser Unsicherheiten wurden Methoden zur Verortung weiterer Tweets an dieser Stelle nicht betrachtet. Diese bieten jedoch sicherlich interessante Möglichkeiten zur weiteren Steigerung der Anzahl geographisch verortbarer Tweets. Um den Konversationsbaum dennoch möglichst genau darstellen zu können, wird dieser leicht modifiziert. Dazu wird eine Unterscheidung von normalen Knoten und Geo-Knoten eingeführt. Geo-Knoten

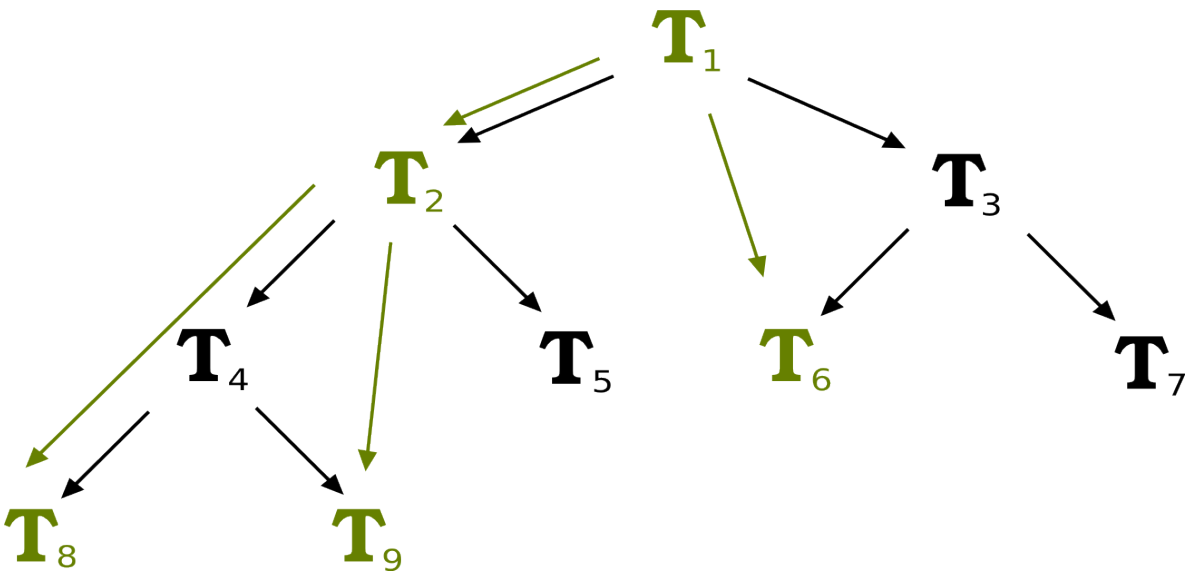


Abbildung 4.5: Der modifizierte Konversationsbaum beinhaltet zusätzliche Verbindungen zwischen Knoten mit geografischer Lokation.

sind all diejenigen Knoten, welche eine geografische Lokation in Form von GPS-Koordinaten besitzen. Der Geo-Elternknoten eines Geo-Knotens ist der nächste Geo-Knoten auf dem Pfad zu Wurzel bzw. die Wurzel selbst. Allen Geo-Knoten werden zusätzliche Kanten eingefügt, die sie mit ihren Geo-Elternknoten verbinden. Daraus resultiert ein weiterer eigenständiger Baum, der im Folgenden Geo-Konversationsbaum genannt wird. Abbildung 4.5 bildet sowohl den Konversationsbaum als auch den resultierenden Geo-Konversationsbaum ab. Dabei sind die Tweets mit Geolokation bzw. Geo-Knoten und deren Verbindungen untereinander grün dargestellt. Der grüne Baum ist also derjenige, der letztlich auf einer Karte dargestellt werden kann und dabei dem originalen Konversationsbaum am nächsten kommt. Die Wurzel ist unabhängig von ihrer geografischer Lokation immer Bestandteil des Geo-Konversationsbaumes, auch wenn sie ohne Lokation später nicht angezeigt werden kann. Dies ist notwendig um einen zusammenhängenden Geo-Konversationsbaum zu gewährleisten.

4.5 Aufbau der Heatmap

Da direkt auf Karten abgebildete Graphen in Form von Knoten-Kanten-Diagrammen mit fixen Positionen der Knoten, wie bereits beschrieben, nicht auf gewöhnliche Layout-Techniken zurückgreifen können, ist deren Lesbarkeit auf Karten möglicherweise stark eingeschränkt. Um diesem Problem entgegenzuwirken wird, wie bereits angesprochen, eine Heatmap verwendet, welche wie das Knoten-Kanten-Diagramm aus dem Geo-Konversationsbaum generiert wird.

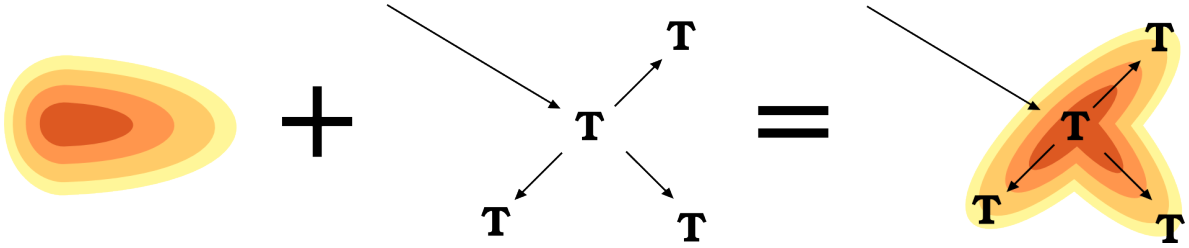


Abbildung 4.6: Aufbau der Heatmap aus einzelnen Splats

4.5.1 Verwendung von Splats

Um eine Heatmap zu erzeugen, welche die Verbreitung von Informationen nachempfindet, werden den einzelnen Geo-Knoten gerichtete Splats zugewiesen. Jeder Geo-Elternknoten bekommt, je Geo-Kindknoten, einen Splat zugewiesen, der in Richtung des Kindknoten gerichtet ist. Abbildung 4.6 zeigt den konzeptuellen Aufbau der Heatmap. Kombiniert man die einzelnen Splats in Richtung der Kinder, geben diese das Gesamtbild der Verbreitungsrichtung wieder. An Stellen an denen sich viele Splats überlagern, summiert sich deren Intensität auf.

4.5.2 Bildüberlagerung durch Alphablending

Eine Möglichkeit die Splats zu kombinieren stellt die Verwendung halbtransparenter Bilder dar. Bei Überlappung zweier Bilder addieren sich deren Farbwerte in Abhängigkeit der Transparenz auf. Zur Bestimmung der resultierenden Endfarbe kann beispielsweise der Porter Duff Algorithmus verwendet werden. Dieser berechnet die neue Farbe und Transparenz nach folgender Vorschrift, wobei A, B und C Farben und α_A, α_B und α_C Alphatransparenzwerte darstellen.

Definition 4.5.1

$$C = \frac{1}{\alpha_C}(\alpha_A A + (1 - \alpha_A)\alpha_B B), \quad \alpha_C = \alpha_A + (1 - \alpha_A)\alpha_B \quad \textit{[Wikb]}$$

Die Verwendung von Alphablendung ist einfach zu realisieren und hat in der Regel eine gute Laufzeit. Es muss lediglich das halbtransparente Bild eines Splats vorliegen. Dieses wird geladen, rotiert und an die entsprechende Stelle gezeichnet. Die Farben der einzelnen Pixel werden dann, nach der ober beschriebenen Vorschrift, schrittweise errechnet. Möchte man eine sehr große Anzahl von überlappenden Splats darstellen ist man jedoch von der Auflösung des Alphakanals abhängig. Außerdem können die Eigenschaften der Splats nicht beliebig zur Laufzeit geändert werden. Möchte man verschiedene Splats benutzen müssen mehrere Bilder vorliegen oder vorhandene Bilder zur Laufzeit angepasst werden. Es können jedoch nicht alle Variationen eines Splats gespeichert werden und eine Anpassung des Bildes zur Laufzeit kann nicht ausreichen sein, oder zu schlechten Laufzeiten führen.

4.5.3 Errechnen der Splats

Da Alphablending die Möglichkeiten zur Anpassung der Splats limitiert, ist eine besser geeignete Variante, die direkte Berechnung der Splats zur Laufzeit. Dabei kann den Splats eine beliebige Kernelfunktion, beispielsweise ein Gauß-, Chauchy- oder Picard-Kernel, zugrundeliegen, deren Parameter zur Laufzeit angepasst werden können. Durch Anpassung der Parameter lässt sich beispielsweise die Größe oder Streckung eines Splats festlegen. So kann ein Splat in Abhängigkeit der Distanz zwischen Kind- und Elterknoten dargestellt werden. Der Splat wird dabei weiter in Richtung des Kindknotens gestreckt, je weiter dieser vom Elternknoten entfernt ist.

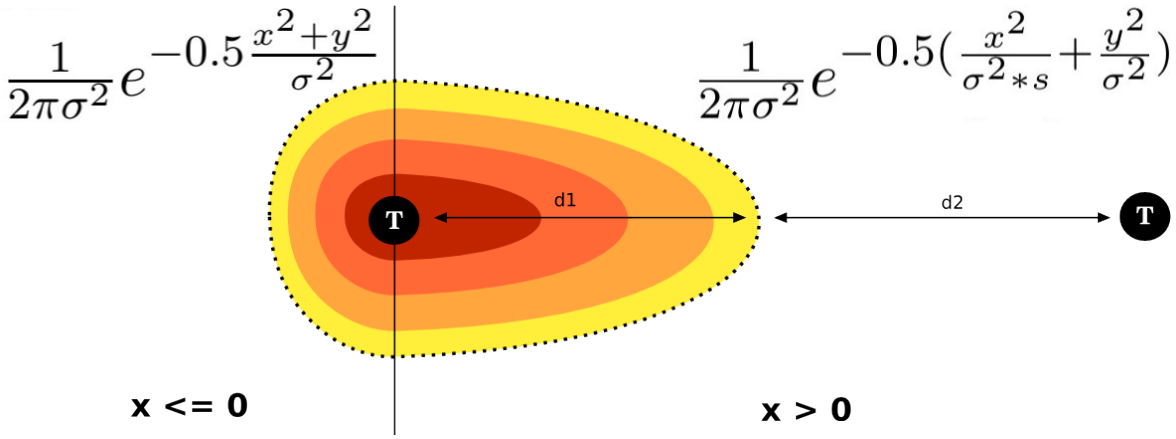


Abbildung 4.7: Splat bestehend aus zwei Exponentialfunktionen

Wu et al. stellten mit OpinionFlow [WLY+14] einen Gaußkernel, der zur Darstellung der Meinungsverbreitung geeignet ist, vor. In dieser Arbeit wird eine abgeänderte Variante dieses Gaußkernels zur Repräsentation der Splats betrachtet. Abbildung 4.7 zeigt einen Splat der von einem Elternknoten zu dessen Kindknoten gerichtet ist. Um gerichtete Spalts zu erhalten werden zwei unterschiedliche Exponentialfunktionen verwendet.

Definition 4.5.2

$$k(x,y,s) = \begin{cases} \frac{1}{2\pi\sigma^2}e^{-0.5(\frac{x^2}{\sigma^2*s}+\frac{y^2}{\sigma^2})} & x \leq 0 \\ \frac{1}{2\pi\sigma^2}e^{-0.5\frac{x^2+y^2}{\sigma^2}} & x > 0 \end{cases}$$

Dabei ist s der neu eingeführte Streckfaktor des Splats. Dieser kann, durch Anpassung eines Parameters m so gewählt werden, dass die Länge der Splats bestimmte Vorgaben bezüglich der Entfernung zwischen Elternknoten und Kindknoten erfüllt. Der Streckfaktor wird mit folgender Gleichung berechnet.

Definition 4.5.3

$$s = -\frac{m*d^2}{(2*\sigma^2*ln(\frac{breakValue}{normalization})} \; mit \; m = \frac{1}{2}$$

Wird $m = \frac{1}{2}$ gewählt, so ergibt sich der Streckfaktor s, so dass der Splat die Hälfte der Distanz zwischen Eltern- und Kindknoten überbrückt. Die Distanzen $d1$ und $d2$ aus Abbildung 4.7 sind also gleich groß. Die Konstante $normalization$ stellt die Normalisierungskonstante dar und wurde zum besseren allgemeinen Verständnis der Formel nicht ausgeschrieben. Der Parameter $breakValue$ gibt den kleinsten Wert an, bis zu dem die Berechnung des Splats fortgesetzt werden soll. Dieser Parameter muss später auch beim Zeichnen der Splats berücksichtigt werden. Die Ausdehnung des Splats, ganuer der Parameter m, soll später über das Benutzerinterface manipuliert werden können.

Zur Berechnung der Distanz wird neben der Position des Elternknotens auch Position des Kindknotens benötigt. Damit sich die Größe der Splats beim Aus- und Einzoomen automatisch anpasst, müssen die Geopositionen der Knoten zunächst in Pixelpositionen, abhängig von der aktuellen Zoomstufe der Karte, umgerechnet werden. Erst dann kann die Distanz zwischen den Knoten ermittelt werden. Die Pixeldistanz zwischen zwei Punkten ändert sich im Gegensatz zur geografischen Distanz beim Aus- und Einzoomen der Karte. Die Splats werden also automatisch mitskaliert. Letztendlich erhält man ein Pixelgitter in den Ausmaßen des Splats in welchem die Werte der Exponentialfunktionen gespeichert sind.

4.5.4 Voraggregation in Pixelgitter

Nachdem der Splat erzeugt wurde, muss dieser mit den anderen Splats kombiniert werden, sodass letztendlich eine Heatmap entstehen kann. Hierzu muss ein Pixelgitter in den Ausmaßen des aktuelle Viewports in Pixeln, angelegt werden. Mittels den Pixelpositionen von Eltern- und Kindknoten, wird das zuvor erzeugt Pixelgitter des Splats innerhalb des Viewport-Pixelgitters positioniert und gedreht. Die Werte des Splat-Gitters werden anschließend in das Viewport-Gitter übertragen und dort mit bereits vorhandenen Werten zuvor berechneter Splats aufsummiert.

4.5.5 Errechnen des Farbschemas

Sind alle Splats in das Viewport-Gitter übertragen, müssen diesen Farben zugewiesen werden. Dabei soll eine lineare Farbskalierung in Abhängigkeit vom aktuell höchsten Wert des Viewport-Gitters verwendet werden. Listing 4.1 zeigt die Berechnung einer solchen Farbcodierung. Zunächst wird der maximale Wert innerhalb des aktuellen Viewport-Gitters ermittelt. Die Werte der einzelnen Gitterzellen werden anschließend durch diesen maximalen Wert geteilt, um deren Position auf der Farbskala zu bestimmen (Zeile 3). Zusätzlich kann ein Faktor zwischen 0 und 1 zur Gewichtung des maximalen Gitterwerts festgelegt werden. Ein Wert von 0.5 wirkt sich wie eine Verdopplung der betrachteten Werte aus, stellt also eine lineare Skalierung der Werte um den Faktor 2 dar. Werte die nach dieser Skalierung außerhalb der Farbskala liegen, weil diese nun größer als der ursprünglich maximale Wert sind, werden wieder auf den maximalen Wert herabgesetzt (Zeile 5, 6 und 7). Der

zugehörige Pixel wird anschließend in der entsprechenden Farbe eingefärbt. Durch die zusätzlichen Gewichtungsfaktor des maximalen Gitterwertes können bei Bedarf auch schwächere Details der Heatmap hervorgehoben werden.

```
Color color;

int colorIndex = (int)(value*(numberOfColors-1)/(maxValue*faktor));

if(colorIndex > (numberOfColors-1)){
      colorIndex = numberOfColors-1;
}

color = colorScale[colorIndex];
```

Listing 4.1: Programmausschnitt für Zuweisung der Farben

5 Umsetzung

Dieses Kapitel widmet sich hauptsächlich der Anwendung, die während der Arbeit implementiert wurde. Dabei werden zunächst die einzelnen Komponenten der Benutzeroberfläche vorgestellt. Darauf folgend wird das Beziehung von Twitterdaten und deren anschließende Verarbeitung durch die Anwendung erläutert. Zuletzt wird noch einmal auf die Erzeugung der Heatmap und die Berechnung der zugrundeliegenden Splats eingegangen.

5.1 Benutzeroberfläche

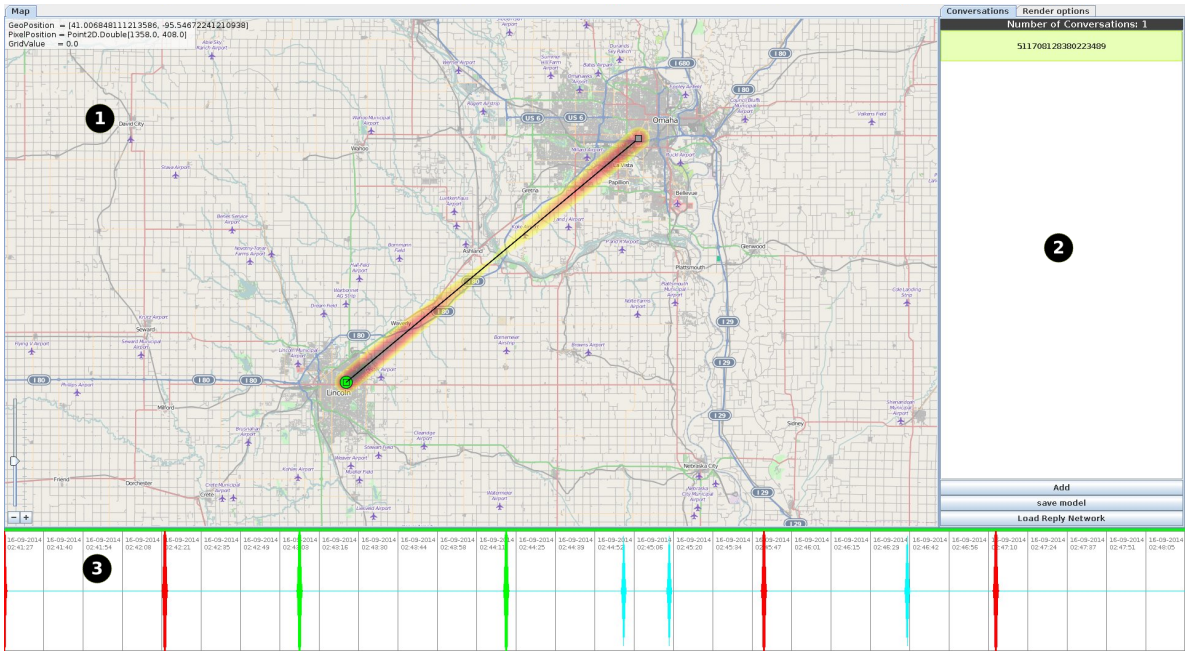


Abbildung 5.1: Benutzeroberfläche der Anwendung, bestehend aus Kartenansicht (1), Zeit- und Abfragelimitleiste (3) und einer Seitenleiste zur Darstellung der geladenen Konversationen und Renderoptionen (2).

Die Benutzeroberfläche des Programms, zu sehen in Abbildung 5.1, ist in drei Bereiche aufgeteilt. Den meisten Raum innerhalb des Fenster nimmt die interaktive Karte ein (1). Diese wird ebenfalls zur Darstellung des Konversationsgraphen und der Heatmap genutzt. Auf der rechten Seite des Fensters befindet sich die Konversationsleiste, welche eine Übersicht über die aktuell dargestellten Konversationen anzeigt (2). Zudem können in einem separaten Tab Optionen bezüglich der Visualisierung

gesetzt werden. Im unteren Bereich des Fensters befindet sich eine Leiste zur zeitlichen Einordnung der Tweets und eine Leiste zur Visualisierung des Abfragelimits (3).

5.1.1 Kartendarstellung

Zur Darstellung der Karte wird das OpenSource-Tool JXMapViewer2 [Ste] verwendet. Es bietet unter anderem spezielle Java-Swing-Komponenten zur Darstellung von Kartenmaterial und zugehörigen Kontrollschaltflächen an. Des Weiteren unterstützt es Karten-Overlays, die Verwendung von verschiedenen Kartendiensten und offline Kartenmaterial. In dieser Arbeit werden die Kartendaten von Openstreetmaps [Fou] genutzt.

Auf der Karte kann durch Drag & Drop navigiert werden. Zum Ein- oder Auszoomen innerhalb der 15 angebotenen Zoomstufen kann das Mausrad oder die Kontrollschaltfläche verwendet werden. In der oberen linken Ecke der Kartendarstellung wird dauerhaft ein Informations-Overlay angezeigt, welches unter anderem die geografische Position des Mauszeigers innerhalb der Karte anzeigt. Die Darstellung des Knoten-Kanten-Diagramms und der Heatmap erfolgt ebenfalls mittels Overlays. Diese können den Renderoptionen wahlweise ein und ausgeschalten werden.

5.1.2 Konversationsleiste und Renderoptionen

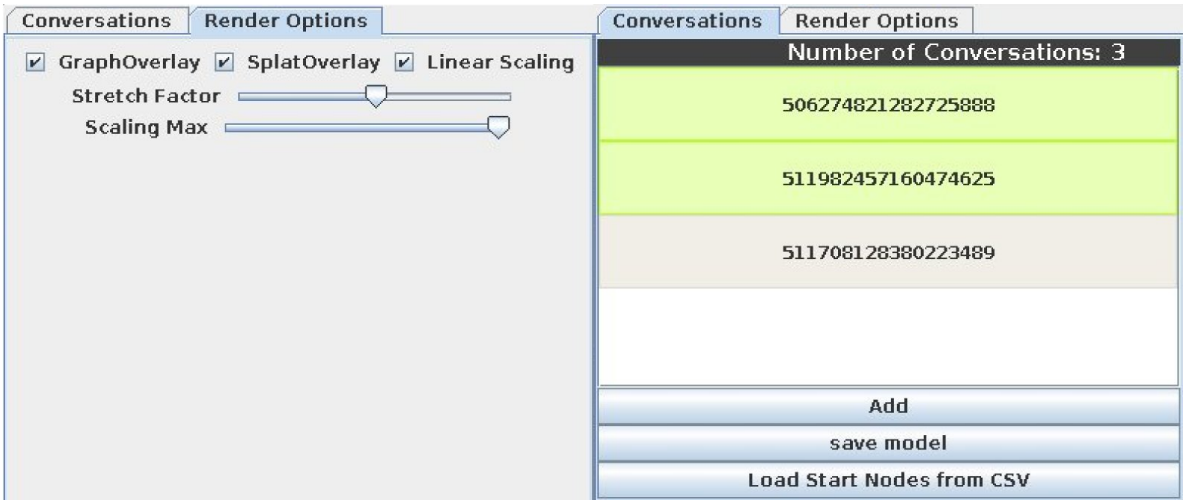


Abbildung 5.2: Links: Renderoptionen zur Visualisierung; Rechts: Konversationsleiste mit drei geladenen Konversationen

Die Konversationsleiste gibt einen Überblick über die momentan geladenen Konversationen. Neben der Möglichkeit einzelne Konversationen durch Angabe der ID des Start-Tweets zu laden, lassen sich auch CSV-Dateien einlesen, deren beinhaltete Tweets als Start-Tweets zum Aufbau von Konversationen verwendet werden. Die beinhalteten Tweets aller aktuell geladenen Konversationen lassen sich zudem wiederum in eine CSV-Datei speichern.

Aktive bzw. momentan auf der Karte angezeigte Konversationen werden grün hinterlegt dargestellt. Grau hinterlegte Konversationen sind hingegen inaktiv und werden nicht auf der Karte angezeigt. Durch klicken auf die jeweilige Konversation kann deren Status von aktiv zu inaktiv und umgekehrt geändert werden. Die Konversationsleiste ist in Abbildung 5.2 rechts zu sehen.

Hinter dem zweiten Tab verbergen sich Optionen, welche die aktuelle Darstellung des Konversationsgraphen bzw. der Heatmap auf der Karte betreffen. Die Darstellung des Graphen und der Heatmap kann einzeln ein- und ausgeschaltet werden. Außerdem kann die Größe der einzelnen Splats der Heatmap reguliert und zu einer alternativen Farbskalierung gewechselt werden. Über einen Slider kann zudem die lineare Farbskalierung verändert werden, um niedrigere Werte stärker zu gewichten. Die Renderoptionen sind in Abbildung 5.2 links zu sehen.

5.1.3 Zeitleiste und Abfragelimitleiste

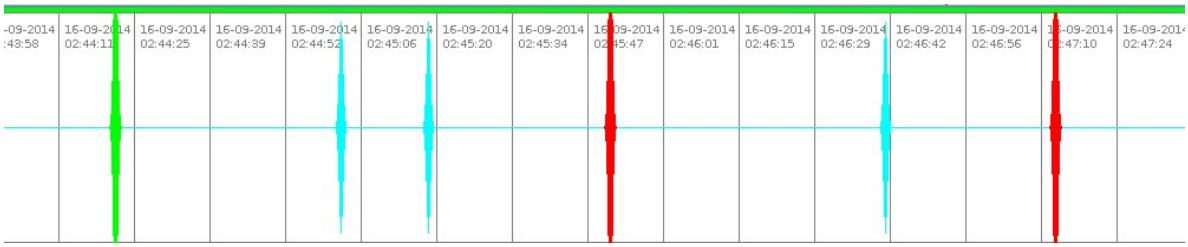


Abbildung 5.3: Zeitleiste und Abfragelimitleiste

Die Zeitleiste, dargestellt in Abbildung 5.3, gibt einen Überblick über das zeitliche Auftreten der Tweets. Dabei werden alle Tweets aus aktuell aktiven Konversationen berücksichtigt. Fallen zwei Tweets auf den gleichen, bzw. sehr eng beieinander liegende Zeitpunkte, summiert sich der Ausschlag der Zeitleiste an den entsprechenden Stellen auf. Je höher der Ausschlag, desto höher das Aufkommen von Tweets in diesem Zeitabschnitt. Außerdem werden die Tweets zusätzlich farbig nach Stimmung unterschieden. Während die Farben grün und rot Tweets mit positiver bzw. negativer Stimmung kennzeichnen, steht die Farbe blau für Tweets mit neutraler Stimmung.

Die Abfragelimitleiste, in Abbildung 5.3 über der Zeitleiste, dient der Anzeige der Zeit, die gewartet werden muss, bis ein neues Zeitfenster anbricht und die Abfragekapazität der Rest-API erneut aufgestockt wird. Sie ist zu Beginn komplett grün gefärbt und signalisiert, dass noch Kapazität für weitere Abfrage vorhanden ist. Ist das Abfragelimit erreicht, füllt sich die Leiste von links nach rechts, in Abhängigkeit der verbleibenden Zeit, die bis zum Erneuten aufstocken der Abfragekapazität aussteht, mit einem roten Balken. Erreicht der rote Balken den rechten Rand, färbt sich dieser wieder grün und es kann erneut abgefragt werden.

5.2 Abfrage von Twitterdaten mit Rest-API und Twitter4J

Twitterdaten werden von der Anwendung mit Hilfe der Rest-API direkt von Twitter oder aus einem lokalen Datenbestand geladen. Zur Anbindung an die Twitter-APIs wird Twitter4J verwendet. Twitter4J [Teab] ist eine freie Java-Bibliothek, die alle wichtigen Funktionen der Twitter-API 1.1 unterstützt. Dabei müssen Retweets und Replies auf unterschiedliche Weise bezogen werden.

5.2.1 Beziehen von Retweets

Die Twitter Rest-API und Twitter4J bieten von Haus aus eine Funktion zur gezielten Abfrage von Retweets eines bestimmten Tweets an. Diese gibt jedoch nur die jeweils letzten 100 Retweets eines Tweets zurück. Dies kann zu verschiedenen Problemen führen. Zum einen sind Retweets, die zum Zeitpunkt der ersten Abfrage nicht zu den letzten 100 gehören, nicht mehr erreichbar, zum Anderen ist es möglich Retweets zu verpassen, wenn zwischen zwei Abfrageintervallen mehr als 100 Retweets abgesetzt wurden. Wählt man die Zeitintervalle zwischen den Abfragen entsprechend kurz, sollten in den meisten Fällen jedoch keine Probleme auftreten.

5.2.2 Beziehen von Replies

Um Replies von bestimmten Tweets abzufragen bieten die Rest-API und Twitter4J keine expliziten Funktionen an. Daher müssen zunächst alle Replies abgefragt werden, die an den Urheber des relevanten Tweets gerichtet sind. Anschließend wird überprüft welcher der zurückgegebenen Replies sich tatsächlich auf den relevanten Tweets bezieht. Um die Menge der zurückgegebenen Tweets weiter einzuschränken, können außerdem nur diejenigen Replies angefragt werden, die zeitlich nach dem relevanten Tweet abgesetzt wurden. Hierzu kann das SinceID-Parameter verwendet werden.

5.3 Beziehen von Twitterdaten aus Offline-Datenbestand

Neben der Abfrage von Twitterdaten direkt von Twitter, werden in dieser Arbeit zusätzlich Twitterdaten aus einem zuvor gesammelten Datenbestand verwendet. Neben schnelleren Zugriffszeiten kann so die Abfrage-Limitierung der Twitter-API umgangen werden. Über große Zeiträume hinweg gesammelte Daten können jedoch schnell sehr speicherintensiv werden und sich daher oftmals nur nach spezieller Vorfilterung effizient nutzen lassen.

5.3.1 Ursprung und Umfang der Daten

Die Twitterdaten, die im Rahmen dieser Arbeit zur Verfügung standen, wurden am Institut für Visualisierung und Interaktive Systeme der Universität Stuttgart zu Forschungszwecken gesammelt und umfassen alle Replies mit geografischer Lokation, die im August und September 2014 weltweit abgesetzt wurden. Dabei beinhalten die Daten aus August 81.665.270 Replies und die Daten aus September 67.777.461 Replies. Dies entspricht einem Speicherbedarf von 14,3 GB für die Daten aus

August und 11,9 GB für die Daten aus September. Abbildung 5.4 zeigt die Anzahl der abgesetzten

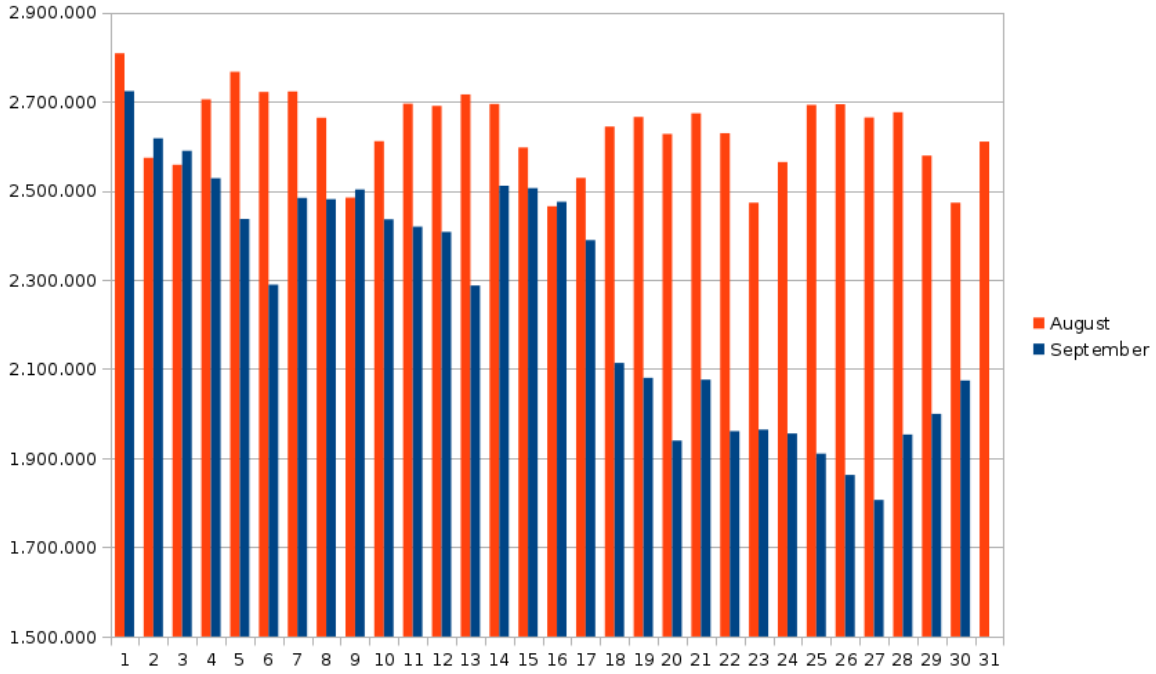


Abbildung 5.4: Im August und September abgesetzte Tweets pro Tag

Replies aus August und September tagesgenau auf. Insgesamt wurden im August 2014, vier Tage ausgenommen, täglich jeweils mehr Replies mit geografischer Lokation abgesetzt, als im September 2014. Die durchschnittliche Anzahl der abgesetzten Replies liegt damit im August mit 2.634.363,5 Replies pro Tag etwa 16,6 % höher als im September mit 2.259.248,7 Replies pro Tag. Weiterhin fällt auf, dass die Anzahl der täglich abgesetzten Replies in der zweiten Septemberhälfte deutlich nachlässt, während sich die täglich abgesetzten Replies aus August durchweg auf einem Level bewegen. In Vergleich mit der ersten Hälfte des Septembers 2014 wurden in der zweiten Septemberhälfte ca. 17,9 % weniger Replies mit geografischer Lokation abgesetzt.

5.3.2 Aufbau und Inhalt der Daten

Die Replies des Offline-Datenbestandes sind tagesgenau in CSV-Dateien gespeichert. Jeder Eintrag beinhaltet dabei die folgenden Informationen:

- Tweet ID
- Nutzer ID
- Zeitstemplel der Form YYYYMMDDhhmmss

- Breitengrad
- Längengrad
- Ortsangabe
- Inhalt des Tweets
- ID des ursprünglichen Tweets
- Nutzer ID des Urhebers des ursprünglichen Tweets
- Nutzername des Urhebers des ursprünglichen Tweets

```
#tweetID          #userID     #timeStamp  #latitude   #longitude  #placeName                          ...

514443495889969153 2227925374 20140923155741 -23.503423 -46.355668 Itaquaquecetuba, Sao Paulo        ...
509738067868254208 130967354  20140910162000 14.588364 -90.549368 Guatemala                          ...
515773749195337729 59650653   20140927080339 29.730246 -95.625099 Houston, TX                       ...
509725396716109824 223077082  20140910152939 1.398299 110.344336  Kuching, Bahagian Kuching         ...
514383007184990208 1026157909 20140923115720 -7.515788 110.800247 Ngemplak, Boyolali                ...
```

Listing 5.1: Tweets in CSV-Datei

Da sich die Daten aus dem offline Bestand mit den per Twitter-API verfügbaren Daten decken und darüber hinaus eine bessere Verfügbarkeit aufweisen, werden sie in dieser Arbeit als Datenquelle bevorzugt. Ausnahmefälle bei dem sich die Daten des Bestands nicht mit den Daten der API decken, können durch nachträgliches Löschen eines Tweets oder das Entfernen der Lokation durch den Benutzer auftreten. In beiden Fällen würden die Tweets des offline Datenbestand jedoch mehr Informationen als die der API beinhalten. Somit stellen diese Fälle kein Problem dar. Listing 5.1 zeigt einen Auzug einer CSV-Datei aus September 2014.

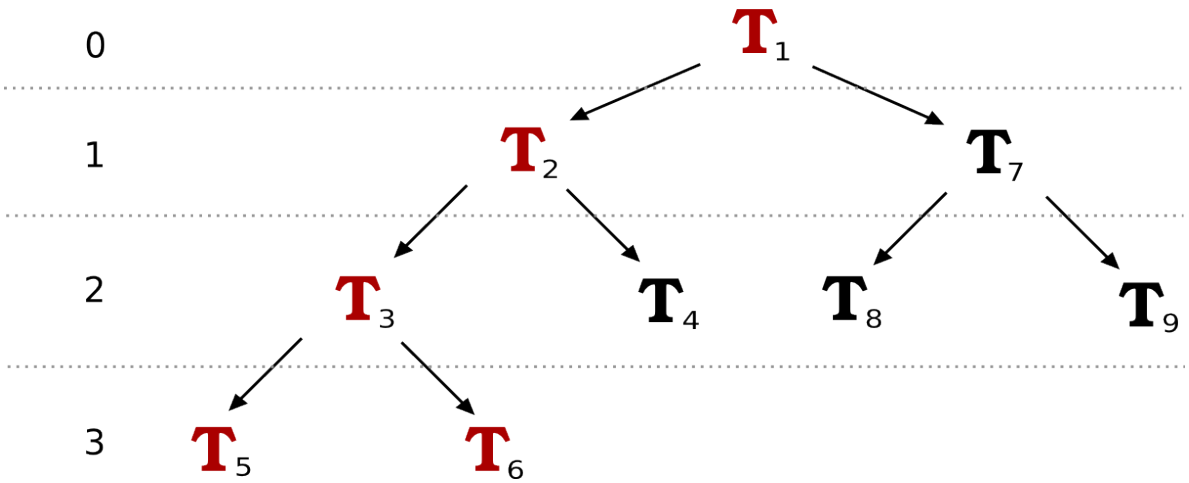


Abbildung 5.5: Tiefenfilterung eines Konversationsbaums; Tiefe am linken Rand

5.3.3 Filterung der Daten

Im Rahmen der Arbeit wurde eine weitere Filterung der Datenmenge vorgenommen, um diese auf möglichst interessante Konversationen einzugrenzen und bezüglich dieser zu analysieren. Dabei wurden die Daten unter anderem auf die Tiefe der enthaltenen Konversationsbäume untersucht.

In einzelnen Filterstufen wurden dabei Konversationsbäume, die eine gewisse Mindesttiefe nicht erreichten herausgefiltert. Abbildung 5.5 zeigt, wie eine Tiefenfilterung im Rahmen der Arbeit vorgenommen wurde. In diesem Beispiel umfasst eine Tiefenfilterung der Tiefe 3 alle rot markierten Tweets. Neben den beiden Tweets, die sich tatsächlich auf Tiefe 3 befinden, werden auch alle Tweets die zwischen diesem und dem Wurzel-Tweet liegen, ausgewählt. Wird der gefilterte Datensatz später geladen, müssen die fehlenden Tweets bis zur Wurzel nicht über die Twitter-APIs bezogen werden.

5.3.4 Analyse der gefilterten Daten

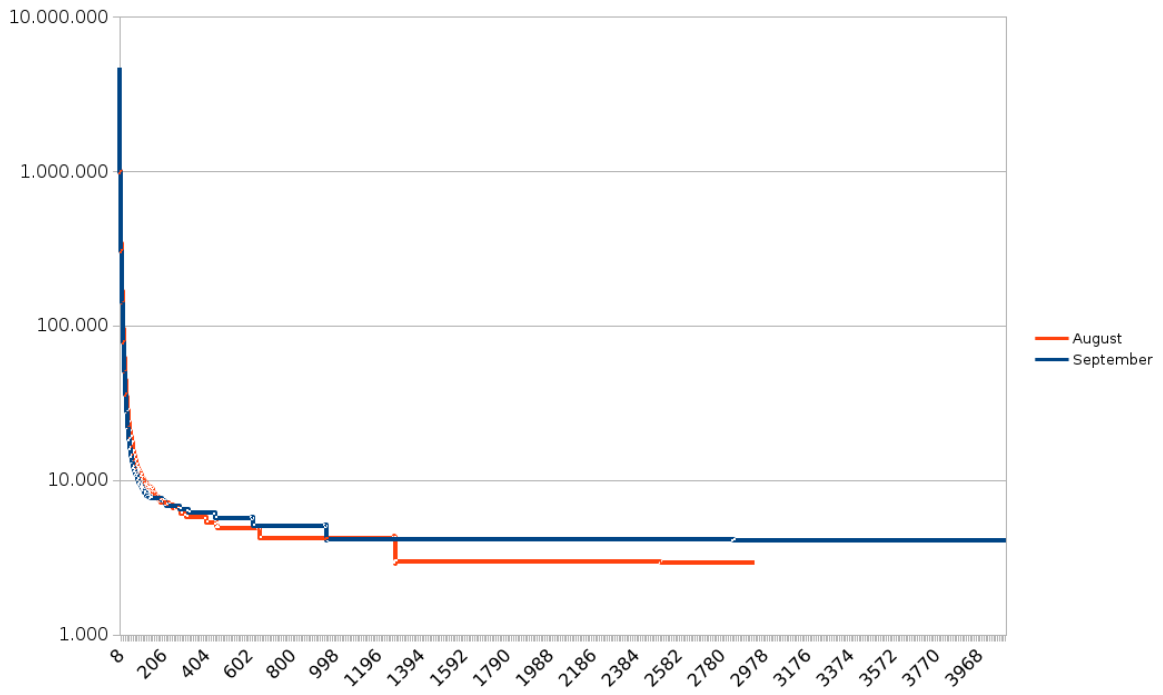


Abbildung 5.6: Ergebnisse der Tiefenfilterung

Abbildung 5.6 zeigt die Ergebnisse des Filtervorgangs. Dabei zeigt die Y-Achse die Anzahl der verbleibenden Tweets und die X-Achse die Tiefenstufe des Filters an. Mit zunehmender Tiefe sinkt die Anzahl der Tweets schnell ab und hält sich dann auf einem weitgehend konstanten Level, von dem sie direkt auf 0 absinkt. Dieses Verhalten wird durch eine kleine Anzahl von Benutzern hervorgerufen, die sogenannte Bot-Accounts betreiben. Bot-Accounts sind Twitter Accounts, die automatisiert bestimmt Inhalte twittern, retweeten oder replien und oftmals zum Spamversand oder Ähnlichem

verwendet werden. Dabei gibt es Bot-Accounts die sich selbst tausendfach replien und somit sehr lange Reply-Ketten, tiefe Konversationen, bilden. In den Daten von September 2014 befindet sich eine solche Reply-Kette, mit einer Tiefe von 4090 Tweets. Diese Kette ist in allen vorherigen Filterstufen mit geringerer Tiefe ebenfalls enthalten und fällt erst bei einer Filterung mit Tiefe 4.091 komplett heraus. Die längste Kette im August 2014 besitzt eine Tiefe von 2.929 Tweets.

Abzüglich der Replies-Ketten von Bot-Accounts sind die Mehrzahl der Konversationen auf Twitter sehr flach. Diese Beobachtung deckt sich mit existierenden Arbeiten zur Analyse von Twitterdaten. Kwak et al. [KLPM10] untersuchten die Tiefe von Retweet-Bäumen und die Anzahl der partizipierenden Benutzer. Eine Großzahl der Retweet-Bäume waren nicht tiefer als drei Retweets. Der Anteil an Bäumen der Tiefe 1, machte dabei 95.8%, der Anteil der Bäumen mit Tiefe kleiner 6 sogar 97.6% der Baume aus.

5.4 Verarbeitung der Twitterdaten

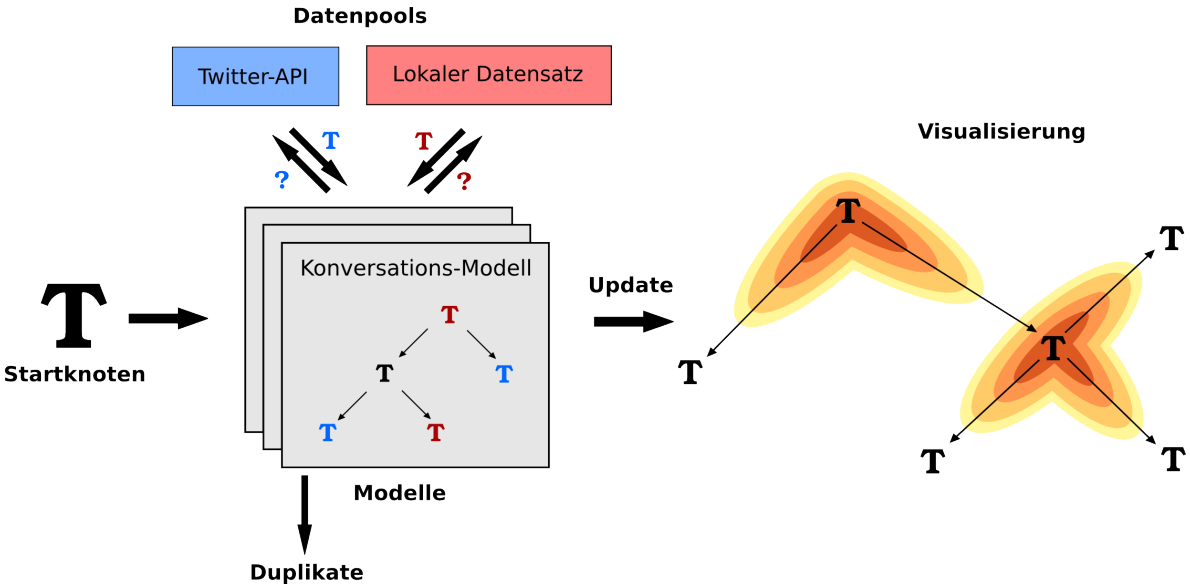


Abbildung 5.7: Ausgehend von einem Start-Tweet wird die Konversation mit Hilfe der Twitter-API und eines lokalen Datensatzes vervollständigt. Bei Änderungen des Modells wird die Visualisierung aktualisiert. Duplikate einer Konversation werden aussortiert.

Beim Start der Anwendung sind keine Konversationen in der Konversationsleiste vorhanden. Fügt der Benutzer, durch Angabe einer TweetID einen Start-Tweet und somit eine Konversation hinzu, wird diese zunächst, wie im Konzept in 4.3.1 beschrieben, zur Wurzel vervollständigt. Hierzu wird eine rekursive Methode aufgerufen, die prüft ob es sich beim jeweiligen Tweet um einen Reply bzw. einen Retweet handelt. Abhängig davon wird überprüft ob der Tweet einen Eltern-Tweet besitzt und dieser ermittelt. Dabei wird bei jedem neu hinzugefügten Tweet überprüft, ob dieser bereits in einer anderen Konversation vorhanden ist. Ist der Wurzel-Tweet erreicht und auch dieser noch nicht vorhanden, ist

die Konversation kein Duplikat und kann weiter vervollständigt werde. Wird bei der Vervollständigung zur Wurzel ein bereits vorhandener Tweet entdeckt, so ist die Konversation bereits vorhanden und wird verworfen. Anschließend beginnt die Vervollständigung des Konversationsbaums, indem, wie in Kapitel 4.3.2 beschrieben, rekursiv die Kindknoten bereits vorhandener Baumknoten gesucht werden. Bei der Abfrage der Tweets wird den offline gespeicherten Daten, aus erwähnten Gründen, Vorrang gegeben.

Jede Konversation wird dabei in einem eigenem Konversations-Modell gespeichert. Dieses beinhaltet neben einer HashMap, die aus TweetID-Tweet-Paaren besteht, auch zusätzliche Informationen zu dessen Aktivität, ID des Wurzelknotens und den Zeitintervall über welchen sich die Konversation erstreckt. Letzteres wird zu Bestimmung des Ausschnittes der Zeitleiste benötigt. Die Geo-Tweets einer Konversation werden zudem zusätzlich mit ihren Längen und Breitengrad in einem Quadtree, beschrieben in Kapitel 3.2.2, der die Fläche der Erde abdeckt, gespeichert. So kann beim Rendern der Daten effizient festgestellt werden welche Tweets im aktuellen Viewport liegen. Wird einer Konversation ein neuer Tweets hinzugefügt werden über Events die Overlays benachrichtigt, die für das Rendern des Konversationsbaumes zuständig sind. Abbildung 5.7 zeigt eine vereinfachte Darstellung des Prozesses, den die Twitterdaten in der Anwendung durchlaufen.

Das Graph-Overlay zeichnet die Knoten und Kanten des Geo-Konversationsbaums in Form eines Knoten-Kanten-Diagramms direkt auf die Karte. Das Splat-Overlay ist für die Darstellung der einzelnen Splats bzw. der Heatmap auf die Karte zuständig. Die genaue Prozess zum Berechnung und Darstellung der Splats wird in folgendem Abschnitt beschrieben.

5.5 Erzeugung der Heatmap

Wie bereits im Konzept erwähnt, sollen die einzelnen Splats der Heatmap zur Laufzeit, durch eine Kernelfunktion erzeugt werden. Um diese für evt. spätere Anwendungen beliebig austauschen zu können wurde ein Interface, Listing 5.2, für einen beliebigen Splat-Kernel implementiert.

```
1 public interface SplatKernel {
2
3       public double[][] calculateSplat(Point2D sourcePos, Point2D targetPos);
4 }
```

Listing 5.2: Interface für einen beliebigen Splatkernel

Ein Splat-Kernel muss dabei lediglich eine Funktion implementieren, die aus den bereits in Pixelkoordinate übergebenen Punkten das Pixelgitter der Splats errechnet und zurückgibt. Zur konkreten Berechnung der Splats wurde ein Gaußkernel analog zur Ausführung der Konzepte in Kapitel 4.5.3 implementiert. Dieser ruft für jede Gitterzelle des Splat-Gitters die in Listing 5.3 gezeigte Gaußfunktion auf. Somit werden Zelle für Zelle die entsprechenden Werte des Splat-Gitters berechnet.

Anschließend wird der Splat auf dem Pixelgitter auf der Position des Elternknotens platziert und in Richtung des Kindknotens gedreht. Die Werte des Splats werden dabei auf die Werten bereits übertragenen Splats aufsummiert. Die Werte des Pixelgitters werden anschließend, wie in Kapitel 4.5.5 beschrieben, auf eine Farbskala übertragen. Die zugrundeliegende Farbpalette wird mit Prefuse [Teaa], einem in Java geschriebenen Visualisierungs-Toolkit, generiert und verläuft, mit zunehmenden Werten, linear von weiß über gelb und rot nach schwarz. Die eingefärbten Splats werden anschließend von Splat-Overlay auf dem Bildschirm angezeigt.

```
1  private double gauss(int x, int y, double bandwidth, double stretchFactor){
2
3        if(x > 0){
4              return normalization*
5                    Math.exp(-0.5*(
6                    Math.pow(x, 2)/(Math.pow(bandwidth, 2)*stretchFactor)+
7                    Math.pow(y, 2)/Math.pow(bandwidth, 2)
8                    ));
9        }else{
10             return normalization *
11                   Math.exp(-0.5*(
12                   Math.pow(x, 2)/Math.pow(bandwidth, 2) +
13                   Math.pow(y, 2)/Math.pow(bandwidth, 2)
14                   ));
15       }
16 }
```

Listing 5.3: Umsetzung des Gaußkernels

6 Anwendungsfall

In diesem Kapitel wird die entstandene Anwendung genutzt, um einen zuvor gefilterten Datensatz zu analysieren. Dabei wird zunächst der gefilterte Datensatz beschrieben. Anschließend werden die einzeln, während der Analyse, durchgeführten Schritte erläutert und die resultierenden Visualisierungen anhand von Abbildungen erklärt und interpretiert. Zuletzt werden die durchgeführten Schritte nochmals zusammengefasst und die Ergebnisse der Fallstudie erläutert.

6.1 Daten und Ziel der Anwendung

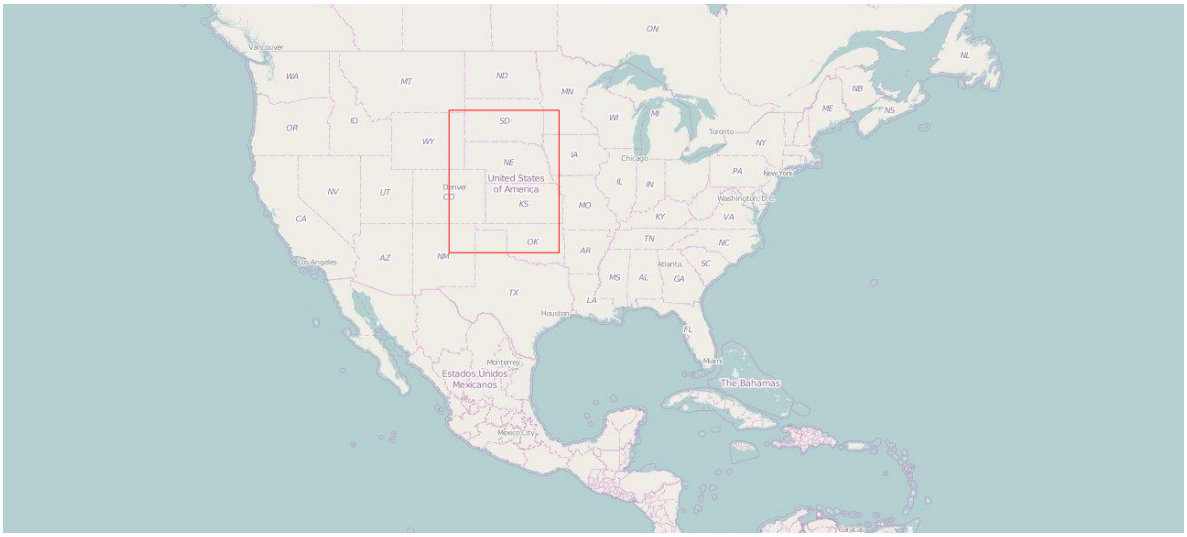


Abbildung 6.1: Für geografische Filterung betrachteter Ausschnitt

Wie bereits erwähnt, standen während dieser Arbeit Twitterdaten, Replies mit geografischer Lokation, zur Verfügung, welche nach der Tiefe vorkommender Konversationsbäume gefiltert wurden. Aus den gefilterten Twitterdaten von September wurde der Datensatz der Tiefe 8 entnommen. Dieser Datensatz beinhaltet noch 389.270 Replies bei relativ hoher Tiefe der Konversationsbäume. Die Replies des Datensatzes wurde anschließend zusätzlich auf deren geografische Lokation gefiltert, sodass ausschließlich Replies erhalten blieben, die im Zentrum Nordamerikas liegen. Abbildung 6.1 zeigt den Ausschnitt, aus dem Replies entnommen wurden. Als zugrundeliegender Datenpool, aus dem die Tweets zu Vervollständigung der Konversationen geladen werden, werden im folgenden Anwendungsbeispiel die Reply-Daten der Tiefe 8 verwendet. Die geografisch gefilterten Daten werden zum automatischen Einlesen als Startknoten verwendet.

In diesem Anwendungsfall werden die gefilterten Twitterdaten, welche Replies aus dem oben genannten Ausschnitt enthalten, zunächst ihren Konversationen zugeordnet und mit Hilfe der bereitgestellten Twitterdaten der Tiefe 8 vervollständigt. Anschließend sollen die Konversationen auf deren geografische Verbreitung untersucht werden. Ziel ist es festzustellen, ob Konversationen mit Tweets aus dem Zentrum Nordamerikas einen generellen Trend in ihren Verbreitungswegen ausweisen.

6.2 Durchführung

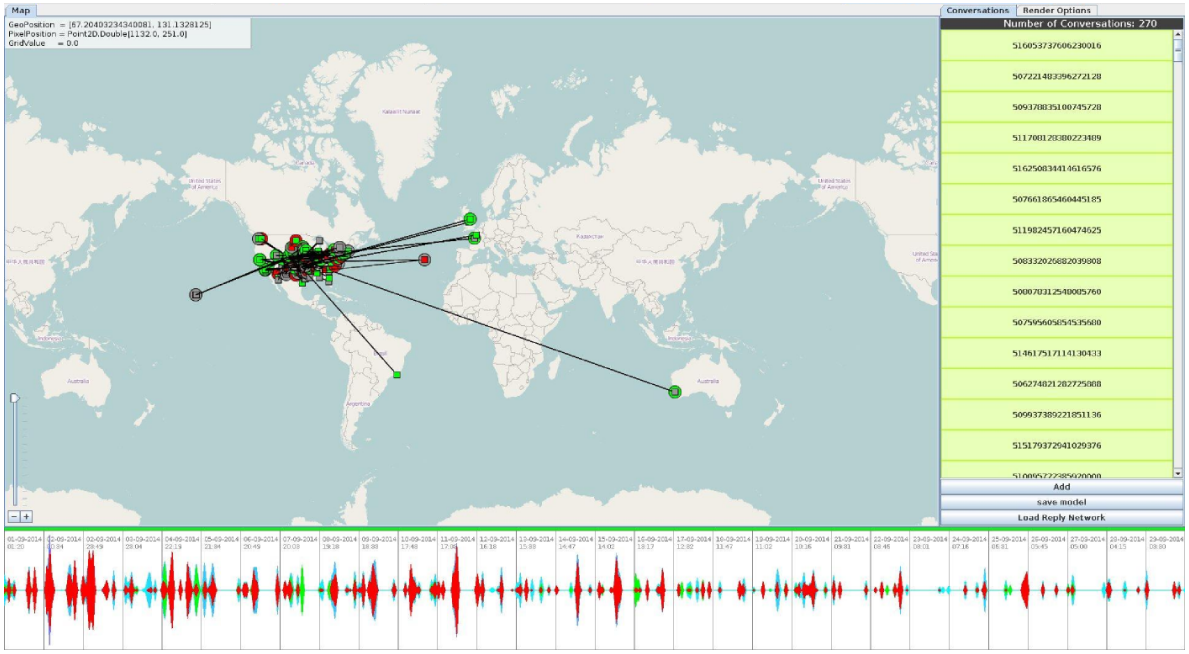


Abbildung 6.2: Benutzeroberfläche der Applikation mit geladenen Konversationen

Nachdem die Anwendung gestartet wurde, braucht diese zunächst einen Moment, um den zugrundeliegenden offline Datenpool, ca. 94,4 MB, zu laden. Anschließend kann über die Schaltfläche „Load Start Nodes from CSV“ der geografisch gefilterte Datensatz ausgewählt werden.

Alle im Datensatz enthaltenen Replies werden anschließen als Startknoten einer Konversation geladen und die Anwendung versucht den Konversationsbaum aus dem zugrundeliegenden Datenpool aufzubauen. Da nicht alle als Startknoten eingefügte Replies unterschiedlichen Konversationen angehören, werden die doppelt vorhandenen Konversationen bei deren Detektion gelöscht. Sind alle Daten verarbeitet bleiben 270 unterschiedliche Konversationen übrig. Diese werden in der Konversationsleiste angezeigt. In der Zeitleiste lässt sich deren Verteilung über den Monat, September 2014, betrachten. Abbildung 6.2 zeigt die Benutzeroberfläche mit den geladenen Konversationen.

Das Knoten-Kanten-Diagramm überlagert sich sehr stark, sodass im Zentrum Nordamerikas lediglich eine große Ansammlung von Knoten erkennbar ist. Nur wenn man sich weiter vom Zentrum

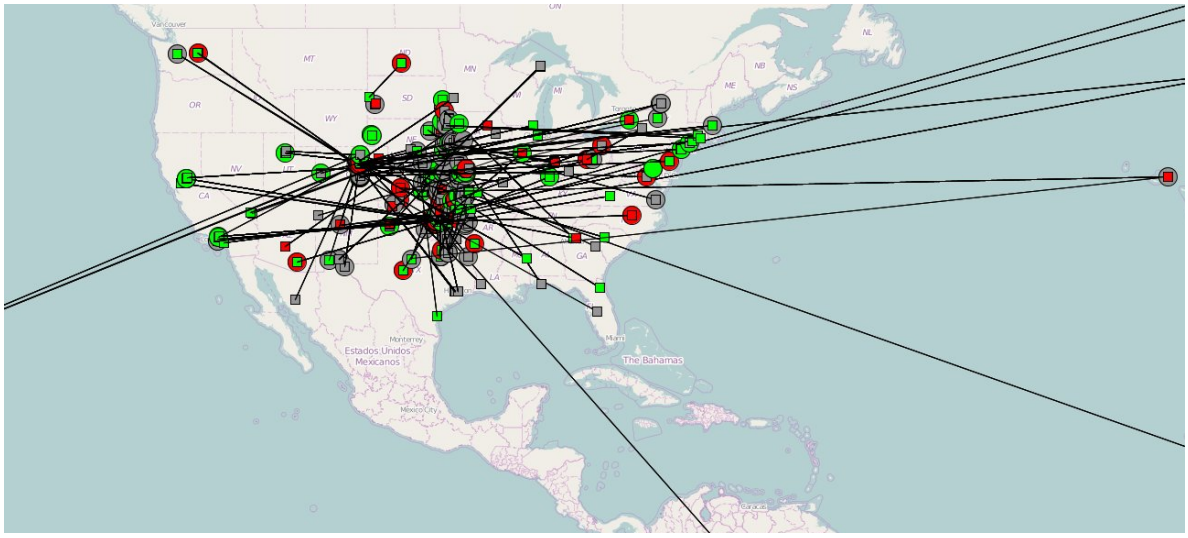
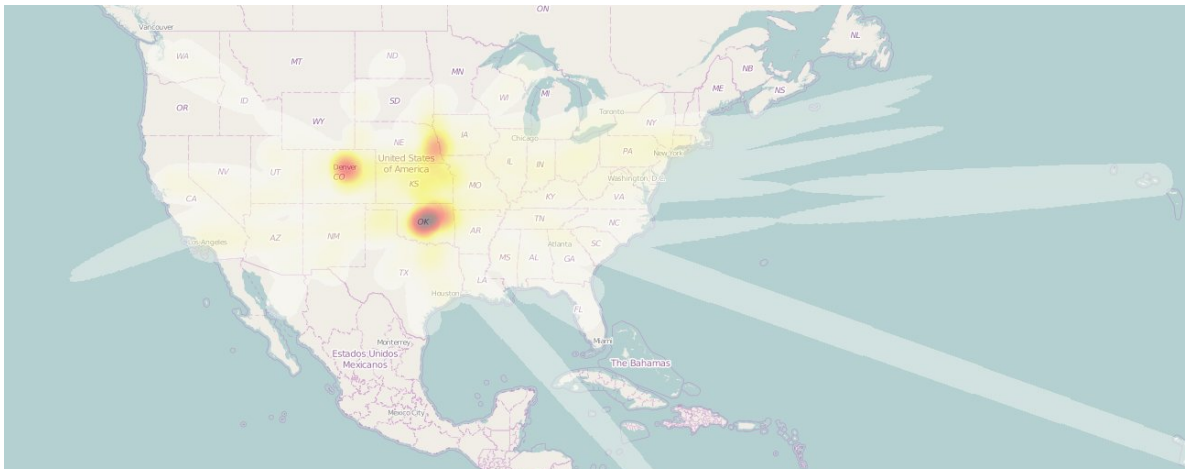


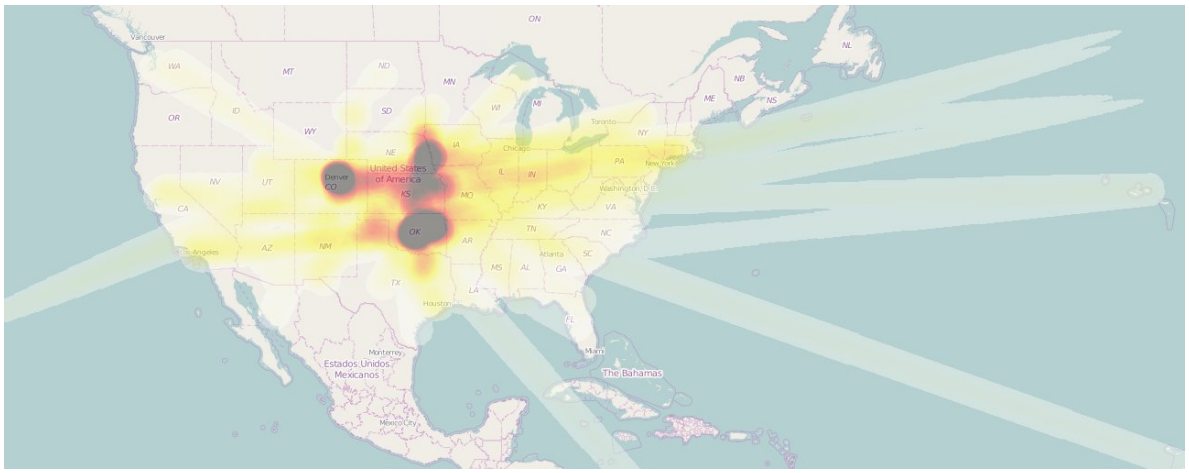
Abbildung 6.3: Knoten-Kanten-Diagramm aus Konversationen des geladenen Datensatzes

entfernt lassen sich einzelne Kanten, z.b nach Europa oder Australien, verfolgen. Abbildung 6.3 zeigt das resultierende Knoten-Kanten-Diagramm auf der Karte. Um einen besseren Überblick zu erlangen, wird daher zur Heatmap-Darstellung gewechselt.

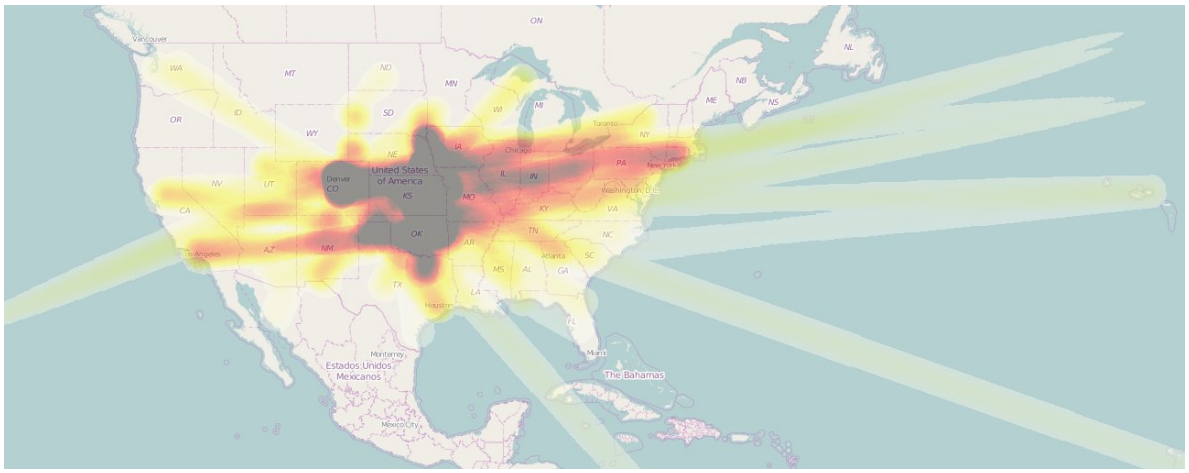
Auf der Heatmap, Abbildung 6.4a, kann man drei dicht beieinander liegende rote Bereiche im Zentrum Nordamerikas entdecken. An diesen Bereichen ist das Aufkommen von weiterverbreiteten Replies besonders hoch. Eine generelle Verbreitungsrichtung ist noch nicht zu erkennen. Lediglich der unterste rote Bereich weist eine leichte Tendenz nach Osten auf. Durch das sehr hohe Aufkommen an den drei sichtbaren Bereichen werden die Bereiche mit geringeren Aufkommen sehr stark abgewertet, also weiß oder maximal hellgelb dargestellt. Um weitere Einsichten zu erlangen, wird daher der maximale Wert, der für die Farbskalierung verwendet wird, reduziert. Die Farbwerte werden also neu skaliert, sodass weniger hohe Werte aufgewertet werden. Abbildung 6.4b zeigt die Heatmap mit neuer Farbskalierung. Der maximale Wert wurde dabei auf 25% reduziert. Außerdem wurden die Richtungen der Splats stärker gewichtet. Hierzu wurde der Streckfaktor der Splats angepasst, sodass diese 75% der Strecke bis zum Kindknoten überbrücken. Nun sind neben den drei roten Bereichen auch Verbreitungsrichtungen sichtbar. Man erkennt, dass zwischen den roten Bereichen sehr stark kommuniziert wird. Außerdem kann in gelb und orange eine schwächere Verbreitung ausgehend von Zentrum in Richtung der Ostküste festgestellt werden. Um diesen Effekt genauer beobachten zu können, wird erneut der Maximalwert der Farbskalierung auf 10% verringert. In Abbildung 6.4c, können die Verbreitungsrichtungen nun noch deutlicher erkannt werden. Während die Verbreitung der meisten Replies innerhalb des Zentrums stattfindet, lässt sich eine weitere, wenn auch schwächere, Tendenz in Richtung der Ostküste entdecken.



(a) Resultierende Heatmap: Splat-Reichweite beträgt 50%, Farbskalierung mit 100% des Maximalwerts



(b) Resultierende Heatmap: Splat-Reichweite beträgt 75%, Farbskalierung mit 25% des Maximalwerts



(c) Resultierende Heatmap: Splat-Reichweite beträgt 75%, Farbskalierung mit 10% des Maximalwerts

Abbildung 6.4: Resultierende Heatmaps

6.3 Ergebnis

Unter Verwendung des implementierten Prototyps wurde die geografische Verbreitung von Reply-Konversationen, mit einer Mindesttiefe von 8, und mindestens einem Knoten aus Zentral-Nordamerika untersucht. Dabei wurde festgestellt, dass der Großteil der Konversationen Zentral-Nordamerika entspringen und dieses nicht verlassen. Das Verbreitungsgebiet der Konversationen beschränkt sich hier also auf das nahe Umfeld ihres Ursprungs. Durch eine feinere Abstimmung der Farbskalierung und der Größe der Splats, konnten schwächere Verbreitungsmuster sichtbar gemacht werden. Neben der Ausbreitung nahe des Ursprung verbreiten sich Konversationen im Beispiel stärker zur Ostküste. Dieser Trend fällt im Vergleich zum enormen Interkommunikation im Zentrum jedoch sehr schwach aus und konnte nur durch eine extreme Abwertung des maximalen Wertes auf 10% erreicht werden. Eine sich klar abzeichnende Verbreitungsrichtung existiert in diesem Datensatz daher nicht.

7 Zusammenfassung und Ausblick

Im Verlauf dieser Arbeit wurden ausgehend von einem Workshop mit Domäneexperten, Konzepte entwickelt, um Informationen zur Informationsdiffusion, innerhalb des sozialen Netzwerks Twitter, für den Zivil- und Katastrophenschutz zugänglich zu machen.

Während des Workshops zeigte sich, dass ein großes Anliegen der Experten die Verfolgung selbst herausgegebener Nachrichten ist, um so auf deren Akzeptanz zu schließen und ein Bild über die Informationslage zu erlangen. Ein weiterer Schwerpunkt war das gezielte Ausmachen von Empfängern und Urhebern von Informationen, um Kontakt mit freiwilligen Helfern aufzunehmen oder dem Ursprung von Falschmeldungen nachzugehen. Einige Experten äußerten zudem Bedenken bei der Verwendung eines Knoten-Kanten-Diagramms zur Darstellung der Informationsverbreitung, aufgrund dessen Unübersichtlichkeit bei großen und dichten Graphen.

Im Folgenden wurden die verfolgten Konzepte und deren Umsetzung beschrieben. Zunächst wurde das Beziehen von Kommunikationsnetzwerken und dessen Strukturierung in Konversationen erläutert. Die aus den Konversationen aufgebauten Konversationsbäume beinhalten zu Beginn Knoten mit und ohne geografischer Lokation. Daher wurden aus den Konversationsbäumen Teilbäume extrahiert, die ausschließlich aus den Knoten mit geografischer Lokation bestehen. Die resultierenden Geo-Konversationsbäume können anschließend als Grundlage zur Visualisierungen auf der Karte dienen. Zum Einen wird die Visualisierung durch ein Knoten-Kanten-Diagramm realisiert, dessen Knoten die Tweets der Konversation repräsentieren und an der entsprechenden geografischen Lokation auf der Karte verortet sind. Ob ein Knoten einen Reply oder Retweet darstellt und welche Stimmung die einzelnen Tweets widerspiegeln, wird durch die Form und Farbe der Knoten kodiert. Zum Anderen wird eine weitere Art der Visualisierung in Form einer Heatmap unterstützt. Diese besteht aus gerichteten Splats, welche ausgehend von den Elternknoten des Geo-Konversationsbaumes in Richtung der Kindknoten gerichtet sind. Die Splats werden von einer zugrundeliegende Kernelfunktion berechnet, so dass deren Parameter zur Laufzeit angepasst werden können. Im Zuge der Arbeit wurde hierzu ein Gaußkernel verwendet. Die Kombination der generierten Splats ergibt anschließend die Heatmap. Die Farbskalierung der Heatmap, sowie die Parameter des Spalts, zur können Laufzeit über das Benutzerinterface manipuliert werden.

Abschließend wurde ein Anwendungsfall vorgestellt, bei welchem die Verbreitung von Konversationen, die Replies aus Nordamerika enthalten, untersucht wurde. Dabei wurde festgestellt, dass sich ein Großteil der Kommunikation innerhalb des ausgewählten Ausschnitts Nordamerikas abspielt. Unter Verwendung der Heatmap konnten nach Justierung der Splat- und Farbskalierungsparameter auch schwächere Trends erkannt werden. So wurde eine schwache Ausbreitung in Richtung der Ostküste festgestellt.

7.1 Diskusionn

Die während der Arbeit entwickelten Konzepte zur Visualisierung erwiesen sich im vorgestellten Anwendungsfall als hilfreich, um Verbreitungswege von Nachrichten übersichtlich auf einer Karte zu visualisieren. Im Anwendungsbeispiel konnten selbst schwache Verbreitungsrichtungen durch das Variieren der Farbskalierung und Splat-Reichweite sichtbar gemacht werden. Auch zur besseren Verfolgung der dünnen und oftmals langen Kanten des Knoten-Kanten-Diagramms scheinen sich die Splats der Heatmap gut zur eignen. Probleme ergaben sich beim Erkennen der Endpunkte der Splats, bei ausgeblendetem Knoten-Kanten-Diagramm. Zwar weiß der Benutzer wie viel Prozent der Strecke zwischen Anfangs und Endpunkt ein Splat zurücklegt, ein intuitives und schnelles schließen auf den daraus resultierenden Endpunkt scheint jedoch nicht möglich. Auch bei der Aggregation vieler Splats kann dies zu Problemen führen. Betrachtet man die Heatmap des Anwendungsfalls in Abbildung 6.4c könnte ein Benutzer fälschlicherweise annehmen, die Nachrichten würden sich nach Europa verbreiten. Macht man sich jedoch bewusst, dass die Länge des Splats im Beispiel auf 75% eingestellt wurde, wird klar, dass sich die Verbreitungsrichtung auf die Ostküste bezieht. Durch gleichzeitiges Einblenden des Knoten-Kanten-Diagramms kann solchen Missverständnissen vorgebeugt werden, vorausgesetzt, dass dieses an den entsprechenden Stellen gut lesbar ist. Um in Knoten-Kanten-Diagrammen mit viel Visual Clutter einen übersichtlichen Graph mit klar erkennbaren Anfangs- und Endpunkten anzubieten, könnte zusätzlich z.B. Edge Bundling verwendet werden.

Im Datensatz der während der Arbeit zur Verfügung stand, konnten jedoch keine größeren Menge an Konversationen gefunden werden, die deutlich ausgeprägten Trends folgen. Allerdings beinhalteten die zur Verfügung stehenden Daten ausschließlich Replies. Der Bezug von zusätzlichen Retweets für größere Mengen an Konversationen ließ sich auf Grunde der strengen Limitierungen, denen die Twitter Rest-API unterworfen ist, nicht realisieren. In wie weit sich diese gewonnenen Erkenntnisse auf Retweets einer Konversation übertragen lassen, bleibt daher unklar.

Einzelne, aus Replies bestehende Konversationen des Datensatzen wiesen hingegen starke Gemeinsamkeiten auf. So scheinen die meisten aus Replies bestehenden Konversationen zwischen zwei verschiedenen Benutzern stattzufinden. Selbst bei verhältnismäßig tiefen Konversationen nimmt die Anzahl der Nutzer kaum zu. Die beobachteten Benutzer hielten sich zum Großteil in Städten auf und bewegen sich meist gar nicht oder nur gering, sodass die geografische Lokation der Tweets nur schwach variiert. Ein typisches Bild einer solchen Konversation, in der zwei Benutzer teilnehmen, besteht daher aus zwei sichtbaren aufeinander zeigenden Splats, die wiederum durch Überlagerung mehrerer gleichartiger Splats entstehen. Abbildung 5.1 zeigt eine solche Konversation. Eine naheliegende Vermutung ist, dass sich Benutzer, die in langen Reply-Ketten kommunizieren, kennen und Replies für Gespräche untereinander nutzen.

Ein geeigneter Datensatz mit Tweets einer realen Krisensituation stand im Rahmen der Arbeit leider nicht zur Verfügung und konnte auf Grund der erwähnten Limitierungen auch nicht direkt von Twitter bezogen werden. Die Untersuchung der Verbreitungswege solcher Nachrichten verbleibt daher für weiterführende Arbeiten. Aufgrund genannter Studien, welche ein verstärktes Aufkommen von Tweets in Krisenlagen feststellten, lässt sich jedoch vermuten, dass sich die Diffusion solcher Nachrichten großteils auf das nähere Umfeld des Krisengebietes bezieht.

7.2 Weiterführende Arbeiten

Aus den Erkenntnissen dieser Arbeiten haben sich viele weitere interessante Fragestellungen ergeben. Diese bieten Ansatzpunkte für künftige, weiterführende Arbeiten.

Die Abfrage der Konversationen von Twitter wurde aufgrund verschiedener Überlegungen über die Rest-Api realisiert. Ohne der Verwendung eines Offline-Datensatzes werden jedoch schnell die vorgegebenen Obergrenzen zur Abfrage von Tweets erreicht. Die Entwicklung besserer Abfragestrategien, welche sowohl die Rest- als auch die Streaming-API verwenden, könnten diese Probleme möglicherweise weitgehend auflösen. Vor allem die von Twitter neu eingeführten Page-Streams stellen zukünftig eine interessante Alternative dar, um nutzerbezogene Tweets zu empfangen.

Auch der in Kapitel 4.3 beschriebene Aufbau der Kommunikationsbäume kann effizienter gestaltet werden. Im Moment werden doppelte Konversationen gelöscht. Diese könnten jedoch mit ihrem Gegenstück zusammengeführt werden. Somit könnte der Baum schneller aufgebaut und doppelte Anfragen reduziert werden.

Einen weiteren Ansatzpunkt stellt die Verortung zusätzlicher Tweets auf der Karte dar. Da nur ein Bruchteil der täglich abgesetzten Tweets mit Angaben zur Lokation gekennzeichnet sind, können diese oft nicht repräsentativ betrachtet werden. Es existieren jedoch Techniken um Tweets anhand ihrer Inhalte und Benutzerinformationen bestimmten Regionen zuzuordnen. Diese Techniken könnten verwendet werden, um eine größere Anzahl an Tweets auf der Karte zu verorten und somit die Aussagekraft der kartenbasierten Analyse zu verbessern. Mittels dieser Methoden ermittelte Lokationen stellen jedoch bestenfalls eine Annäherung an die tatsächliche geografische Lokation dar und bergen daher gewisse Unsicherheiten.

Auch die Visualisierung der Heatmap kann weiter optimiert werden. Das Rendern der Heatmap kann für viele Konversationen bei aktueller Umsetzung sehr langsam werden. Dies liegt vor allem daran, dass die Splats pixelgenau und je nach Zoomstufe und Größe in sehr hoher Auflösung berechnet und dargestellt werden. Dieses Verfahren könnte beispielsweise durch die Berechnung von weniger Pixeln, zwischen denen die Farben interpoliert werden, beschleunigt werden.

Zudem sind Werkzeuge für tiefergehende Analysen und zur detaillierten Untersuchung einzelner Tweets wünschenswert. So ist eine Filterung bezüglich der Zeit und der geografischen Lokation wünschenswert. Eine weitere Idee ist das Einführen eines Analysewerkzeugs, welches es ermöglicht, einer Konversation auf der Karte schrittweise über ausgewählte Zeitintervalle, ähnlich eines Debuggers, zu verfolgen.

Literaturverzeichnis

[BTH+13] H. Bosch, D. Thom, F. Heimerl, E. Puttmann, S. Koch, R. Kruger, M. Worner, T. Ertl. Scatterblogs2: Real-time monitoring of microblog messages through user-guided filtering. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2022–2031, 2013. (Zitiert auf den Seiten 11 und 12)

[BTW+11] H. Bosch, D. Thom, M. Worner, S. Koch, E. Puttmann, D. Jackle, T. Ertl. Scatterblogs: Geo-spatial document analysis. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, S. 309–310. IEEE, 2011. (Zitiert auf Seite 12)

[BVKW12] M. Burch, C. Vehlow, N. Konevtsova, D. Weiskopf. Evaluating partially drawn links for directed graph edges. In *Graph Drawing*, S. 226–237. Springer, 2012. (Zitiert auf Seite 32)

[CLS+12] N. Cao, Y.-R. Lin, X. Sun, D. Lazer, S. Liu, H. Qu. Whisper: Tracing the spatiotemporal process of information diffusion in real time. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2649–2658, 2012. (Zitiert auf den Seiten 13 und 14)

[Dif] Diffen. ReTweet vs. Reply. URL http://www.diffen.com/difference/ReTweet_vs_Reply. (Zitiert auf den Seiten 24 und 25)

[Fou] O. Foundation. OpenStreetMaps. URL http://www.openstreetmap.org. (Zitiert auf Seite 44)

[Hol06] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):741–748, 2006. (Zitiert auf Seite 31)

[HVW09] D. Holten, J. J. Van Wijk. Force-Directed Edge Bundling for Graph Visualization. In *Computer Graphics Forum*, Band 28, S. 983–990. Wiley Online Library, 2009. (Zitiert auf Seite 31)

[KH10] A. M. Kaplan, M. Haenlein. Users of the world, unite! The challenges and opportunities of Social Media. *Business horizons*, 53(1):59–68, 2010. (Zitiert auf Seite 9)

[KKEM10] D. Keim, J. Kohlhammer, G. Ellis, F. Mansmann. *Mastering the Information Age - Solving Problems with Visual Analytics*. Eurographics Association, 2010. URL http://books.google.de/books?id=vdv5wZM8ioIC. (Zitiert auf den Seiten 17 und 18)

[KLPM10] H. Kwak, C. Lee, H. Park, S. Moon. What is Twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, S. 591–600. ACM, 2010. (Zitiert auf Seite 50)

[MRH+10] R. Maciejewski, S. Rudolph, R. Hafen, A. Abusalah, M. Yakout, M. Ouzzani, W. S. Cleveland, S. J. Grannis, D. S. Ebert. A visual analytics approach to understanding spatiotemporal hotspots. *Visualization and Computer Graphics, IEEE Transactions on*, 16(2):205–220, 2010. (Zitiert auf Seite 15)

[QHZZ11] Y. Qu, C. Huang, P. Zhang, J. Zhang. Microblogging after a major disaster in China: a case study of the 2010 Yushu earthquake. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, S. 25–34. ACM, 2011. (Zitiert auf den Seiten 9, 11 und 12)

[Ste] M. Steiger. JXMapViewer2. URL https://github.com/msteiger/jxmapviewer2. (Zitiert auf Seite 44)

[TC05] J. Thomas, K. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society Press, 2005. URL http://books.google.de/books?id=DybZPAAACAAJ. (Zitiert auf Seite 17)

[Teaa] P. Team. Prefuse. URL http://prefuse.org/. (Zitiert auf Seite 52)

[Teab] T. Team. Twitter4J. URL http://twitter4j.org/. (Zitiert auf Seite 46)

[Twia] Twitter. FAQs zum Hinzufuegen Deines Standorts zu Deinen Tweets. URL https://support.twitter.com/articles/484789-faqs-zum-hinzufugen-deines-standorts-zu-deinen-tweets. (Zitiert auf Seite 25)

[Twib] Twitter. Twitter Company. URL https://about.twitter.com/de/company. (Zitiert auf Seite 23)

[Twic] Twitter. Twitter Documentation. URL https://dev.twitter.com/overview/documentation. (Zitiert auf den Seiten 25, 26 und 27)

[Twid] Twitter. Was sind Antworten und Erwaehnungen. URL https://support.twitter.com/articles/85468-was-sind-antworten-und-erwahnungen. (Zitiert auf Seite 25)

[VHSP10] S. Vieweg, A. L. Hughes, K. Starbird, L. Palen. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, S. 1079–1088. ACM, 2010. (Zitiert auf Seite 11)

[Wika] P. Wiki. Definition: Tree (Graph Theory). URL https://www.proofwiki.org/wiki/Definition:Tree_(Graph_Theory). (Zitiert auf Seite 20)

[Wikb] Wikipedia. Alpha Blending. URL http://de.wikipedia.org/wiki/Alpha_Blending. (Zitiert auf Seite 38)

[Wikc] Wikipedia. Heat map. URL http://en.wikipedia.org/wiki/Heat_map. (Zitiert auf Seite 22)

[WLY+14] Y. Wu, S. Liu, K. Yan, M. Liu, F. Wu. OpinionFlow: Visual analysis of opinion diffusion on social media. 2014. (Zitiert auf den Seiten 14, 31 und 39)

[YLC+12] J. Yin, A. Lampert, M. Cameron, B. Robinson, R. Power. Using social media to enhance emergency situation awareness. *IEEE Intelligent Systems*, 27(6):52–59, 2012. (Zitiert auf Seite 11)

[ZCW+14] J. Zhao, N. Cao, Z. Wen, Y. Song, Y.-R. Lin, C. Collins. FluxFlow: Visual Analysis of Anomalous Information Spreading on Social Media. *Visualization and Computer Graphics, IEEE Transactions on*, PP(99):1–1, 2014. doi:10.1109/TVCG.2014.2346922. (Zitiert auf Seite 12)

Alle URLs wurden zuletzt am 23. 01. 2015 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift