Institut für Visualisierung und Interaktive Systeme
Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Diplomarbeit Nr. 3313

Visuelle Analyse von Social Media Bewegungsdaten mittels kontextbasierter Annotation

Dominik Jäckle

Studiengang:	Softwaretechnik
Prüfer:	Prof. Dr. Thomas Ertl
Betreuer:	M. Sc. Harald Bosch Dipl.-Inf. Dennis Thom
begonnen am:	4. April 2012
beendet am:	4. Oktober 2012
CR-Klassifikation:	H2.8, H3.1, H3.3, H5.2

Kurzbeschreibung

Seit die sozialen Medien im Zeitalter des Web 2.0 ihren Durchbruch feierten, steigt die Anzahl der Benutzer enorm. Die Benutzergruppen reichen heutzutage vom zehnjährigen Kind über den ganz normalen Bürger bis hin zum Prominenten. Das Ziel ist dabei fast immer das gleiche: die restliche Welt über persönliche oder manchmal auch geschäftliche Empfindungen, Beobachtungen und Ereignisse aufzuklären. Mit der Verwendung standortbezogener Dienste sind immer mehr Menschen bereit, mit ihren Veröffentlichungen auch ihre Position zu teilen. Meist sind diese Daten über diensteigene Schnittstellen öffentlich zugänglich. Durch Sammeln und Analysieren der Daten kann beispielsweise dem Analyst die Möglichkeit eingeräumt werden, sich ein Bild der Bewegungsprofile, die in Verbindung mit einem Ereignis stehen, zu machen.

In dieser Diplomarbeit wird untersucht, ob sich innerhalb dieser Daten Bewegungsmuster mittels interaktiver Visualisierungen erkennen und durch visuelle Annotation, basierend auf Kontextinformation, beurteilen und erklären lassen. Da die Anzahl der Nutzer stetig steigt und somit auch die Datenmenge, liegt ein besonderer Fokus auf der Entwicklung einer Datenstruktur zur Repräsentation und effizienten Aggregation der Bewegungsdaten für lokale sowie globale Anwendungen. Diese Datenstruktur, welche unter anderem auch aggregierte, verknüpfte textuelle Informationen aus den veröffentlichten Nachrichten enthält, wird mit Hilfe von interaktiven Visualisierungskonzepten dem Analysten zur Exploration zur Verfügung gestellt. Es wird gezeigt, dass es unter Verwendung der genannten Konzepte dem Analysten möglich ist, aus Bewegungsdaten Informationen zu extrahieren, die Beurteilungen und Erklärungen von Bewegungsmustern zulassen.

Abstract

Since the social media services have celebrated their breakthrough, the amount of users is increasing drastically. The user groups range from kids to ordinary citizens up to celebrities, but the goal is almost always the same: to educate the rest of the world about personal or even business sentiments, observations and events. With the use of *Location-based Services* (LBS), more and more people are willing to share their publications and their position. Usually these data are publicly available through the interfaces of the services. For example, the analyst can get an idea of movement profiles which are related to an event, by collecting and analyzing the data.

This thesis investigates if movement patterns can be indentfied within this data while using interactive visualizations and also if it is possible to assess and explain these patterns by means of visual annotation, based on context information. As the number of users, and therefore also the amount of data increases constantly, a particular focus is set on the development of a data structure for a local and also global application which is able to represent and to aggregate the movement data in an efficient way. The data structure also contains associated aggregated textual information and will be presented to the analyst for exploration with the aid of interactive visualization concepts. It is shown that by using the named concepts, it is possible to extract information out of the movement data to assess and explain movement patterns.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Verzeichnis der Listings

Verzeichnis der Algorithmen

1 Einleitung

Mit dem Siegeszug der sozialen Medien veröffentlichen immer mehr Menschen ihre Position, Erlebnisse, Empfindungen und Beobachtungen, für jeden frei zugänglich, wie es zum Beispiel bei Twitter der Fall ist. Die Veröffentlichung der Position wird durch die Verwendung von GPS-Koordinaten realisiert, mit deren Hilfe sich Bewegungsprofile erstellen lassen. Für die Simulation und Vorhersage in unterschiedlichen Anwendungsgebieten, spielen die Bewegungsdaten dabei eine wichtige Grundlage.

Ein Beispiel aus diesem Bereich ist der Versuch der Vorhersage für Kursentwicklungen des Dow Jones. Anhand eines großen Datenvolumens haben Forscher in [BMZ10] eine Möglichkeit gefunden, die Entwicklung des Dow Jones anhand von Stimmungen, die Benutzer in ihren Tweets veröffentlichen, vorherzusagen. Die Genauigkeit für die kurzfristige Vorhersage der Entwicklung lag bei hohen 86,7 Prozent.

Ein weiteres Beispiel aus diesem Bereich ist das Projekt *Where's George*[1], welches mitunter auch Ideengeber für diese Diplomarbeit ist. Mit Hilfe einer interessierten Community wird die Bewegung von Dollar-Noten in den USA verfolgt. Auf dieser Basis wird versucht, Rückschlüsse auf das Reise- und Bewegungsverhalten von Menschen zu ziehen.

Man kann anhand beider Beispiele deutlich erkennen, welches Potential darin liegt. Diese Diplomarbeit widmet sich der Aufgabenstellung, Bewegungsdaten, die über Twitter gesammelt werden, mit Hilfe interaktiver Visualisierungen zu analysieren. In diesem Rahmen wird untersucht, ob sich Bewegungsmuster erkennen und durch visuelle Annotation, welche auf Kontextinformationen basiert, beurteilen und erklären lassen. Um dies umsetzen zu können, muss im ersten Schritt die Datenmenge drastisch eingeschränkt werden, weil nicht alle Daten fehlerfrei sind; oft sind verwendete Positionsangaben ungenau oder angegebene Uhrzeiten fehlerhaft. Da mit dem Durchbruch des Smartphones und anderer Technologien die Datenmenge ins Unermessliche steigt, bedarf es innovativer und neuer Methoden zur Aggregation und visuellen Analyse, um dem Analyst auf intuitive Art und Weise ein tieferes Verständnis für die Daten zu ermöglichen. Erreicht werden kann dies durch die Anwendung verschiedener Konzepte. Es werden geeignete Datenstrukturen benötigt, die eine effiziente Aggregation und Visualisierung der Daten in einem globalen sowie lokalen Umfeld unterstützen. Mit Hilfe von Textmining-Verfahren werden relevante Schlagwörter extrahiert und mit den Bewegungsdaten verknüpft. Die Realisierung der Konzepte stellt darüber hinaus eine Plattform für den Analyst dar, die es erlauben soll, Bewegungen über Zeiträume hinweg

[1] Where's George: http://www.wheresgeorge.com/

entweder kollektiv oder einzeln zu verfolgen, Rückschlüsse auf Reise- und Bewegungsgewohnheiten zu ziehen und den Analyst durch geeignete Visualisierungskonzepte bei der Exploration zu unterstützen.

Der Aufbau dieser Diplomarbeit orientiert sich an der Aufgabenstellung. Zunächst gibt Kapitel 2 einen Überblick über die Grundlagen. Dazu gehören zum einen Erläuterungen zu den verwendeten Datenquellen, wie man auf sie zugreift und wie sie zu charakterisieren sind und zum anderen wie geospatiale Daten und zugehörige Textinformationen repräsentiert werden können.

Kapitel 3 ordnet diese Arbeit in thematisch verwandte Arbeiten ein. Dazu gehören hauptsächlich jede, die thematisch abdecken, wie Bewegungsdaten analysiert und die vorhandenen Informationen gebündelt werden, um Überdeckungen zu vermeiden. Dies resultiert aus dem hohen Datenvolumen, welches zur Verfügung steht.

Die nachfolgenden Kapitel 4 und 5 widmen sich der Erarbeitung einer Lösungsstrategie sowie der Realisierung. Die Erarbeitung der Lösungsstrategie beginnt grundlegend bei der Sammlung der Twitterdaten und deren Aggregation. Dazu gehört unter anderem auch die Filterung der Daten, sodass Bewegungsdaten von Benutzern mit fehlerhaften oder unbrauchbaren Nachrichten frühzeitig ausgeschlossen werden. Des Weiteren wird hier das Datenbankschema erarbeitet, welches als Grundlage für die Aggregation und Visualisierungs- und Interaktionskonzepte dient. Das Kapitel der Realisierung ist in zwei Hauptteile aufgeteilt: die Aggregation und die Visualisierung. Beide orientieren sich zunächst an der zugehörigen Benutzeroberfläche, gehen dann jedoch gezielt auf die verwendeten Ideen und Algorithmen im Hintergrund ein.

Das Kapitel 6 widmet sich der Auswertung mit Hilfe einer Fallstudie. Im Rahmen dieser Fallstudie wird anhand von zwei Ereignissen mit weltweitem Einfluss und einem Vergleich des Reiseverhaltens untersucht, ob die verwendeten Daten eine sinnvolle Exploration zulassen und ob sich Bewegungsmuster erkennen, beurteilen und erklären lassen. Anschließend an dieses Kapitel folgt eine Zusammenfassung (Kapitel 7), welche das Erarbeitete nochmals reflektiert, ein Fazit zieht und einen Ausblick auf Themen gibt, die gegebenenfalls noch entwickelt und eingearbeitet werden können.

2 Grundlagen

Dieses Kapitel gibt einen Überblick über Techniken, Themen und Technologien, welche zum Verständnis der Ausarbeitung der Problemstellung beitragen. Zu Beginn werden Konzepte der Visualisierung mit ihrem relativ neuen Fachgebiet Visual Analytics vorgestellt. Nachfolgend werden die verwendeten Datenquellen charakterisiert und Grundlagen im Bereich von geospatialen Daten erörtert. Abschließend werden die Grundlagen zur Darstellung textueller Informationen erläutert.

2.1 Visual Analytics

Mit der steigenden Datenmenge wird es heutzutage immer schwerer, in kurzer Zeit daraus brauchbare Erkenntnisse zu gewinnen. Verwendete Visualisierungen basieren immer noch auf Techniken, welche veraltet und nicht mehr passend für diese mittlerweile sehr großen und komplexen Daten sind. Visual Analytics ist ein recht neues Gebiet der Visualisierung. Es versucht den Anwender zielgerichtet zu unterstützen, indem die Informationen visuell repräsentiert werden mit der Möglichkeit, durch direkte Interaktion mit der Darstellung, Schlussfolgerungen zu ziehen, einen erweiterten Einblick zu bekommen und in der Entscheidungsfindung zu helfen [KMS+08].

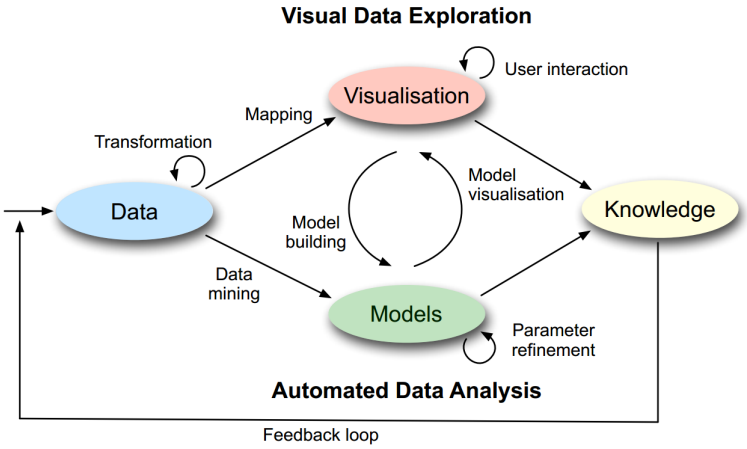

Abbildung 2.1: Der Visual Analytics Prozess (Bildquelle: [KKEM10])

Ein Bestandteil der Visualisierung ist die explorative Analyse. Der Benutzer startet ohne oder mit nur anfänglichen Erkenntnissen. Durch freie, interaktive Suche nach Strukturen und Trends führt die Visualisierung der Daten zu Hypothesen [Ert10]. Dieser Teil wurde abgebildet im Visual Analytics Prozess [KKEM10], wie in Abbildung 2.1 zu sehen ist. Es werden automatische sowie visuelle Analysemethoden kombiniert. Dieser Prozess umfasst folgende Hauptteile:

Datenmenge (Data): Bevor jegliche Analysemethoden auf die Daten angewendet werden können, wird im ersten Schritt der zumeist heterogene Datensatz vorverarbeitet und transformiert. Damit werden andere Datendarstellungen erreicht. Dazu gehören beispielsweise das Normalisieren oder das Gruppieren der Daten, um einige zu nennen.

Modelle (Models): Nach der Datentransformation kann der Analyst sich dafür entscheiden, automatische Analysetechniken anzuwenden. In diesem Fall werden Techniken des Data Mining angewendet, um Modelle aus den Daten zu generieren.

Visualisierung (Visualization): Nach der Datentransformation oder der Erstellung von Modellen können die Daten visualisiert werden. Durch die Visualisierung werden dem Analysten Interaktionsmöglichkeiten gegeben, mit denen unter anderem Modelle erstellt werden und die darin befindlichen Parameter verfeinert werden können. Beispielsweise kann ein Temperaturdatensatz auf verschiedene, passende Farben abgebildet werden. Der Nutzer kann nun ein zeitliches Modell daraus erstellen und dessen Parameter, wie z. B. die spatiale Abbildung, verändern, sodass nur noch Temperaturen im Raum Berlin angezeigt werden.

Wissen (Knowledge): Dies bezeichnet die Erkenntnis oder den Mehrwert, welche man durch die Modelle und die Visualisierung erhält.

Auffällig ist der Wechsel zwischen automatisierter und visueller, interaktiver Analyse. Dies hilft Anomalien und Fehler sehr früh zu erkennen, um qualitativ hochwertigere Schlüsse ziehen zu können.

B. Shneiderman führte in [Shn96] das Mantra für Informationssuche ein: Zuerst sich den Überblick verschaffen, dann Vergrößern und Filtern und Details auf Anfrage („overview first, zoom/filter, details on demand“). Am besten kann man sich dieses Mantra anhand eines Szenarios aus dem Bereich der Visualisierung vorstellen.

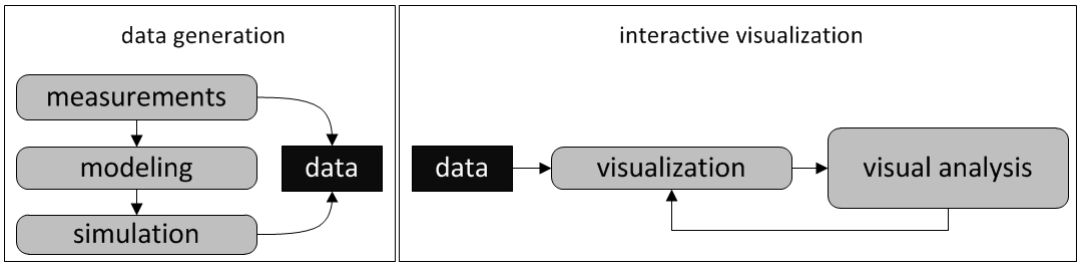

Abbildung 2.2: Szenario: Interaktive Nachverarbeitung (Bildquelle: [Ert10])

Abbildung 2.2 zeigt das Visualisierungsszenario der interaktiven Nachverarbeitung [Ert10]. Das Shneiderman-Mantra lässt sich hierauf im Bereich der direkten Interaktion mit der Visualisierung anwenden. Als Beispiel soll die Verkehrsdichte in England dienen. Die Daten werden durch Sensoren gesammelt und in einer Übersichtskarte für den Analyst dargestellt. Nun kann er sich durch Anwendung von Filter und Zoom nur den Bereich rund um London anzeigen lassen. Falls es den Analyst jetzt noch interessiert, was die Höchstgeschwindigkeiten auf Straßen sind, kann er sich dies auf Anfrage hin präsentieren lassen.

Bei Visual Analytics wird das Shneiderman-Mantra erweitert. Es heißt hier: Zuerst die Analyse, das Wichtige zeigen, dann Vergrößern, Filtern und nochmals analysieren, Details auf Anfrage („Analyse First - Show the Important - Zoom, Filter and Analyse Further - Details on Demand“) [KMS+08]. Dieses erweiterte Mantra resultiert aus dem Fakt, dass Daten immer größer und komplexer werden. Die Anwendung auf die Analyse der Verkehrsdichte in England würde sich hier wie folgt ändern: Die Daten werden zunächst vorverarbeitet. Vorstellbar ist, dass die Daten für einen Semantischen Zoom verarbeitet werden, d.h. umso weiter der Benutzer die Karte vergrößert, umso mehr Verkehrsdaten zu Straßen kommen hinzu. Somit wird immer das Wichtigste gezeigt. Durch Filter und Zoom kann beispielsweise die Verkehrsdichte wieder auf das Einzugsgebiet von London eingeschränkt werden. Spezielle Informationen gibt es wieder auf Anfrage.

2.2 Datenquellen

In dieser Diplomarbeit werden zwei Quellen verwendet: Twitter und Geonames. Dieses Kapitel gibt einen kurzen Überblick über beide Quellen.

2.2.1 Twitter

Twitter [twi] wurde in San Francisco im Jahre 2006 gegründet und gehört mittlerweile zu den größten Echtzeit-Informationsnetzwerken weltweit. Benutzer können hier ihre Interessensgebiete mit anderen Meinungen, Ideen und Geschichten verbinden sowie Neuigkeiten anderer Benutzer abonnieren. Twitter unterstützt mittlerweile mehr als 20 Sprachen und kann über das Smartphone als App verwendet werden. Anhand der Twitter-Daten können Bewegungsprofile angelegt werden, sofern eine Twitter-Nachricht (Tweet) auch mit Geokoordinaten abgesendet wird. Für die Erstellung der Bewegungsprofile werden ausschließlich Daten verwendet, die von den Benutzern ausdrücklich zur Veröffentlichung freigegeben wurden. In den nachfolgenden zwei Unterkapiteln wird näher auf die Eigenschaften von Twitter, speziell Tweets, eingegangen und wie diese Daten heruntergeladen werden können.

2.2.1.1 Characteristika

Eine Twitter-Nachricht, oder auch Tweet genannt, kann verschiedene Informationen sowie Zeichen, welche auf eine Informationssammlung oder weitere Kontakte schließen lassen, enthalten. Tabelle 2.1 zeigt, welche Informationen in einem Tweet enthalten sein können.

Bezeichnung	Beschreibung
Hashtag	Wird durch ein # gekennzeichnet und beschreibt ein Thema, unter welchem gegebenenfalls mehrere Nutzer Informationen bereitstellen.
Mention	Ein Benutzer kann einen anderen Nutzer erwähnen, indem er seinen Namen hinter ein @ schreibt.
Bild	Jedem Tweet kann zusätzlich ein Bild angehängt werden, entweder direkt oder über ein externes Programm.
URL	Eine Referenz auf eine andere Webseite kann immer mit **http://** eingeleitet werden.
Geokoordinaten	Sofern freigeschaltet, wird ein Tweet immer mit entsprechenden Geokoordinaten abgeschickt.

Tabelle 2.1: Enthaltene Informationen in einem Tweet

Des Weiteren gibt es verschiedene Arten von Tweets. Dazu gehören: der normale Tweet, der Antwort-Tweet, der Re-Tweet und der Direkte-Tweet. Ein Antwort-Tweet bezeichnet einen eigens verfassten Tweet als Antwort auf den Tweet eines anderen Nutzers. Eingeleitet wird eine Anwort immer mit der Mention des Nutzers, welchem man antworten will. Beim Re-Tweet wird eine bereits existierende Nachricht nochmals getwittert und mit einer kleinen Markierung versehen, die darauf hinweist. Ein Direkter-Tweet ist eine persönliche Nachricht an einen anderen Nutzer, welcher von niemandem sonst gelesen oder heruntergeladen werden kann. Alle anderen Tweets sind normale Tweets. [twi]

Das soziale Netzwerk Twitter wird von vielen verschiedenen Interessensgruppen genutzt. Um nur ein paar zu nennen: Privatpersonen, Parteien, Firmen, Entwickler, Schauspieler, Radiostationen, usw. Interessant für diese Arbeit sind alle, bis auf die Nutzer, hinter welchen ein Algorithmus steckt, wie es z. B. bei Bots der Fall ist. Bots sind Programme, welche automatisch Tweets veröffentlichen. Häufig besitzen Bots die Eigenschaft, Tweets im selben Format zu veröffentlichen oder gar von einer Geoposition aus, die nicht zutreffend ist. Diese Nutzertypen gilt es vor jeglicher Verarbeitung herauszufiltern, da sie eine erfolgreiche Analyse gegebenenfalls negativ beeinträchtigen. Nach einer Studie der Universität in Milan [Cal12] sind bis zu 46% aller Twitter-Nutzer Bots, die Firmen folgen. Bots folgen nicht nur Firmen, sondern auch normalen Nutzern. Es kann davon ausgegangen werden, dass die Anzahl an Bots, die normalen Nutzern folgen, vergleichbar hoch ist.

2.2.1.2 Daten von Twitter laden

Twitter bietet öffentlich zugängliche Schnittstellen, um auf Twitter-Daten zuzugreifen [twi]: die Search-API, die Rest-API und die Streaming-API. Die Search-API erlaubt Anfragen auf den Twitterinhalt, während die Rest-API die Grundelemente von Twitter zur Verfügung stellt, wie z. B. das Zugreifen auf Benutzerdaten, neue Tweets, usw. Zu beachten ist, dass bei diesen zwei Schnittstellen die Daten bei jeder neuen Anfrage explizit angefordert werden müssen. Bei der Streaming-API verhält es sich etwas anders: Nachdem eine Anfrage erstellt wurde, werden neu veröffentlichte Tweets in Echtzeit nachgeladen. Diese Schnittstelle ist speziell für Entwickler, in deren Anforderungen eine sehr hohe Datenmenge vorgesehen ist. Die Streaming Schnittstelle wird in drei verschiedene Kategorien unterteilt: die Public Streams, die User Streams und die Site Streams.

Public Streams: Über die Public Streams wird ein bestimmter Anteil aller veröffentlichter Tweets zur Verfügung gestellt. Laut Twitter werden pro Tag etwa 340 Millionen Tweets von 140 Millionen aktiven Benutzern veröffentlicht (Stand: März, 2012). Die Höhe des Anteils ist abhängig von der Zugangsebene, die dem Entwickler zugeteilt wird. Auf der untersten Ebene (default access level) erhält man Zugriff auf Tweets, die mit maximal 400 Schlüsselwörtern, 5000 Benutzerkonten und 25 geographischen Boxen verbunden sind. Es gibt verschiedene Endpoints, um auf die Public Streams zuzugreifen. Endpoints sind von Twitter festgelegte Schnittstellen für unterschiedliche Zwecke. Es wird unterschieden zwischen dem Filterhose, dem Samplehose und dem Firehose. Der Firehose dient dem Abgreifen aller veröffentlichten Tweets, während der Samplehose lediglich einen Teil aller Tweets zur Verfügung stellt, was über die Zugangsebene definiert ist. Anhand des Filterhose können Tweets verfolgt werden, die sich innerhalb einer bestimmten geographischen Box befinden.

User Streams: Beinhalten den Zugriff auf die Daten von jeweils einem einzelnen Nutzer.

Site Streams: Sind die erweiterte Version der User Streams. Hier ist es möglich auf mehrere User Streams gleichzeitig zuzugreifen.

Von den drei soeben beschriebenen Streams eignen sich für diese Diplomarbeit am besten die Public Streams unter Verwendung des Filterhose. Dadurch können effizient Bewegungsprofile über einen längeren Zeitraum hinweg erstellt werden. Das Abgreifen und Abspeichern der Daten ist kein Bestandteil dieser Arbeit. Dies geschah bereits in vorhergehenden Arbeiten, auf denen hier aufgebaut wird. Es liegt somit ein Datensatz vor, welcher nur Tweets mit Geopositionen enthält. Dies sind pro Tag etwa 4 Millionen Tweets.

2.2.2 Geonames

Geonames [geo] ist eine freie, geographische Datenbank, welche zum Download über einen Datenbankexport zur Online-Verwendung durch Webdienste zur Verfügung steht. Eine Besonderheit stellen die Feature-Codes von Geonames dar. Sie sind eine Zusatzinformation, beispielsweise für Städte, welche definiert, um was für Städte es sich handelt. So kann eine Stadt stark bevölkert, sehr religiös, eine Hauptstadt, etc., sein. Geonames ist unter anderem

auch für das Semantische Web von Bedeutung, da in der Geonames-Ontologie Toponyme eine eindeutige URL mit zugehörigem RDF Web-Service besitzen.

Des Weiteren existiert eine frei verfügbare Datensammlung aller Städte, die mehr als tausend Einwohner vorweisen können. Enthalten sind Informationen wie der Name, die exakten Geokoordinaten, die Zeitzone, uvm. Auf Grund der bekannten Population, der Geoposition und der lokalen Verfügbarkeit nach Download ist diese Liste wichtig für die Vorverarbeitung der gesammelten Twitterdaten. Die genaue Beschreibung findet in Kapitel 4 statt.

2.3 Darstellung geospatialer Daten

Die Darstellung geospatialer Daten spielt eine große Rolle, um z. B. Rückschlüsse auf das Reiseverhalten zu ziehen. In diesem Kapitel werden die notwendigen Grundlagen dazu erläutert. Zunächst wird allgemein beschrieben, wie Geokoordinaten bestimmt werden und wie sie eine Trajektorie bilden. Dabei spielen auch die Distanzberechnungen zwischen verschiedenen Koordinaten eine Rolle. Zum Schluss wird gezeigt, mit welchen Techniken diese Daten dargestellt werden.

2.3.1 Global Positioning System (GPS)

Das Global Positioning System – kurz GPS – dient der Positionsbestimmung, der Navigation und der Zeitmessung. Das GPS wird heutzutage für viele verschiedene Bereiche verwendet, hauptsächlich jedoch als Wegweiser. Für diese Arbeit spielt das GPS eine elementare Rolle, da es dazu verwendet wird, Twitternachrichten mit räumlichen Positionen zu versehen. Es ergeben sich zwei verschiedene Anwendungsszenarien: Beim ersten Szenario wird eine Nachricht vom heimischen Computer aus geschrieben. Da hier der Benutzer seine Position selbst bestimmen kann, kann dies zu etwas größeren Abweichungen führen. Dazu wird dem verfassten Tweet eine geographische Position zugewiesen. Twitter bietet dem Nutzer die Möglichkeit, selbst auszusuchen, mit welcher Position der Tweet referenziert werden soll. Durch den Durchbruch des Smartphones werden Twitternachrichten mittlerweile auch von unterwegs aus geschrieben. Heutzutage hat fast jedes Handy einen eigenen GPS-Empfänger, mit welchem die Position im zweiten Szenario genauer bestimmt werden kann und in welchem der Benutzer beispielsweise während einer Reise Nachrichten auf Twitter veröffentlicht.

Um die Position des Nutzers bestimmen zu können, muss eine Verbindung zu mindestens vier Satelliten vorhanden sein [DH09]. Im Falle, dass alle Satelliten sowie der Empfänger immer die exakte Uhrzeit kennen, sind nur drei Satelliten notwendig, um die Position zu bestimmen. Da exakte Uhren zu teuer und unhandlich sind, wird der Weg über vier Satelliten gewählt. Ein Satellit sendet immer seine Kennung und die Uhrzeit, zu welcher das Signal gesendet wurde, sowie seine Position an den Empfänger. Wenn das Signal beim Empfänger eintrifft, wird die Zeit abgeglichen und aus der Differenz der beiden Zeiten lässt sich die Entfernung zum Satelliten bestimmen. Um die Position des Empfängers nun

einzuschränken, werden zwei weitere Satelliten benötigt. Der Empfänger befindet sich im Schnittpunkt der Kugeln um die Satelliten, deren Radius der Distanz zwischen Empfänger und Satellit gleicht. Da die Empfängeruhren nicht immer auf die Millisekunde genau laufen, benötigt man – damit es nicht zu einer zu großen Abweichung der eigentlichen Position kommt – einen vierten Satelliten. Mit einer zeitlichen Abweichung schneiden sich die vier Kugeln der Satelliten nicht in einem Punkt. Es wird die Zeit folglich solange synchronisiert, bis sich alle vier in einem Punkt schneiden.

GPS wird in zwei verschiedene Klassen unterteilt, welche unter anderem die Genauigkeit ausdrücken:

Precise Positioning Service (PPS): Der PPS [Defo7] dient ausschließlich der militärischen Nutzung und wird verschlüsselt übertragen. Dieser GPS-Dienst ist zudem ziemlich genau. Er erreicht eine Genauigkeit von 5,9 m bei 95% der Messungen.

Standard Positioning Service (SPS): Der SPS [Defo8] wurde im Vergleich zum PPS entworfen, um jedem Bürger die Positionsbestimmung kostenfrei und unverschlüsselt zur Verfügung stellen zu können. Dabei gibt es Einschränkungen in der Genauigkeit. Momentan werden 7,8 m bei 95% der Messungen garantiert.

Um die Genauigkeit noch weiter zu erhöhen, bedient man sich einer Technik namens Differential-GPS (DGPS). Beim DGPS [DH09] wird die Genauigkeit verbessert, indem bekannte, auf der Erde befindliche Referenzstationen zur Bestimmung der Position hinzugenommen werden. Von diesen Referenzstationen ist die genaue Position bekannt und diese übermitteln die Korrektursignale an den Nutzer. Durch die Differenz können aktuelle Messfehler ermittelt und somit auch korrigiert werden.

Eine Studie, durchgeführt mit einem Smartphone (ein IPhone 3G) mit eingebautem GPS-Empfänger, testete die Positionsbestimmung mittels GPS, WiFi und dem mobilen Funknetzwerk [Zano9]. Im Mittel lag der Fehler beim GPS bei 7,7m horizontal und 8m vertikal. Die Bestimmung über WiFi und über das Funknetzwerk schnitten deutlich schlechter ab. Der horizontale Fehler lag im Mittel über WiFi bei 74m und über das Funknetzwerk bei 599m.

2.3.2 Trajektorie

Eine GPS-Trajektorie bezeichnet den Weg eines Benutzers über die Zeit [ZZXM09]. Eine benutzerspezifische Trajektorie wird erzeugt, indem die vermerkten Aufenthaltsorte – durch GPS gekennzeichnet – nach der Zeit miteinander verbunden werden. Im Rahmen dieser Diplomarbeit wird eine Reise oder eine Verbindung zwischen Städten auch als Benutzertrajektorie bezeichnet.

2.3.3 Haversine

Sind GPS-Koordinaten bekannt, so kann es unter Umständen interessant sein, den Abstand zwischen diesen Punkten zu berechnen, wie beispielsweise bei einem Navigationsgerät, bei

dem der Nutzer eine Rückmeldung haben will, wieviel Kilometer das Ziel noch entfernt ist. Die kürzeste Entfernung zwischen zwei Punkten auf einer Kugeloberfläche wird Orthodrome genannt, auch bekannt als Luftlinie [BS08].

Eine Möglichkeit die Länge eines Orthodrome zu berechnen ist die Verwendung der Haversine-Formel. R. W. Sinnott stellte diese Möglichkeit 1984 in dem Magazin „Sky and Telescopes“ vor [Sin84]. Die Haversine-Formel wird auf die Trigonometrie zurückgeführt und wurde früher von Seefahrern benutzt. Das Referenzmodell für die Erde ist beim Haversine eine Kugel. Dadurch kann die Berechnung der sphärischen Trigonometrie zugrunde gelegt werden. Die Haversine-Formel ist wie folgt definiert:

$$haversin(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1-\cos(\theta)}{2} \tag{2.1}$$

Die Bezeichnung Haversine leitet sich aus dem Englischen „half versed sine“ ab, was soviel bedeutet wie die Hälfte des Sinus versus. Der Sinus versus ist definiert als eins minus den Kosinus. Unter Verwendung der Haversine-Formel lässt sich die Formel für die Abstandsbestimmung zweier Geopositionen bestimmen. Die Positionen sind durch den Längengrad und den Breitengrad bestimmt ($pos1(lat1, long1)$ und $pos2(lat2, long2)$):

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat1) \cdot \cos(lat2) \cdot \sin^2\left(\frac{\Delta long}{2}\right) \tag{2.2}$$

$$c = 2 \cdot \arctan 2\left(\sqrt{a}, \sqrt{1-a}\right) \tag{2.3}$$

$$d = R \cdot c \tag{2.4}$$

R ist hierbei der Erdradius mit $6371km$ und $\Delta lat = lat1 - lat2$. Das Zwischenergebnis c gibt die Orthodrome im Bogenmaß an. d ist das endgültige Ergebnis, wobei die Einheit gleich der Einheit des Erdradius R ist.

Eine weitere Möglichkeit die Entfernung zweier Geokoordinaten zu berechnen, wurde 1975 von Thaddeus Vincenty in [Vin75] vorgestellt. Anstatt eine Kugel als Referenz wird ein Ellipsoid verwendet. Die Ergebnisse werden somit auch genauer, da die Erde mehr einem Ellipsoid als einer Kugel ähnelt. Die Genauigkeit bewegt sich in einem Bereich innerhalb 0,5 mm. Da zu jener Zeit die Rechenleistung von Computern sehr beschränkt war, ist die von Vincenty vorgestellte Implementierung sehr sparsam und einfach gehalten.

Bezieht man die Höhe mit Bezug zum Erdschwerefeld mit ein, führt dies zu einer verfeinerten Definition der Erde. Es wird somit zwischen dem Geoid als mathematische Erdfigur und dem Ellipsoid als einer dem Geoid annähernden Bezugsfläche unterschieden [Tor02]. Die Abstandsberechnung ist auf einem Geoid natürlich etwas komplizierter, sodass in dieser Arbeit nicht näher darauf eingegangen wird. Abbildung 2.3 zeigt einen Vergleich der Referenzmodelle.

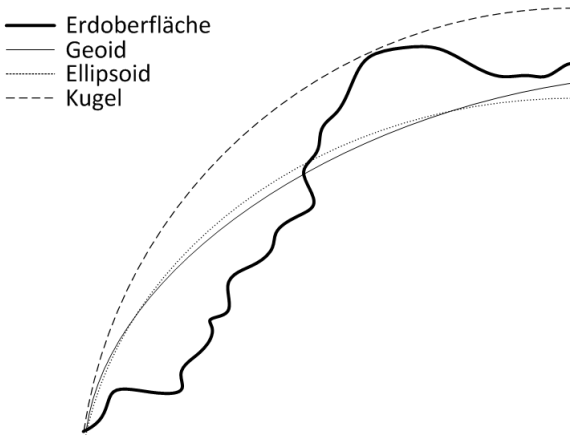

Abbildung 2.3: Vergleich verschiedener Referenzmodelle (Bildquelle: [Kla02])

Für die Distanzberechnung zwischen Geokoordinaten im Rahmen dieser Diplomarbeit reicht die Genauigkeit der Haversine-Formel aus. Es wird keine Genauigkeit im Millimeterbereich benötigt.

2.3.4 Daten-Mapping

Daten können auf verschiedenste Dinge abgebildet werden, wie etwa Formen, Farben, Texturen uvm. In diesem Kapitel werden drei für diese Diplomarbeit relevante Abbildungstechniken vorgestellt: auf Farbe, auf Zeit-Intervalle und auf Positionen mittels Voronoi-Zellen.

Abbildung auf Farbe

Bei der Abbildung auf einen spezifischen Farbwert wird ein skalarer Datenwert auf eine Farbe abgebildet. Beispiele dafür sind die Abbildung der Temperatur auf einen Farbwert zwischen Blau und Rot oder die Abbildung der Menschendichte auf der Erde auf einen dazu passenden Farbwert. Allgemein nennt sich diese Abbildung Transfer-Funktion. Typischerweise ist die Darstellung einer Transfer-Funktion RGBA, wobei A den Alpha-Wert angibt. Der weitverbreiteste Ansatz ist das Speichern der RGBA-Werte in einer Lookup-Tabelle, welches die diskrete Form einer Transfer-Funktion darstellt und auch in dieser Diplomarbeit so verwendet wird [Ert10].

Durch Abtasten der Transfer-Funktion an einer Menge von diskreten Punkten kann aus jeder beliebigen Funktion eine Lookup-Tabelle erstellt werden. Bei der Implementierung mittels einer interpolierten Lookup-Tabelle wird folgendermaßen vorgegangen [HJ04]: Der Skalarwert

wird auf einen Farbwert in einer Lookup-Tabelle abgebildet, welche eine benutzerdefinierte Größe an Farbeinträgen besitzt. Die Abbildung lässt sich nun wie folgt ausdrücken:

$$i = n \cdot \left(\frac{s_i - min}{max - min} \right) \tag{2.5}$$

Dabei stellt i den gesuchten Index in der Lookup-Tabelle dar, welche n-Einträge besitzt. Für den Skalarwert s_i gilt: $s_i \in]min, max[$. min stellt den kleinsten und max den größten vorhandenen Skalarwert dar. Der in der Klammer befindliche Bruch besitzt somit immer einen Wert zwischen 0 und 1 und gibt durch Multiplikation mit der Größe der Lookup-Tabelle immer den gesuchten Index aus.

Im Zusammenhang dazu wird zwischen Pre-Shading und Post-Shading unterschieden [Ert10]:

Pre-Shading: Den Daten werden zuerst die Farben zugewiesen und erst dann zwischen den Farben interpoliert.

Post-Shading: Findet Anwendung in dieser Diplomarbeit. Die Daten werden, wie zuvor beschrieben, erst interpoliert und dann die jeweiligen Farbwerte zugewiesen.

Mit der Abbildung auf Farbe sind allerdings auch einige Probleme verbunden. Um nur ein paar zu nennen [Ert10]: Die Farben sind nicht intuitiv zugeordnet; beispielsweise wird einer sehr warmen Temperatur die blaue anstatt die rote Farbe zugeordnet. Darüber hinaus gibt es Probleme mit der Wahrnehmung, wenn sich sehr ähnliche Farben überlappen. Auch kann es zu einer Überladung der Information kommen, falls zu viele Farbintervalle gewählt werden und der Benutzer nicht mehr klar dazwischen unterscheiden kann. Es sollte daher immer ein passendes Farbschema für die Abbildung gewählt werden.

Abbildung auf Zeitintervalle

Bei zeitlich stark abhängigen Datensätzen ist es unverzichtbar, die hierarchisch angeordneten Zeitdimensionen zu analysieren. Betrachtet man beispielsweise eine zehnstündige Flugreise, so sind die einzelnen Sekunden und Minuten gegebenenfalls uninteressant. Deshalb bietet es sich an, die Reise in stündliche Intervalle einzuteilen. Passiert in einer spezifischen Stunde etwas Interessantes, wie etwa Turbulenzen, eine Notlandung oder ähnliches, so kann man die Dimension beispielsweise auf Minuten reduzieren, um das Geschehen analysieren zu können.

Auch die Darstellung von zeitlich abhängigen Daten, wie etwa auf Landkarten, hängt sehr von der zeitlichen Dimension ab [KKEM10]. Werden Reisen z. B. in halbstündlichen Intervallen dargestellt, so werden Stopps dazwischen einfach ausgeblendet, auch um eine Informationsüberflutung zu vermeiden. Die Wahl der Dimension hängt somit davon ab, ob der Analyst einen Überblick über den Datensatz braucht oder sich auf ein spezielles Ereignis konzentriert. Eine individuelle Skalierung kommt ihm entgegen, da kleinere Ereignisse meistens in größeren enthalten sind.

Bei der Abbildung auf Zeit müssen immer zwei Aspekte im Sinn behalten werden [KKEM10]: Erstens muss man die zeitlichen Primitive beachten. Eine zeitliche Dimension kann als ein Zeitpunkt oder als ein Zeit-Intervall, das eine Zusammenfassung von Zeitpunkten ist, gesehen werden. Die Abbildung sollte hier von der Problemstellung und den Eigenschaften der Daten abhängig gemacht werden. Zweitens ist die temporäre Struktur der Daten bedeutend. Es wird zwischen drei verschiedenen Strukturen unterschieden: geordnete Zeit, verzweigende Zeit und multiple Perspektiven. Die geordnete Zeit wird unterteilt in lineare (kontinuierlicher Zeitfortschritt, so wie auch wir es wahrnehmen) und zyklische Zeit (wiederkehrende Ereignisse wie die Jahreszeiten usw.). Verzweigende Zeit tritt oft beim Vergleich verschiedener Szenarien auf. Multiple Perspektiven haben viel mit der Wahrnehmung eines Ereignisses aus verschiedenen Perspektiven zu tun. So beschreiben beispielsweise Menschen ihre Impressionen von Ereignissen in sozialen Netzwerkseiten aus unterschiedlicher Sicht.

Abbildung auf Position

Wie bei der Zeit hängt auch die Abbildung auf die Position vom Datensatz und dem Interessengebiet des Analysten ab [KKEM10]. Eine Variante, um Positionen abzubilden ist die Landkarte in Voronoi-Zellen zu unterteilen. Man besitzt eine Liste an interessanten Städten und möchte nun die nächstliegendsten Positionen direkt zuordnen.

Das Voronoi-Diagramm, oder auch Dirichlet-Zerlegung genannt, dient der Raumzerlegung und kann über die Euklidische Distanz $dist(p,q)$ zwischen den Punkten p und q mit $p = (p_x, p_y)$ und $q = (q_x, q_y)$ definiert werden, wie in 2.6 zu sehen ist [Fis04].

$$dist(p,q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \tag{2.6}$$

Sei nun initial die Punktmenge (für das Beispiel wäre es eine Liste mit Städten mit exakten Koordinaten) $P = \{p_0, .., p_{n-1}\}$ gegeben mit n verschiedenen Punkten. Das Voronoi-Diagramm für P ist die Unterteilung der Fläche in n Zellen, jeweils eine Zelle für einen Punkt. Liegt nun ein Punkt q in der Voronoi-Zelle von Punkt $p_i \in P$, so muss gelten:

$$dist(q, p_i) < dist(q, p_j) \qquad \forall p_j \in P, j \neq i \tag{2.7}$$

Abbildung 2.4 zeigt das fertige Voronoi-Diagramm.

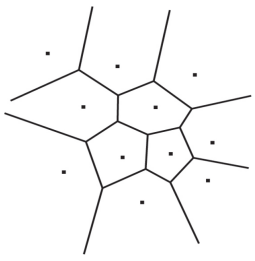

Abbildung 2.4: Voronoi-Diagramm (Bildquelle: [Fis04])

Zurück zum Städte-Beispiel: Angenommen, die Städte-Punktmenge sei $P = \{Stuttgart, Paris\}$. Nun sucht man die nächstliegende Stadt für *Vaihingen*. *Vaihingen* ist nach Definition in der Voronoi-Zelle von Stuttgart enthalten und wird so auch zugeordnet. Gerade in Übersichtskarten hat diese Abbildung große Vorteile, da die Information nicht verloren geht und der Benutzer keine Informationsüberflutung zu befürchten hat, es sei denn, es wurden zu viele relevante Städte gewählt.

2.3.5 Effizientes Rendern

Das effiziente Rendern spielt vor allem dann eine Rolle, wenn in einer Benutzeroberfläche enorm viele Informationen dargestellt werden. Diese Problemstellung bezieht sich hier auf die Darstellung und Suche von Städten in einer Karte. Sollen beispielsweise ausschließlich alle deutschen Städte dargestellt werden, so ist es ineffizient über alle weltweiten Städte zu iterieren. Effizient hingegen ist die Speicherung der Daten in Quadtrees [FB74]. Der Name leitet sich aus der Struktur ab: ein Baum, in welchem jeder Knoten vier oder keine Kinder besitzt. Der Quadtree beschreibt die zweidimensionale Abbildung eines Raumes auf einen Baum. Der Raum wird zunächst durch den Wurzelknoten in vier Quadranten aufgeteilt: NE, NW, SW und SE. Beim Einfügen einer weiteren zweidimensionalen Koordinate wird geprüft, in welchem Quadranten sich die Koordinate befindet und spannt dort wieder vier neue Quadranten auf. Abbildung 2.5 zeigt die exemplarische Darstellung eines Quadtrees mit der Annahme, dass die Quadranten, die auf derselben Baumtiefe liegen, gleich groß sind.

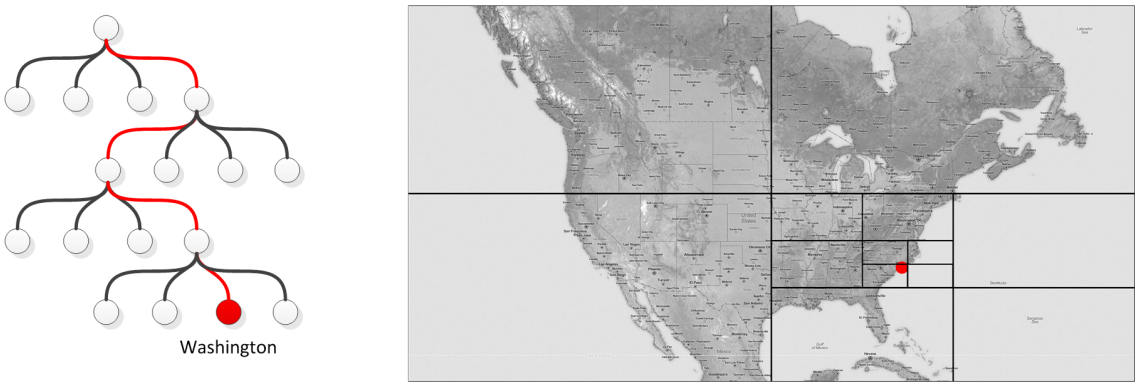

Abbildung 2.5: Quadtree

Wie zu sehen ist, können leere Bereiche einfach übersprungen werden. Bei der Suche nach „Washington“ wird als erstes der Quadrant ausgewählt, der Washington enthalten könnte. Dieses Verfahren wird rekursiv bis zum entsprechenden Knoten durchgeführt. Die Laufzeitersparnis ist hoch. Musste man vorher über alle Städte iterieren, so wird jetzt nur über Quadranten iteriert, was in einer logarithmischen Laufzeitkomplexität zum Ausdruck kommt.

2.4 Darstellung relevanter Textinformationen

Da wir heute im Zeitalter des Internets mit textuellen Informationen geradezu überflutet werden, spielt die Darstellung dieser Texte eine sehr große Rolle, um möglichst viele Informationen in kurzer Zeit zu erhalten. Es gibt dabei verschiedene Ansätze. Ein recht intuitiver Ansatz ist, den Text durch Bilder darzustellen. Im alten Ägypten war die Schrift durch Bilder und Symbole geprägt, die Hieroglyphen. Ursprünglich als reine Bilderschrift gedacht, haben sie Informationen schon damals visualisiert. Interessant ist der umgekehrte Weg, wie er heute oft gebraucht wird. Beispielsweise kann der Text

„Heute ist ein sehr sonniger Tag. Ich glaube, ich fahre heute mit meiner Familie in Urlaub!“

durch eine Sonne für das Wetter und eine Palme mit einer Hängematte am Strand für den Urlaub dargestellt werden. Diese zwei Bilder können aber nur gewählt werden, wenn die Semantik bekannt ist. Um einfach Informationen aus einem Text zu ziehen, ohne die Bedeutung zu kennen, gibt es effiziente Ansätze. Eine Möglichkeit ist, die Vorkommenshäufigkeit von Wörtern in Texten zu bestimmen und diese mit ihrem Vorkommen zu gewichten. Diese Gewichtung wird durch das tf-Maß ausgedrückt [MPS09]. Tf bezeichnet genau diese Gewichtung von Wörtern, steht für den englischen Ausdruck *term frequency* und wird angegeben als $tf_{t,d}$, wobei das Tiefgestellte den Term und die Dokument-Nummer kennzeichnet. Es lässt sich nun eine Ordnung hereinbringen, indem die gewichteten Wörter absteigend sortiert werden. Ein Nachteil ist, dass nun die Bedeutung verloren geht, da nur nach Häufigkeit sortiert wird. So sind nach dem tf-Maß die Sätze *"Daniel ist schneller als Lars"* und *"Lars ist schneller als Daniel"* identisch. Obwohl diese Sätze semantisch unterschiedlich sind, scheinen sie einen ähnlichen Inhalt zu besitzen.

Des Weiteren haben offensichtlich nicht alle Wörter eines Textes dieselbe Bedeutung. Daher kam die Idee der sogenannten *stop words* auf [MPS09]. Dies sind beispielsweise Bindewörter, Präpositionen usw., die nicht zur Bedeutung beitragen. Diese werden nicht in die Gewichtung miteinbezogen.

Um diese Listen mit gewichteten Wörtern zu visualisieren, gibt es mehrere Möglichkeiten. Die hier verwendete benutzt Stichwortwolken, auch bekannt als Tag-Clouds. Abbildung 2.6 zeigt die wohl bekannteste Art der Stichwortwolken-Visualisierung, Wordle [Fei]. Das Prinzip dahinter ist einfach zu verstehen. Die Wörter werden alphabetisch sortiert und bekommen zugewiesen, ob sie vertikal oder horizontal angezeigt werden sollen. Nun werden diese Wörter nacheinander auf einer unsichtbaren Spirale abgebildet. Dies verläuft wie folgt: Das erste Wort wird in der Mitte der Spirale gesetzt. Alle weiteren Wörter werden entlang der Spirale vom Ursprung aus verschoben, bis sie mit keinem davor eingefügten Wort überlappen. Neben den Vorteilen, dass der Platz besser ausgenutzt wird und in eine beliebige Form eingepasst werden kann, birgt diese Form der Darstellung jedoch auch Nachteile. Zum einen ist der Rechenaufwand hoch und zum anderen kann die Suche nach bestimmten Wörtern sehr umständlich werden [Jän].

Abbildung 2.6: Wordle Stichwortwolke (Erstellt mit [Fei])

Eine andere Möglichkeit ist das direkte Hintereinanderschreiben der Wörter, entweder alphabetisch oder nach Gewichtung sortiert, wie es in dieser Diplomarbeit umgesetzt wird. Die Vorteile sind die geringe Rechenzeit (im Vergleich zu Wordle) und die sehr gute Lesbarkeit [HR08] .

Für die Berechnung der Schriftgröße kann gleich vorgegangen werden wie für die Farbe in Kapitel 2.3.4. Im Gegensatz zu Farbwerten wählt man eine minimale und eine maximale Schriftgröße. Die angezeigte Schriftgröße $FontSize(i)$ berechnet sich aus [Wäs10]:

$$FontSize(i) = MinFontSize + IncreaseFactor \cdot Weight(i) \tag{2.8}$$

Der Gewichtungsfaktor errechnet sich aus dem Vorkommen des Wortes:

$$Weight(i) = \frac{f_i - f_{min}}{f_{max} - f_{min}} \tag{2.9}$$

Der Gewichtungsfaktor für das Wort i ergibt sich aus dem Quotienten zwischen der Differenz aus der Häufigkeit des Wortes f_i, dem niedrigsten vorkommenden Häufigkeitswert f_{min}, der Differenz aus dem häufigsten vorkommenden Häufigkeitswert f_{max} und dem niedrigsten f_{min}.

3 Verwandte Arbeiten

Im Folgenden wird diese Arbeit in existierende Arbeiten eingegliedert. Dazu werden verwandte Arbeiten zu den Kernthemen dieser Diplomarbeit vorgestellt. Zu den Hauptthemen gehören folgende zwei: Zum einen die Analyse von Trajektorien. Dabei wird untersucht, wie Informationen aus Trajektorien herausgezogen oder gegebenenfalls sogar semantisch analysiert werden können. Zum anderen der Umgang mit den enthaltenen Informationen. Es ist in diesem Zusammenhang interessant, wie einerseits extrahierte textuelle Informationen dargestellt und anderseits Trajektorien zusammengefasst werden können. Um diese Aufgabenstellung zu lösen kann man entweder den Weg über die Bewegung oder über die Aufenthaltsorte gehen. Beide werden in diesem Kapitel anhand existierender Arbeiten erörtert.

3.1 Analyse von Trajektorien

In diesem Teil wird exemplarisch gezeigt, welche technischen Möglichkeiten es gibt, den Analysten optimal bei der Exploration von großen Datenmengen zu unterstützen, und wie interessante Orte und Ereignisse gefunden werden können ohne direkt mit den dargestellten Daten interagieren zu müssen.

Die Umsetzung der Problemstellung dieser Diplomarbeit wird inspiriert durch die Arbeit von Adrienko und Adrienko [AA08]. Sie legten in ihrer Veröffentlichung den Fokus auf interaktive und visuelle Komponenten für Bewegungsdaten. Dazu gehören das Einschränken der Daten über die Zeit sowie die aggregierte Darstellung von Bewegungen. In [AA08] werden Visualisierung und Datenbankverfahren geschickt miteinander verknüpft, sodass sie sich gegenseitig verstärken. Da Reisen nicht explizit in den Daten definiert werden, die hauptsächlich aus Geokoordinaten und Zeitstempeln bestehen, gilt es diese durch visuelle Analyse herauszufinden. Aus den Daten werden zunächst die Rastplätze herausgefiltert. Dies sind Orte, wo der Benutzer angehalten hat. Dann werden die Rastplätze gruppiert, um oft besuchte Plätze zu identifizieren und am Ende auf einer Karte dargestellt. Die nachfolgende Exploration der Daten wird unterstützt durch Histogramme, die die Häufigkeiten von Pausen in zeitlichem Bezug angeben. Durch unterschiedliche Datenbankverfahren kann der Analyst die Daten darüber hinaus noch auf andere Arten darstellen lassen. Die Darstellung der Trajektorien führt oft zu Überdeckungen, welche durch einen Zeitslider als Interaktionskomponente reduziert werden können. Durch Auswahl des passenden Zeitintervalls werden die Daten entsprechend eingeschränkt. Leider geht dadurch oft die Übersicht verloren. Eine Möglichkeit sich die Übersicht zu verschaffen, ist das Clustern oder Bündeln von Daten. Hier kommt ein progressives Clustern zum Einsatz. Unter Verwendung von verschiedenen

Distanzfunktionen zum Clustern können diese nacheinander, also progressiv, aufeinander angewandt werden. Zusätzlich werden die gebündelten Trajektorien farbig, entsprechend dem Cluster kodiert. Die Dicke der Verbindung entspricht proportional der Anzahl der enthaltenen Bewegungen. Jede Verbindung wird auch durch einen Pfeil gekennzeichnet und gibt an, in welche Richtung die Trajektorien verlaufen. Durch die Ansicht über drei Dimensionen werden beispielsweise Bewegungsgeschwindigkeiten und hier zeitliches Verhalten offensichtlich. So können z. B. gebündelte Trajektorien, die womöglich gleich aussehen, in ihrer Dynamik verglichen werden.

Indem Ereignisse und Orte mit Reisen verbunden werden, kann mehr über die Semantik dieser Reisen in Erfahrung gebracht werden. Ein Ansatz, Ereignisse und interessante Orte über die Vorverarbeitung der Daten mittels Clustering zu finden, wurde von Kisilevich et al. in [KMK10] präsentiert. Dies ist beispielsweise für Gemeinden, Stadtplaner und Dienstleister von Interesse. Für diese Aufgabenstellung wurde das P-DBSCAN (Photo-Density-Based Spatial Clustering of Applications with Noise) entworfen, als Weiterentwicklung des auf dichtebasierten Algorithmus DBSCAN. Durch die Einführung eines Dichteschwellwerts und der adaptiven Dichte ist es möglich, vor allem sehr große Datenmengen zu analysieren und trotzdem interessante Ereignisse und Orte zu finden.

Wie soeben beschrieben, sind Reisen oft durch Zwischenstopps getrennt. Eine Reise kann somit in Teilreisen unterteilt werden. Durch die Analyse der Teilreisen und der Stopps können Rückschlüsse auf das Reiseverhalten gezogen werden, was auch für die Problemlösung in dieser Diplomarbeit von Bedeutung ist. Guc et al. präsentierten in [GMSK08] einen Ansatz für die Vereinigung von semantischer Analyse im Vorfeld und geeigneter Visualisierung, um durch geschickte Annotation der Daten Trajektorien klar in einen semantischen und physikalischen Teil zu trennen. Trajektorien bieten durch GPS hochwertige, geographische Informationen, jedoch ist die Semantik nicht immer ersichtlich. Das Prinzip von Episoden spielt hierzu eine große Rolle. Eine Trajektorie wird in Kurzreisen und Episoden unterteilt. Eine Episode gibt eine semantisch homogene Sektion einer Trajektorie an. Homogenität wird durch Geschwindigkeit, räumliche Distanz, Bewegungsfreiheit und die Abweichung einer geradlinigen Verbindung definiert. Kurzreisen sind dagegen die Zusammenfassung von Episoden, die sich jedoch auf einer semantisch höheren Ebene befinden. Es kann natürlich immer zu Fehleinschätzungen kommen. Unterstützt werden solche Analysen und Vermutungen jedoch auch über den Text, den ein Benutzer während seiner Reise veröffentlicht.

3.2 Clustern von Informationen

Bisher wurde beschrieben, inwiefern Trajektorien analysiert und Informationen extrahiert werden können sowie in welcher Weise mit ihnen interagiert werden kann. Dieser Teil setzt den Fokus nun auf Techniken zur Darstellung der Daten. Dazu gehören Aggregationsmodelle und Darstellungsmöglichkeiten für textuelle Daten sowie das gezielte Clustern von Informationen, entweder in Form von ganzen Verbindungen oder in Form von Aufenthaltsorten. Da diese Arbeit einen hierarchischen Ansatz über Voronoi-Zellen umsetzt, der abhängig

von der Vergrößerungsstufe ist, werden in diesem Abschnitt auch hierarchische Ansätze zu Bündelungen von Orten sowie Kanten erläutert.

3.2.1 Aggregation von Textdaten

Da in dieser Diplomarbeit der Ansatz über die Termfrequenz und die Darstellung über Schlagwortwolken gewählt wurde, werden im Folgenden drei Veröffentlichungen erläutert, die sich der Semantik in Schlagwortwolken widmen. Es gibt große Unterschiede in der Art der Anwendung und in der Art der Darstellung, um den Analysten bestmöglich bei der Exploration von neuen Datensätzen zu unterstützen.

Bei der Analyse einer Reise ist möglicherweise auch die textuelle Information unmittelbar davor und danach von Interesse. Es bietet sich an, diese Informationen aggregiert direkt auf der Verbindung anzuzeigen. In [KKEE11] wurde ein Ansatz auf Basis von Entitäten vorgestellt, welcher nicht nur den textuellen Kontext von Entitäten in Form von Schlagwortwolken visualisiert, sondern auch die Beziehungen zwischen diesen Entitäten. Der gewählte Name WordBridge folgt aus der Sichtweise, dass Brücken aus Schlagwörtern Entitäten miteinander verbinden. Die Idee wird visualisiert in Form von Knoten, die durch Liniensegmente miteinander verbunden sind. Das Verfahren ist in mehrere Schritte aufgeteilt: Zunächst werden Entitäten identifiziert und dann anhand der Termfrequenz Schlagwörter extrahiert. Dabei besitzt jede Entität eine Schlagwortwolke aus den am höchsten gewichteten Schlagwörtern. Eine Verbindung zwischen zwei Entitäten besteht, falls extrahierte Schlagwörter in beiden Entitäten vorkommen. Diese werden ebenso als Schlagwortwolke direkt auf der Verbindung visualisiert. Die Darstellung der Schlagwortwolken wird auf zwei Arten realisiert: kreisförmig und linear angeordnet. Durch passende Fallstudien wird das Potential dieser Idee offensichtlich, jedoch weisen die Autoren auch auf fehlende Benutzerstudien hin, um das neu erworbene Verständnis über bestimmte Texte zu verifizieren.

Diese Diplomarbeit basiert auf zeitabhängigen Daten, die unter Beachtung der Zeitkomponente der Visualisierung angepasst werden. Aus der Problemstellung heraus, dass generierte Schlagwortwolken lediglich einen Zeitpunkt festhalten, entwickelten Lee et al. in [LRKC10] einen Ansatz, der sich der Trendanalyse von Schlagwortwolken anhand von Sparklines widmet. Schlagwortwolken sind eine relative Repräsentation der Frequenz, der Bekanntheit oder der Wichtigkeit in Form der Textgröße. Aber genau wie die Daten selbst, entwickeln sich auch die Schlagwortwolken über die Zeit hinweg. Das allgegenwärtige Problem dabei ist, dass diese Entwicklung der Schlagwortwolken über die Zeit hinweg nicht repräsentiert wird. Der präsentierte Ansatz namens SparkClouds verbindet die Darstellung über Schlagwortwolken mit derjenigen einzelner Schlagwörter über die Zeit hinweg mittels Sparklines. Unterstützt werden zwei Anwendungsszenarien: das Szenario der Übersicht und das der punktuellen Zeitdarstellung. Die Lösung wird über zwei verschiedene Schriftgrößenkodierungen erreicht. Für das Szenario der Übersicht wird die Termfrequenz des gesamten Zeitraums betrachtet, wohingegen für die zeitlich punktuelle Betrachtung die Termfrequenz der Abbildung der Terme auf den gewählten Zeitpunkt entspricht.

Der Ansatz von Nguyen und Schumann in [NS10] geht noch einen Schritt weiter. Dieser heißt Taggram und verbindet die Darstellung von Termen in Schlagwortwolken mit geomarkierten Daten auf Karten. Die Terme können in beliebige Formen, wie etwa Ländergrenzen, gesetzt und dadurch als geovisuelle Analysemöglichkeit gesehen werden. Da diese Diplomarbeit Bewegungsprofile anhand von Reisen analysiert, wird dieses Vorgehen nicht weiter vertieft.

3.2.2 Clustern von Trajektorien

Das Clustern von Trajektorien entstand hauptsächlich aus dem Problem heraus, dass bei großen angezeigten Datenmengen die Überdeckung der Trajektorien so hoch war, dass nur noch schwer Trends und Richtungen erkannt werden konnten. Im Folgenden werden zwei Ansätze vorgestellt, die in dieser Arbeit keine Anwendung finden, jedoch als Ideengeber für hierarchisches Clustern dienen. Das Hauptziel beider Ansätze ist die Lesbarkeit bei einem großen Datensatz zu verbessern und Trends offensichtlich zu machen.

Um die Überdeckung der Kanten in einem Graphen zu reduzieren, nutzt der Ansatz von Holten und Wijk [HVW09] das Prinzip der Anziehungskräfte der Physik. Außerdem wird es erleichtert, Muster in den Kanten zu erkennen. Unter Verwendung der Anziehungskraft aus der Physik ist dieser Ansatz selbst organisierend, indem die Kanten sich gegenseitig anziehen. Dies funktioniert ohne die Erstellung einer Hierarchie oder eines Kontrollnetzes. Wenn sich die Kanten für diesen Ansatz eignen, werden sie in Segmente durch Punkte unterteilt, die sich gegenseitig anziehen. Dabei spielen nach dem physikalischen Modell die Kraft in lineare Richtung und die anziehende Kraft zum Gegenpart auf der anderen Linie eine Rolle. Durch die iterative Berechnung und Nachjustierung wird das Verfahren leistungsstärker.

Die Ergebnisse des Verfahrens mittels der Anziehungskraft erinnern äußerlich stark an Flow Maps, die von Phan et al. [PXY+05] präsentiert wurden. Ursprünglich stammen Flow Maps aus der Darstellung einer Bewegung, wie beispielsweise die Anzahl von Migranten usw. Verbindungen oder besser gesagt Kanten, die das gleiche Ziel haben, werden zusammengefasst. Die Umsetzung erfolgt hierarchisch, indem die Eingangsmenge an Knoten mit entsprechenden Positionen in hierarchische Cluster unterteilt und durch einen Binärbaum repräsentiert wird. Da die Wurzel des Baumes eine Kombination aus Clustern ist, wird der Baum so umgebaut, dass ein Knoten die neue Wurzel bildet. Der gewählte Knoten ist der Ursprung der Flow Map. Im nächsten Schritt werden die Cluster räumlich angeordnet. Die exakten Positionen gehen zwar verloren, jedoch werden die relativen Abstände zueinander bewahrt.

3.2.3 Clustern von Aufenthalten

Um Überdeckungen zu vermeiden, kann man auch die Anzahl an Positionen durch das Clustern von Aufenthalten reduzieren. Die Reduktion der Anzahl von Geopositionen findet auch in dieser Diplomarbeit Anwendung. Kisilevich et al. präsentieren in [KKR10] einen Ansatz, bei dem über das Clustern von Aufenthalten Informationen über Reisesequenzen aus

Trajektorien gewonnen werden. Dies kann dazu verwendet werden, interessante Plätze zu finden oder ein Verständnis für die naheliegenden Plätze zu gewinnen. Das Ziel ist darüber hinaus ein automatischer Ansatz, der dazu dient, Informationen aus semantisch annotierten Reisesequenzen zu gewinnen, indem in Fotos mit Geokoordinaten nach Sequenzmustern gesucht wird. Diese Idee besteht aus vier Hauptschritten, die ausgeführt werden: Im ersten Schritt werden anhand einer Datenbank, die interessante Orte enthält, die Fotos semantisch annotiert und zugeordnet. Dabei dürfen die Entfernungen der Fotos zu den interessanten Orten einen gewissen Schwellwert nicht übersteigen. Die übriggebliebenen Fotos werden im zweiten Schritt über ein dichtebasiertes Clustern bearbeitet. Im dritten Schritt werden unter Verwendung der Zeitangabe aus den Fotos individuelle Sequenzen gebaut, die durch die interessanten Orte verlaufen. Um zwischen Ansässigen und Touristen zu unterscheiden wird ein Zeitintervall von dreißig Tagen gewählt: Ein Aufenthalt mit Fotos, der kürzer dauert, beschreibt wahrscheinlich einen Touristen. Ist der Aufenthalt länger, so handelt es sich höchstwahrscheinlich um einen ansässigen Einwohner. Im letzten Schritt wird ein Algorithmus aus der Bioinformatik auf die Reisesequenzen angewendet, um gewisse Reisemuster zu entdecken. In der Evaluation wurde gezeigt, dass sich das Verfahren für verschiedene räumliche Ausmaße eignet: Dazu gehören Plätze, die viele Besucher und interessante Orte besitzen sowie Plätze, die relativ wenig Besucher und interessante Orte besitzen.

4 Entwurf

Dieses Kapitel erörtert die theoretische Umsetzung der Aufgabenstellung der Diplomarbeit. Logisch aufeinander aufbauend werden alle notwendigen Schritte und Konzepte beschrieben. Angefangen mit der konkreten Aufgabenbeschreibung in Kapitel 4.1, folgt das Kapitel 4.2, das beschreibt, wie die Daten gesammelt und zur eigentlichen Verarbeitung vorgehalten werden. Als Nächstes liegt der Fokus in Kapitel 4.3 auf der Datenvorverarbeitung, d.h. wie die Daten effizient aggregiert werden können, sodass sie als Basis für eine schnelle, interaktive Repräsentation geeignet sind. Aufbauend darauf werden in Kapitel 4.4 Konzepte für die Visualisierung und Interaktion vorgestellt, die zum Ziel haben, die aggregierten Daten möglichst effizient zur direkten Analyse zur Verfügung zu stellen. Abschließend wird gezeigt, wie textuelle Informationen konkret in dem zu entwickelnden Programm extrahiert und dargestellt werden können.

4.1 Aufgabenbeschreibung

Die geomarkierten, digitalen Spuren, die Nutzer in Twitter hinterlassen, sind meist öffentlich zugänglich. Aktuell werden an der Universität Stuttgart Twitter-Daten im Rahmen des BMBF Forschungsprojekts *VASA*[1] (Visual Analytics for Security Applications) aufgezeichnet und stehen für diese Diplomarbeit zur Verfügung. In dieser Diplomarbeit wird untersucht, ob sich mittels interaktiver Visualisierungen Bewegungsmuster erkennen und durch visuelle Annotation, basierend auf Kontextinformationen, beurteilen und erklären lassen. Daher wird als Erstes geprüft, inwieweit sich die Daten überhaupt für eine Analyse eignen. Dabei werden folgende Komponenten entwickelt und implementiert:

- **Datenstrukturen**, die geeignet sind, Bewegungsmuster effizient zu repräsentieren und zu aggregieren; des Weiteren sollen diese Datenstrukturen gleichermaßen für lokale und globale Anwendungen geeignet sein.
- **Strategien**, um aggregierte textuelle Informationen aus Social Media Nachrichten zu gewinnen und diese nachfolgend mit dem zugehörigen Bewegungsmuster zu verknüpfen.
- Verschiedene **interaktive Visualisierungstechniken**, die es dem Analysten ermöglichen, die aggregierten Daten zu explorieren, zu filtern und zu selektieren. Ferner sollen Bewegungsmuster aufgefunden und isoliert werden können.

[1] Pressemitteilung des BMBF: http://www.bmbf.de/press/3042.php

4.2 Datensammlung

Wie in Kapitel 4.1 bereits erwähnt, werden im Rahmen eines Forschungsprojekts an der Universität Stuttgart Twitter-Daten über lange Zeiträume gesammelt. Eine Besonderheit der Twitter-Daten: Sie sind alle mit Geokoordinaten versehen. Gespeichert und vorgehalten werden die Daten in verschiedenen Lucene-Repositories. Pro Tag gibt es zwei verschiedene vorliegende Repositories: Jeweils eins für die südliche und eins für die nördliche Hemisphäre. Lucene[2] ist eine sehr leistungsstarke, skalierbare, in Java programmierte, offene Bibliothek, die für Volltextsuchen geeignet ist. Dies sind nützliche Eigenschaften für das Speichern der Tweets, die, wie in Abschnitt 2.2.1.2 beschrieben wurde, über die Twitter-Schnittstelle gesammelt werden.

4.3 Datenvorverarbeitung

Bei der Vorverarbeitung geht es erst einmal darum, Datenstrukturen zu erzeugen, die dafür geeignet sind, Bewegungsmuster effizient zu repräsentieren und zu aggregieren. Die Vorverarbeitung enthält mehrere Teilschritte: Das Anbieten von zusätzlichen Daten zur Anreicherung der Informationen, das Extrahieren der Bewegungsmuster in passende Strukturen sowie die Aggregation textueller Informationen.

4.3.1 Vorbereitung

In der Vorbereitung werden Daten zur Anreicherung der Twitter-Daten gesucht. Für die Extraktion der Bewegungsmuster in passende Strukturen (siehe Abschnitt 4.3.2.1) werden Informationen über Städte ab einer bestimmten Größe benötigt. Es ist zuverlässiger diese Städtelisten lokal zu halten, anstatt für jede Stadt eine Anfrage an Dienste wie Geonames zu schicken, um die nächstgelegene Stadt mit einer gewissen Mindesteinwohnerzahl zu finden. Das Problem ist hierbei die Verlässlichkeit der Dienste, da keine Uptime von 100% garantiert wird [geo]. Im schlimmsten Fall wird die Vorverarbeitung unterbrochen. Für diese Arbeit werden zwei Quellen verwendet: Für Städte mit internationalen Drehkreuzen DBPedia[3] und für alle restlichen Städte mit einer Mindestgröße von tausend Einwohnern Geonames.

Drehkreuze von DBPedia

DBPedia ist die semantische Abbildung von Wikipedia auf eine Ontologie und wird über das Resource Description Framework (RDF) dargestellt. Wie bei Datenbanken gibt es auch hierfür eine Abfragesprache: Die SPARQL Protocol and RDF Query Language (SPARQL) [HFBPL09].

[2]Lucene: http://lucene.apache.org/
[3]DBPedia: http://de.dbpedia.org/

Listing 4.1 Extraktion der Drehkreuze aus Wikipedia

```
SELECT DISTINCT ?s ?iata ?lat ?lon ?s1h ?s1d {
  ?s  rdf:type <http://dbpedia.org/ontology/Airport> .
  ?s  rdf:type <http://dbpedia.org/ontology/Airport> .
  ?s  <http://dbpedia.org/property/iata> ?iata .
  ?s  <http://www.w3.org/2003/01/geo/wgs84_pos#lat> ?lat .
  ?s  <http://www.w3.org/2003/01/geo/wgs84_pos#long> ?lon .
  ?s  <http://dbpedia.org/property/stat1Header> ?s1h .
  ?s  <http://dbpedia.org/property/stat1Data> ?s1d .

  OPTIONAL {
    ?s <http://dbpedia.org/property/stat2Data> ?s2d .
    ?s <http://dbpedia.org/property/stat2Header> ?s2h .
  }
}
```

Listing 4.1 zeigt die verwendete SPARQL-Abfrage für die Liste der internationalen Drehkreuze. Die Abfrage liefert als Ergebnis einen RDF-Graphen, der die DBPedia-Ressource des Flughafens enthält, dessen Flughafenkürzel, die exakte Position nach Längen- und Breitengrad, den Datenheader sowie den passenden Datenwert. Anhand des Datenwerts lassen sich die größten Drehkreuze leicht ausfindig machen.

Städte mit über 1000 Einwohnern von Geonames

Die Liste mit allen Städten, die mehr als 1000 Einwohner besitzen, ist wesentlich einfacher zu beschaffen als die Liste der Drehkreuze. Geonames kann auch als komplette Datenbank heruntergeladen werden. Diese Datenbank ist alphabetisch sortiert und enthält des Weiteren Listen mit einer Mindestgröße an Einwohnern. Es gilt nur noch, die Liste herunterzuladen und in die lokale Datenbank zu übertragen.

4.3.2 Aggregation von Trajektorien

Bei der Aggregation der Trajektorien gilt es einige Besonderheiten zu beachten. Dazu gehört der nahtlose sowie skalierbare Übergang von einer globalen Ansicht zu einer lokalen. Hier stellt sich die Frage, wie viel Information bei einer Übersichtskarte zweckdienlich ist und wie viel Information bei einer detaillierten Ansicht. Um die Daten passend für diese Problemstellung zu verarbeiten, muss untersucht werden, welche Daten innerhalb des Datensatzes geeignet sind und welche nach gewissen Kriterien herausfallen.

4.3.2.1 Lokal vs. Global

Bei der Problemstellung Lokal vs. Global geht es um einen geeigneten Detaillierungsgrad für Informationen auf der Landkarte. Als Beispiel dient die Kartendarstellung von Google

Maps[4]. Sucht man beispielsweise nach dem Schlagwort „Hotel“, so bekommt man umso mehr Ergebnisse geliefert, je weiter die Karte vergrößert wird. Abbildung 4.1 zeigt am Beispiel von Google Maps, wie sich die Kartenvergrößerung auf die Informationsdarstellung auswirkt.

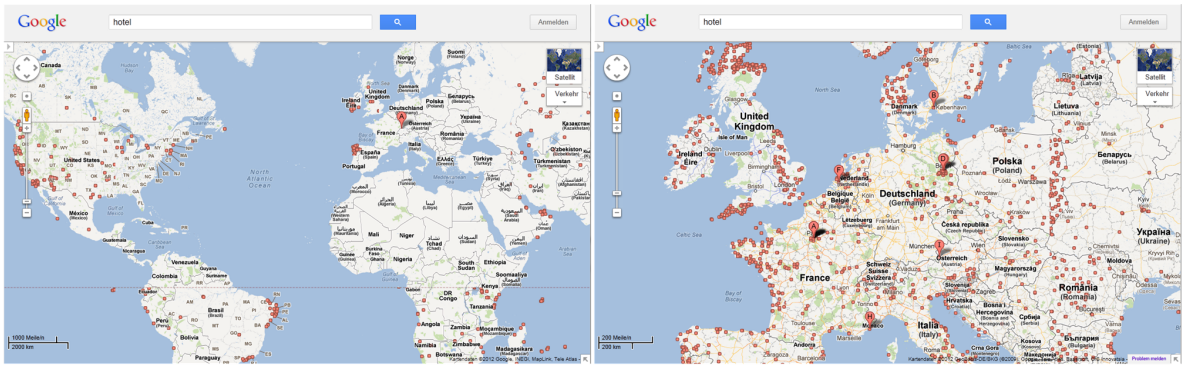

Abbildung 4.1: Vergleich des Informationsgrades unter verschiedenen Auflösungen in Google Maps mit den Suchergebnissen für das Schlagwort „Hotel“. Links: Darstellung der Übersichtskarte. Rechts: Erhöhung des Informationsgrades durch die Vergrößerung des europäischen Teils der Landkarte.

Für die Realisierung des Semantischen Zooms wurde in dieser Arbeit ein Ansatz, basierend auf der Gruppierung in Voronoi-Zellen, betrachtet. Insgesamt gibt es bei der zu verwendenden Karte fünfzehn Vergrößerungsstufen. Betrachtet man die Karte alleinstehend, so kann man beobachten, dass sich die Informationen bei den weiteren Vergrößerungsstufen elf, acht und fünf bedeutend ändern. Bedeutend ändern heißt in diesem Zusammenhang, dass eine nennenswerte Menge an Informationen hinzukommt. Beispielsweise kommen im Schritt von zwölf zu elf zusätzlich zu den Ländernamen etliche Städtenamen hinzu.

Bestimmt man also die Voronoi-Zellen-Mittelpunkte für die verschiedenen Vergrößerungsstufen, so ergibt sich folgende Städteaufteilung: Für die Stufen elf bis fünfzehn sind die dargestellten Städte die etwa 50 größten Drehkreuze der Welt, extrahiert aus der Liste, welche über DBPedia ausgelesen wurde. Für die Stufen sieben bis zehn werden statt der Drehkreuze alle Städte mit mehr als 150 000 Einwohnern als Mittelpunkte der Voronoi-Zellen verwendet. Die Städte mit mehr als 50 000 Einwohnern kommen für die Stufen sechs bis acht zusätzlich hinzu. Für alle restlichen Vergrößerungsstufen werden außerdem alle Städte mit mehr als 15 000 Einwohnern hinzugenommen. Durch das Hinzunehmen von immer mehr Städten erreicht man erstens den Effekt, dass je nach Vergrößerungsstufe mehr oder weniger Informationen angezeigt werden; zweitens werden – falls die Distanz der Reiseverbindungen ebenfalls mit dem Vergrößerungsfaktor mit angepasst wird – die Verbindungen zwischen den Städten granularer. Ein Beispiel: Ein Twitter-Benutzer reist von Berlin über Wolfsburg

[4]Google Maps: http://maps.google.com/

und Frankfurt nach Stuttgart. Bei der Übersichtskarte wird lediglich die Reise von Berlin nach Stuttgart mit ihren Randinformationen dargestellt. Bei weiterer Vergrößerung wird irgendwann die Verbindung in drei verschiedene Verbindungen aufgeteilt: eine von Berlin nach Wolfsburg, eine von Wolfsburg nach Frankfurt und eine von Frankfurt nach Stuttgart.

Um Überdeckungen zu vermeiden und den Informationsgehalt anzupassen, ist im optimalen Fall eine dargestellte Reiseverbindung nicht größer als der Sichtbereich des Analysten. Diese Verbindungen lassen sich relativ einfach herausfiltern, indem bei einer bestimmten Vergrößerungsstufe Verbindungen bis zu einer gewissen Maximallänge zugelassen werden.

Durch die Aufteilung in Voronoi-Zellen variiert auch die Anzahl der darzustellenden Benutzer. Umso weniger Städte dargestellt werden, desto weniger Benutzer sind auch beteiligt. Dies liegt daran, dass bei der Berechnung der kürzesten Wege von der Benutzerposition zu den nächstgelegenen Voronoi-Städten die Wahrscheinlichkeit bei wenig betrachteten Städten sehr hoch ist, dass Start und Ziel einer Reise einer einzigen Stadt zugeordnet werden. In diesem Fall wird die Reise nicht angezeigt.

4.3.2.2 Zeit

Eine Trajektorie beschreibt immer den Weg, den ein Benutzer geht – in diesem Zusammenhang auch bekannt als eine Reise. Wie bereits in Abschnitt 4.3.2.1 beschrieben, werden einzelne Positionen den am nächsten liegenden Städten zugeteilt, abhängig von der Vergrößerungsstufe der Landkarte. Es liegt hier nun nahe, die Trajektorien in Aufenthalte und Reisen zu unterteilen. Anders als im Ansatz von Alvares et al. [AFM+07], wo Aufenthalte und Reisen anhand der zeitlichen Daten bestimmt wurden, werden in diesem Ansatz außerdem Reisen von der Lokation abhängig gemacht. Es dreht sich also hier erst um eine Reise, wenn grobe zeitliche Randbedingungen und der Übertritt von einer Voronoi-Zelle zur anderen gegeben sind. Betrachtet man die Zeitkomponente in diesen Reisen, so ergibt sich folgendes Problem: Befinden sich mehrere Benutzer auf einer Reise in einem gewissen Zeitraum, jedoch mit unterschiedlichem Start und Ende, so würde die Reiseverbindung bei einer Filterung nach Zeit auf der Landkarte immer wieder auftauchen und verschwinden. Die Reisen – vor allem, wenn mehrere Benutzer beteiligt sind – sollten als eine gemeinsame gesehen werden, jedoch mit unterschiedlichen Gewichtungen, sodass eine Reise, je nach gesetztem Zeitfilter, an Relevanz gewinnt oder verliert. Hierzu werden die Reiseverbindungen in halbstündliche Intervalle eingeteilt, die dementsprechend auch gewichtet werden. Dies ist so umsetzbar, da genaues Start- und Enddatum nicht mit Sicherheit bestimmt werden können. Eine minütliche Filtermöglichkeit ist nicht rentabel, weil die Datenbank durch die Verwendung von minütlichen anstatt halbstündlichen Intervallen viel mehr Daten halten muss, was auch zu Leistungseinbußen führen kann.

Im Rahmen dieser Diplomarbeit wurden zwei Lösungsansätze herausgearbeitet, wie das Problem der wechselhaften Reiseverbindungen behoben werden kann: Zum einen über die Anzahl der Benutzer, die beteiligt sind, und zum anderen über die Anzahl der Zeitintervalle über den Reisezeitraum. Beide Ansätze beschäftigen sich mit der Gewichtung über die Zeit der Verbindungen.

Gewichtung über Anzahl der beteiligten Benutzer

Bei der Gewichtung nach Anzahl der aktiven Benutzer werden die Benutzer den Zeitintervallen zugeordnet, in welchen sie getwittert haben, und ins Verhältnis zu allen Benutzern für eine Reise gesetzt. Die Anzahl der Benutzer der selektierten Zeitintervalle steht somit im Verhältnis zu allen Benutzern der Reiseverbindung und trägt genau soviel zur Gewichtung bei. Ein Beispiel soll dies verdeutlichen: An den Tagen Montag und Dienstag sind insgesamt 100 Benutzer auf Reise. Am Montag haben 30 Benutzer getwittert und am Dienstag 70. Wählt der Analyst Montag als Zeitraum aus, so berechnet sich die Gewichtung aus dem Quotienten von 30 und 70. Die Gewichtung der Reise ist in diesem Fall 0,3 und kann auf Farbe, Opazität oder ähnliches abgebildet werden.

Dieser Lösungsansatz ist anwendbar, aber nicht in allen Fällen optimal. Zum einen werden nicht die Reisen an sich betrachtet, sondern eher die Aufenthalte, bei denen Twitter-Nachrichten geschrieben wurden. Zum anderen stellt sich die Frage, was passiert, wenn an einem Tag kein Benutzer eine Nachricht schreibt. In diesem Fall taucht das Problem der wechselhaften Verbindungen wieder auf, sofern nicht eine Mindestgewichtung eingeführt wird.

Gewichtung über Anzahl der Zeitintervalle

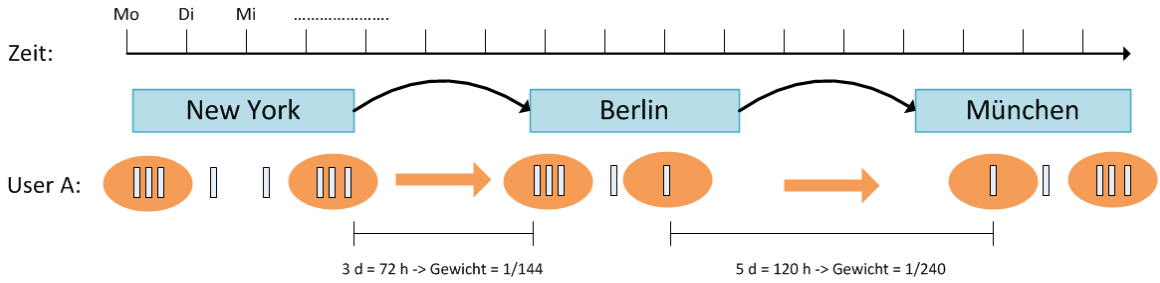

Abbildung 4.2: Beispiel für die Gewichtung nach Anzahl der Zeitintervalle

Zweckdienlicher als die Gewichtung über die beteiligten Benutzer ist die Gewichtung über die Anzahl der Zeitintervalle. Hierbei wird die Anzahl der Intervalle zu einem maximalen Gesamtgewicht addiert. Dies hat bei mehreren Benutzern auf einer Route den Vorteil, dass kein Mindestgewicht angegeben werden muss, damit die Verbindung zwischenzeitlich nicht komplett verschwindet. Bei mehreren Benutzern auf einer Route kommt ebenfalls der Vorteil hinzu, dass je nach ausgewählter Zeitspanne die Gesamtreisen einzelner Benutzer relevanter werden. Reisen beispielsweise zwei Benutzer von Paris nach New York, so beginnt für Benutzer A die Reise am Montag und endet am Freitag, für Benutzer B beginnt die Reise am Dienstag und endet am Mittwoch. Wird die Zeit auf Dienstag und Mittwoch gefiltert, so

wird die Reise von Benutzer B relevanter als die Reise von Benutzer A. Abbildung 4.2 zeigt exemplarisch das Vorgehen für die Bestimmung von Reisen.

Eine Reise beginnt mit dem letzten Tweet, der in einer Voronoi-Zelle veröffentlicht wurde und endet mit dem ersten Tweet beim Übertritt in eine andere Voronoi-Zelle. Dazu sei zu sagen, dass für die textuelle Analyse via Berechnung der Termfrequenz lediglich die Twitternachrichten am Tag des Reisestarts und am Tag des Reiseendes miteinbezogen werden. Zusätzlich werden die extrahierten Terme mit dem Gewicht der Verbindung multipliziert. So ist gewährleistet, dass, umso relevanter eine Reise wird, auch die Terme der Reise relevanter werden. Dies ist ein Beispiel für die Umsetzung des Visual Analytics Prozesses, da die präsentierten Daten durch eine Benutzerinteraktion – wie etwa die Zeitfilterung – neu analysiert und dargestellt werden.

4.3.2.3 Vorfilterung der Trajektorien

Ein Problem bei dem Gebrauch von großen Mengen an ungefilterten Daten ist der Anteil an unbrauchbaren Daten. Dazu gehören Daten, die von sogenannten Bots veröffentlicht werden (siehe dazu Abschnitt 2.2.1.1) oder Daten, bei denen der Aufenthaltsort nicht plausibel ist. In manchen Fällen bestimmen Benutzer ihre Position absichtlich falsch, was jedoch die Ausnahme darstellt. Es gilt eine Lösung zu finden, sodass die Anzahl der relevanten und verwendbaren Nachrichten, die im Rahmen der Vorverarbeitung übrig bleiben, besonders hoch ist. Dazu werden die Daten gefiltert, auch wenn dabei Nachrichten herausfallen, weil sie minimal vom Standard abweichen, sonst aber in Ordnung sind.

Bei Bots ist sehr auffällig, dass Nachrichten oft dem selben Muster folgen und häufiger via Twitter veröffentlicht werden als die von normalen Benutzern. Um diese herauszufiltern, wird die Anzahl der veröffentlichten Tweets zahlenmäßig eingeschränkt. Wie bereits in Abschnitt 2.2.1.2 erwähnt wurde, werden 340 Millionen Tweets von 140 Millionen aktiven Nutzern täglich veröffentlicht (Stand März, 2012). Dies bedeutet, dass im Schnitt ein aktiver Benutzer 2,4 Tweets pro Tag veröffentlicht. Es gibt natürlich Situationen, in denen mal mehr, mal weniger geschrieben wird. Daher wird davon ausgegangen, dass über den gesamt betrachteten Zeitraum hinweg ein Benutzer nicht mehr als einen Tweets pro Stunde veröffentlicht. Betrachtet man beispielsweise nur einen Tag, so dürfen an diesem Tag nicht mehr als 24 Tweets veröffentlicht worden sein. Bei einer Woche sind das bereits insgesamt 168 Tweets. Die gewählte Einschränkung liegt deutlich über dem Schnitt und lässt somit einen größeren Spielraum zu.

Auch die Reisegeschwindigkeit ist oft auffällig. So kann es vorkommen, dass ein Benutzer eine Nachricht um acht Uhr morgens in Berlin veröffentlicht und nur eine Stunde später in New York. Das ist oft darauf zurückzuführen, dass Benutzer den Standort des Tweets manipulieren können. Um die meisten Nachrichten dieses Typs herauszufiltern, wird eine Reisegeschwindigkeit von etwa 1000 Kilometern pro Stunde angenommen. Das heißt, dass zwischen Reiseantritt und -ende bei einer Distanz von 1000 Kilometern nicht mehr als eine Stunde liegen darf.

4.3.3 Aggregation textueller Informationen

Durch den textuellen Informationsgehalt einer Reise können beispielsweise Rückschlüsse auf Erlebnisse und Empfindungen gezogen werden. Abbildung 4.2 zeigt alle von einen Benutzer veröffentlichten Tweets. Aus ihnen wird die Information gewonnen, dass der Benutzer zwei Reisen angetreten hat, einmal von New York nach Berlin und einmal von Berlin nach München. Wie bereits in Abschnitt 4.3.2.2 beschrieben, sind lediglich die Nachrichten am Tag des Reisestarts und am Tag des Reiseendes relevant. Dies lässt sich dadurch erklären, dass es bei diesen Nachrichten sehr wahrscheinlich ist, dass sie zur Bedeutung der Reise entscheidend beitragen. Alle anderen Nachrichten werden verworfen.

Mittels der Term Frequency werden die in den Nachrichten vorhandenen Terme dahingehend eingestuft, wie relevant sie sein könnten. Zusätzlich werden die extrahierten Terme jeweils mit dem Gewicht eines Zeitintervalls multipliziert und zugewiesen. Wie bei der Relevanz der Reise bei gesetzter Zeitfilterung, ändert sich so auch die Relevanz der Terme abhängig von den Benutzern.

4.3.4 Zugrundeliegendes Datenbankschema zur Annotation

Die in diesem Kapitel beschriebene Vorverarbeitung und Aggregation muss für eine erfolgreiche Analyse effizient repräsentiert werden können. Das Datenbankschema in Abbildung 4.3 soll dieser Aufgabe gerecht werden. Es ist schwer, in Echtzeit eine neue Analyse auf den Daten durchzuführen. Daher ist es zweckdienlich, so viel wie möglich bereits aggregiert bereitzustellen. Dabei wird die Struktur von einzelnen Benutzertrajektorien auf die Grundelemente heruntergebrochen und genau so in der Datenbank abgelegt.

Term	(term_id, term, global_term_frequency)
TermMapping	(traj_id, term_id, term_frequency)
Connection	(traj_id, time_start, time_end, zoom_stage, weight, start_voronoi_id, end_voronoi_id)
Place	(place_id, lat, lng)
User	(user_id, user_location, user_mention, user_screenname)
UserMovement	(traj_id, user_id)
Tweet	(doc_id, text, place_id, lat, lng, created_at, user_id, hashtags, urls)

Tabelle 4.1: Detailansicht der Datenbanktabellen

Zu beachten ist, dass die Trajektorie eines Benutzers nicht im Ganzen, sondern in Zeitintervalle, die vor der Vorverarbeitung festgelegt wurden, abgespeichert wird; jene Zeitintervalle, die vor der Vorverarbeitung festgelegt wurden. Der Vorteil ist, dass Teilausschnitte aus Reisen ohne weitere Probleme visualisiert und zusätzlich auf die einzelnen Zeitintervalle

einer Reise separate Informationen abgelegt werden können. So wird mit den Termen, die aus den Nachrichten extrahiert werden, verfahren. Sie bekommen eine extra Gewichtung pro Zeitintervall, signalisiert durch das Attribut *term_frequency* in der Tabelle *TermMapping*.

Abbildung 4.3: Zugrundeliegendes Datenbankschema zur Annotation

In Tabelle 4.1 wird das zugrundeliegende Datenbankschema noch einmal detailliert gezeigt. Vor jeder Vorverarbeitung müssen die Tabellen *Tweet*, *User* und *Place* befüllt werden, da diese

ansonsten nicht stattfinden kann. Diese Daten kommen von den bereits gesammelten Twitter-Daten und den gesammelten Städten, welche für die verschiedenen Vergrößerungsstufen der Karte verwendet werden. Aus diesen werden die Daten vorverarbeitet, aggregiert und in den Tabellen *UserMovement* für die Abbildung von Benutzern auf Zeitintervalle, *Connection* für alle vorkommenden Zeitintervalle, *TermMapping* für die Abbildung von extrahierten Termen auf Zeitintervalle und *Term* für alle extrahierten Terme, abgelegt.

4.4 Visualisierungs- und Interaktionskonzepte

Die vorverarbeiteten Daten müssen auch effizient visualisiert werden, damit es dem Analysten möglich ist, mit den dargestellten Daten auf effiziente Art und Weise explorativ interagieren zu können. In diesem Kapitel werden verschiedene Konzepte vorgestellt, welche als mögliche Lösung für die Problemstellung in Frage kommen. Zunächst wird ein Konzept vorgestellt und gegen andere abgewogen, um zu sehen, wie die Daten bei der Darstellung gehandhabt werden können. Dann werden verschiedene Konzepte zur Filterung und Analyse der Trajektorien veranschaulicht.

4.4.1 Bereitstellung der Daten

Die Daten, die in der Datenbank vorliegen, beschreiben jeweils ein vorher festgelegtes Zeitintervall auf einer Trajektorie. Natürlich müssen diese einzelnen Zeitintervalle geladen werden, um sie auf einer Landkarte aggregiert als Trajektorie oder besser gesagt als eine Reiseverbindung darzustellen. Es wäre aber ineffizient, bei einer Benutzerinteraktion, wie beispielsweise beim Markieren mit der Maus, durch alle Zeitintervalle zu iterieren bis das entsprechende gefunden wird. Aus diesem Grund werden die Daten, nachdem sie geladen werden, in Städteverbindungen gepackt. Jede Verbindung kennt ihre Zeitintervalle, jedoch erhöht sich die Verarbeitungsgeschwindigkeit. Wurden davor alle Zeitintervalle bei einer Interaktion abgefragt, so werden nun nur noch die kompletten Städteverbindungen abgefragt. Möglich ist dieser Schritt, da bei der Darstellung lediglich die aggregierten Zeitintervalle zwischen Städten von Bedeutung sind.

Abbildung 4.4 zeigt exemplarisch den Unterschied zwischen der Selektion eines einzelnen Zeitintervalls und der Selektion einer kompletten Verbindung. Zusätzlich zu den aggregierten Zeitintervallen sind auch die zugehörigen Nutzer von Bedeutung, da ansonsten eine eindeutige Zuordnung nicht möglich ist. Diese Daten werden zur Laufzeit des Programms vorgehalten. Alles andere, wie Textinformation oder spezifische Tweets, werden auf Anfrage mit Hilfe von verschiedenen Datenbank-Indices geladen. Es wäre effizienter, bei einer Selektion die entsprechenden Daten bereits im Speicher vorzuhalten, jedoch würde sich somit auch der initiale Start des Programms spürbar verzögern, bis alle Daten aggregiert sind.

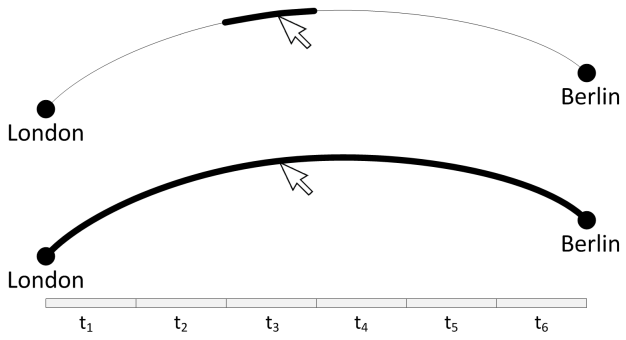

Abbildung 4.4: Vergleich der Selektion einer Verbindung zwischen zwei Städten

Für die Bereitstellung und die Weitergabe von Änderungen dient das Beobachter-Entwurfsmuster [KS09]. Es gibt verschiedene Arten der Umsetzung, das Grundprinzip ist jedoch immer gleich. Aus dem Namen leitet sich auch der Gebrauch ab: Dieses Entwurfsmuster baut eine Beziehung zwischen einem Subjekt und seinen Beobachtern auf. Das Subjekt bezeichnet dabei das Objekt, dessen Zustandsänderung für andere beobachtende Objekte interessant ist. Im Rahmen dieser Arbeit gibt es mehrere Subjekte, für die Bereitstellung der Daten gibt es lediglich ein Objekt, das die Städteverbindungen in Form von konkatenierten Zeitintervallen und allen notwendigen Zusatzinformationen hält. Ändern sich die Daten, werden alle Beobachter automatisch benachrichtigt. Die Daten, die direkt aus der Datenbank vorgehalten werden, wie es hier der Fall ist, sind auf die verschiedenen Vergrößerungsstufen abgebildet. Gibt es einen Datensprung von einer Vergrößerungsstufe zur anderen, wie es beim Semantischen Zoom der Fall ist, so werden die Daten über das Beobachter-Muster an alle notwendigen Beobachter propagiert.

4.4.2 Datenfilterung

Der Analyst hat mehrere Möglichkeiten, wie er die dargestellten Daten filtern kann. Im Folgenden werden alle Möglichkeiten vorgestellt. Die Datenfilterung stellt einen wichtigen Bereich der Informationsvisualisierung dar und kommt hauptsächlich bei der Exploration von Daten zum Einsatz, wenn der Überblick bereits gegeben ist. Egal welcher Filter gesetzt wird, es ändert sich immer die Textaggregation, die für Einzelverbindungen angezeigt wird. Ferner können mehrere Filter nacheinander angewendet und entfernt werden. Dazu werden die Filter in der Reihenfolge, wie sie gesetzt wurden, logisch UND-verknüpft. Dies geht daraus hervor, dass durch die sukzessive Auswahl von Filtern der Datensatz eingeschränkt werden soll. Wird ein Filter entfernt, so werden die Daten für die restlich gesetzten Filter in derselben Reihenfolge erneut gefiltert. Ein Beispiel: Auf den Datensatz werden ein Zeitfilter, ein Kartenfilter und ein Benutzerfilter angewendet. Entschließt sich der Analyst dazu, den Kartenfilter zu entfernen, so wird in diesem Fall der Benutzerfilter auf den Datensatz nach der Zeitfilterung angewendet. Es ergibt sich folglich die Filterreihenfolge: Zeitfilter und dann Benutzerfilter.

Das Setzen eines Filters kann sich auf mehrere Darstellungen auswirken. Eine Kartenselektion sorgt beispielsweise dafür, dass die Zeit oder die beteiligten Benutzer neu aufgelöst werden, ohne dass diese Filter explizit gesetzt werden. Sie stehen in einem semantischen Zusammenhang zueinander. Für die Propagierung der Daten kommt, wie bereits ansatzweise in Kapitel 4.4.1 beschrieben, das Beobachter-Muster zum Einsatz. Die Daten werden in Form von Listen in Objekten gehalten; sollte sich die Liste der visualisierten Verbindungen in irgendeiner Form durch Setzen eines Filters ändern, so wird dies an alle Beobachter propagiert.

4.4.2.1 Textfilter

Um die Trajektorien textuell zu beschränken, wird ein Textfilter eingeführt. Werden verschiedene Schlagwörter zur Suche eingegeben, so wird nach Tweets gesucht, die alle Schlagwörter enthalten, auch nach jenen, die bei Annotation von der Trajektorie ausgeschlossen wurden. Diese Tweets werden in der Textsuche miteinbezogen, da hier alles von Interesse ist, was in Bezug zur angetretenen Reise steht. Dazu gehören eventuell auch Vorbereitungen einige Tage zuvor. Als Ergebnis der Anfrage gibt es eine Liste aller Benutzer, welche passende Tweets veröffentlicht haben. Wenn die Benutzer bekannt sind, wird über alle visualisierten Verbindungen zwischen Städten iteriert – wohlgemerkt nicht die Zeitintervalle – und es werden nur noch jene Verbindungen angezeigt, die die jeweiligen Benutzer der Anfrage enthalten.

4.4.2.2 Zeitfilter

Der Datensatz ist durch die aggregierten Zeitintervalle auf den Zeitfilter zugeschnitten. Die Zeit wird in den Schritten, die durch die Zeitintervalle festgelegt sind, gefiltert. Eine andere Zeitfilterung in anderen Einheiten ist somit nicht möglich. Für eine Zeitfilterung werden alle Zeitintervalle einer Verbindung zwischen Städten betrachtet. Wird die Zeit auf einer Verbindung mit mehreren Benutzern eingeschränkt, so werden die kürzeren Reisen auf dieser Verbindung relevanter als die längeren, falls die längeren nicht mehr komplett im Zeitfilter enthalten sind. Dieser Effekt wird durch die Gewichtung der einzelnen Zeitintervalle auf einer Verbindung zwischen zwei Städten erreicht; denn wird die Zeit eingeschränkt, so werden der längeren Verbindung Gewichte abgezogen, während die kurze Verbindung noch voll repräsentiert ist.

Es ist hier von mehreren Verbindungen zwischen zwei Städten die Rede, jedoch ist genau eine aggregiert, da die Gewichte der verschiedenen Verbindungen jeweils auf ein Zeitintervall aufaddiert werden und somit ein Intervall jeweils die Gesamtinformation enthält (siehe Abschnitt 4.3.2.2).

Mit dem Zeitfilter ist eine Abbildung auf Farbe verbunden. Die gewählte Farbpalette ist eine Interpolation von Schwarz über Gelb zu Rot, wobei Schwarz für den niedrigsten und Rot für den höchsten Wert steht. Diese Farbabbildung hat zum Ziel, eine Unterscheidung zwischen den Verbindungen tätigen zu können. So wird das aufaddierte Gewicht der selektierten

Zeitintervalle ins Verhältnis zu dem aufaddierten Gesamtgewicht der kompletten Verbindung über alle Zeitintervalle gesetzt. Das Ergebnis wird auf die entsprechende Farbe abgebildet. Befindet sich die komplette Verbindung im gefilterten Zeitbereich, so ist sie Rot; je weniger von der Verbindung durch die Filterung übrig bleibt, umso mehr geht die Farbe gegen Schwarz, bis sie ganz verschwindet.

4.4.2.3 Userfilter

Eine weitere Möglichkeit, die dargestellten Daten einzuschränken, ist über die Anzahl der Benutzer. Dabei wird zwischen verschiedenen Filterarten unterschieden. Die erste Art beruht auf den bekannten Benutzerdaten: Da die Daten anonymisiert sind, lässt sich der Datensatz entweder über die eindeutige ID oder über den verwendeten Benutzernamen filtern. Dies geschieht über die Auswahl der Zeichenketten anhand von Präfixen. Anonymisiert bedeutet hier, dass keine Anschrift des Benutzers verfügbar ist, oder er durch den Benutzernamen nicht eindeutig identifiziert werden kann. Angenommen es gibt zwei Benutzer: Benutzer A mit der ID „123456“ und Benuter B mit der ID „123478“. Filtert der Analyst nach „123“, sind beide Benutzer in der Menge des Datensatzes enthalten, während bei einer Filterung nach „12345“ nur Benutzer A angezeigt wird.

Um die Überdeckung von Verbindungen zu reduzieren, ist es zweckdienlich den Datensatz nicht nur durch Ähnlichkeit von Zeichenketten, sondern darüber hinaus mengenmäßig einzuschränken. Daraus ergibt sich die zweite Möglichkeit, wie nach Benutzern gefiltert werden kann: Durch Einschränken der Anzahl der Benutzer wird folglich auch der dargestellte Datensatz eingeschränkt. Eine beliebige Anzahl an Benutzern kann ausgewählt und zahlenmäßig dargestellt werden. Diese Art des Filterns eignet sich besonders für sehr große Datensätze. Wie bei einem Paging-Verfahren können sukzessiv die nächsten Daten dargestellt werden. Problematisch ist der Informationsverlust, der dadurch eintritt. Ein mögliches Szenario hierfür ist die Exploration eines bestimmten Ereignisses – wie beispielsweise ein Ärztekongress – durch einen Analysten. Exploriert der Analyst immer nur Teildatensätze, so bewegt er sich folglich auch immer nur auf Teilinformationen. Der Effekt des gewünschten Gesamtüberblicks der aggregierten Information ist so nicht mehr gegeben. In so einem Fall eignet sich das Setzen eines Textfilters, da hier die Exploration ereignisorientiert stattfindet. Der Nachteil hierbei ist jedoch, dass eventuell Benutzer aus dem Datensatz ausgeschlossen werden, weil nicht mit Hilfe der passenden Stichworte gesucht wurde. Welches Vorgehen zweckdienlicher ist, liegt in diesem Fall deutlich in der Hand des Analysten.

4.4.2.4 Kartenfilter

Unter die Kategorie Kartenfilter fallen mehrere verschiedene Filter. Der Analyst wird durch verschiedene Interaktionen mit der Karte bei seiner Analyse unterstützt. Dazu gehören die Selektion von einzelnen Verbindungen, die Auswahl von bestimmten Verbindungen sowie die Auswahl von interessanten Plätzen.

Einzelselektion von Verbindungen und Benutzern

Möchte man eine Einzelverbindung näher betrachten, so wird die Möglichkeit geboten, diese auszuwählen. Die Auswahl kann entweder über die direkte Selektion mittels der Maus oder über das Auswählen des entsprechenden Benutzers getätigt werden. Wird eine Selektion einer Verbindung über die Maus getätigt, werden die entsprechenden Verbindungen hervorgehoben und gleichzeitig die zugehörigen Benutzer mit ihren vollständigen Trajektorien und Zusatzinformationen angezeigt und visualisiert. Zu den Zusatzinformationen gehören alle veröffentlichten Tweets sowie die aggregierten Textinformationen in Form einer nach Relevanz sortierten Schlagwortwolke, die zu den Benutzern auf der Verbindung gehören.

Die Selektion einzelner Benutzer kann entweder auf die Selektion einer einzelnen Verbindung folgen, um den Informationsgehalt einzuschränken, oder initial erfolgen. Bei einer Benutzerselektion werden nur noch jene aggregierten Verbindungen angezeigt, auf denen die Benutzer unterwegs waren. Zusätzlich werden auch relevante textuelle Informationen in aggregierter Form sowie alle getätigten Tweets dargestellt.

Rechteckselektion von Verbindungen

Um die Anzahl der Informationen und Benutzer des Datensatzes einzuschränken, gibt es die Möglichkeit der ODER-verknüpften Selektion durch Auswahlrechtecke. Dabei werden Rechtecke über Verbindungen oder über die Städte, die den Start oder das Ende einer Verbindung repräsentieren, gezogen. Alle enthaltenden Verbindungen werden hervorgehoben, alle anderen aus dem Datensatz ausgeschlossen. Zusätzlich wird die Liste der Benutzer auf die Selektion angepasst. Werden Verbindungen durch mehrere Rechtecke selektiert, so werden die Ergebnisse logisch vereinigt.

Rechteckselektion von Plätzen

Angelehnt an die Rechteckselektion von Verbindungen können auch jegliche Städte oder Plätze ausgewählt werden, die nicht durch eine unmittelbare Verbindung repräsentiert werden. Das bedeutet, dass nach allen Benutzern und den somit verbundenen Verbindungen gesucht bzw. gefiltert wird. Falls der Analyst beispielsweise ein Rechteck um Kenya zieht, wo in diesem Beispiel keinerlei Verbindungen ihren Ursprung oder ihr Ende haben, so werden alle in der Datenbank abgelegten Tweets nach diesen Geokoordinaten durchsucht. Als Ergebnis werden alle Verbindungen angezeigt, deren Benutzer einen der gefundenen Tweets veröffentlicht hat.

4.4.3 Analyse von Trajektorien

Abgesehen von den Filtermöglichkeiten werden die Selektionen durch die aggregierten Informationen angereichert, durch welche eine Analyse zusätzlich unterstützt wird. Es

kommen hier zwei verschiedene Konzepte zum Einsatz: Die Analyse durch aggregierte textuelle Informationen und die Analyse durch die Verfolgung von Benutzern.

4.4.3.1 Textuelle Analyse

Bei der textuellen Analyse wird die Selektion auf die aggregierte textuelle Information abgebildet. Es gilt dabei, zwischen zwei verschiedenen Abbildungsarten zu unterscheiden: die benutzerspezifische und die selektionsspezifische Abbildung. Bei der benutzerspezifischen wird die aggregierte textuelle Information aller Verbindungen dargestellt, wohingegen bei der selektionsspezifischen nur die Information der ausgewählten Verbindung angezeigt wird. Auch die Darstellung ist unterschiedlich. Bei der Auswahl von Benutzern wird separat die aggregierte Textinformation, absteigend sortiert nach Relevanz, angezeigt.

Abbildung 4.5: Darstellung einer verbindungsspezifischen Schlagwortwolke in linearer Form, sortiert nach Relevanz

Wird jedoch eine einzelne Verbindung selektiert, so wird eine verbindungsspezifische Schlagwortwolke – ebenfalls absteigend nach Relevanz sortiert – direkt auf der Verbindung angezeigt. Abbildung 4.5 zeigt diesen Vorgang exemplarisch. Der Term mit der höchsten Gewichtung wird auf die Mitte der Verbindung abgebildet; danach alle anderen Terme in absteigender Gewichtung, abwechselnd links und rechts davon, bis die Verbindung vollständig gefüllt ist. Die Exploration wird durch diese Darstellung zusätzlich unterstützt, da aggregierte Informationen der Tweets unmittelbar auf der zugehörigen Verbindung angezeigt werden. Bei Abbildung 4.5 wird eine Reise von Berlin nach London repräsentiert. Beide Orte sowie der Reisetag Montag erscheinen in der Schlagwortwolke. Des Weiteren scheint am Flughafen eine Verspätung durch einen Streik verursacht worden zu sein.

Diese Schlagwortwolke verändert sich, wie bereits in Kapitel 4.4.2 beschrieben, mit dem Setzen von Filtern. Die Zeitkomponente hängt mit allen verwendeten Filtern zusammen und wird bei jeweiliger Filterung neu aufgelöst. Die Zeitkomponente ist die Abbildung der Zeitintervalle mit anhängenden Gewichten, welche auch an die Terme gebunden sind. Wird die Zeit gefiltert oder durch Setzen eines anderen Filters neu aufgelöst, so verändert sich auch die Relevanz der dargestellten Terme.

4.4.3.2 Analyse der Benutzerbewegung

Die textuelle Analyse ist bereits eine sehr gute Möglichkeit, um Informationen aus Trajektorien und die damit verbundenen Verbindungen zu extrahieren. Es gibt weitere Konzepte, wie die Exploration von Benutzern umgesetzt werden kann, aus welchen hier passende vorgestellt werden.

Angefangen auf der niedrigsten Abstraktionsebene, den Verbindungen zwischen einzelnen Städten, kann mit Hilfe eines Pfeils auf der Linie festgelegt werden, für welche Richtung man die Benutzer und damit verbundenen Trajektorien und Schlagwortwolke angezeigt bekommen möchte. Zusätzlich wird bei einer Selektion, wie bereits erwähnt, die Zeitkomponente neu aufgelöst und eine extra Ansicht für den zeitlichen Vergleich von einzelnen Tweets verschiedener Benutzer eingeblendet. Es gibt zwei Möglichkeiten: Entweder können alle Benutzer auf einer Zeitachse anzeigt werden, oder auf getrennten Zeitachsen. Bei der Anzeige auf einer Zeitachse werden für eine eindeutige Zuordnung die jeweiligen Benutzer über den HSV-Farbraum farbkodiert. Die Berechnung der Position einzelner Tweets auf der Zeitachse geschieht relativ zum berechneten Gesamtzeitraum eines Benutzers.

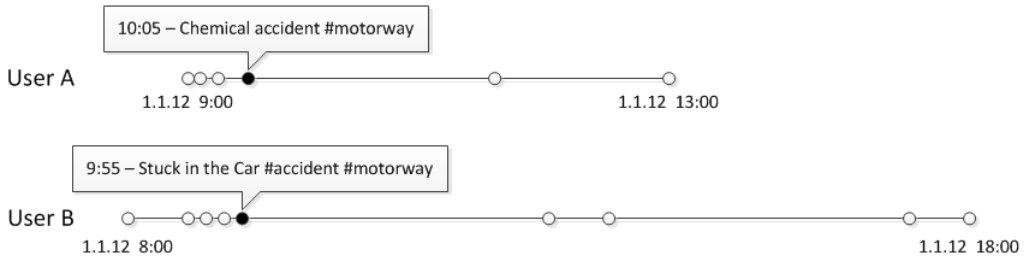

Abbildung 4.6: Zeitachsen-Visualisierung mehrerer Benutzer

Abbildung 4.6 zeigt am Beispiel zweier Benutzer, wie dieses Konzept in einer Visualisierung umgesetzt wird. Auf jeder Zeitachse werden in zeitlich korrekten Abständen die Tweets relativ zueinander positioniert. Angenommen beide Nutzer waren auf derselben Route unterwegs, lassen sich so Gemeinsamkeiten finden. Bei mehreren Benutzern können durch Anhäufungen von Tweets zur selben Zeit Ereignisse entdeckt werden. Diese Art der Visualisierung wird immer aktiv, wenn die Benutzerliste eingeschränkt wird, sei es durch eine Selektion oder einen Kartenfilter. Auch angetretene Reisen können erkannt werden, wenn auf einmal das Veröffentlichungsmuster gebrochen wird.

5 Umsetzung

Dieses Kapitel gibt einen Überblick über die Umsetzung der Hauptkomponenten, die in Kapitel 4 vorgestellt wurden. Dieser Teil der Diplomarbeit ist gegliedert in zwei Hauptteile: die Datenbankstruktur in Abschnitt 5.1 und die Benutzeroberfläche in Abschnitt 5.2. Im Teil der Datenbankstruktur wird auf generelle Konzepte eingegangen, welche für die Datenhaltung erforderlich sind. Im Teil der Benutzeroberfläche wird nicht thematisch, sondern nach Oberflächenkomponenten getrennt beschrieben, wie die Konzepte umgesetzt werden.

5.1 Datenbankstruktur

Die Umsetzung der vorgestellten Datenbankstruktur (siehe Kapitel 4) erfolgt über eine PostgreSQL[1] Datenbank. Laut Herstellerangaben ist PostgreSQL ein sehr leistungsstarkes, quelloffenes und objekt-relationales Datenbanksystem. Die Größe der Datenbank ist lediglich durch den verfügbaren Speicher begrenzt. Zusätzlich gibt es interessante Erweiterungen, wie PostGIS, was durch die Umsetzung des GiST (Generalized Search Tree) als Datenbank für Geoinformationssysteme dienen kann. Auch die Volltextsuche wird durch spezielle Strukturen und Indices unterstützt.

Bei der Umsetzung der Datenbankstruktur kommen verschiedene Indices zum Einsatz: Hash, B-Baum und GIN. Ein Index ist eine extra angelegte Datenstruktur, welche eine schnellere Suche in der Datenbank ermöglicht. Besonders geeignet sind solche Strukturen für Datenbanken mit großen Datenmengen, wie es bei dieser Diplomarbeit der Fall ist.

Listing 5.1 Erstellung eines Hash-Index in PostgreSQL

```
CREATE INDEX userid index
ON user
USING hash (user_id);
```

Listing 5.1 zeigt exemplarisch das Erstellen eines Hash-Index für die Spalte *user_id* in der Tabelle *user*. Da die Suche ohne Index in einer Datenbank im Bereich $\mathcal{O}(n)$ bei n zu durchsuchenden Zeilen liegt und somit vor allem bei großen Datenmengen zu langsam ist, bedarf es dieser verschiedenen Indices.

[1] PostgreSQL: http://www.postgresql.org/

Hash

Der Hash-Index [PD08] wird hier für Tabellenspalten verwendet, die einen Zahlenwert beinhalten. Mit dieser Methode kann annäherungsweise eine Laufzeitkomplexität von $\mathcal{O}(1)$ erreicht werden. Zurückzuführen ist die Laufzeit auf die Idee hinter dem Hashing. Dabei wird bereits beim Speichern dafür Sorge getragen, dass die Objekte effizient mit Hilfe eines Suchschlüssels gefunden werden können. Mittels einer Hash-Funktion wird der eigentliche Wert auf eine Position in der Tabelle abgebildet, was wiederum bedeutet, dass bei einer Suche sehr schnell diese Position gefunden wird. Die genaue Funktionsweise wird hier nicht weiter erläutert.

B-Baum

Möchte man nicht nach Gleichheit innerhalb der gespeicherten Daten suchen, sondern innerhalb gewisser Bereiche, so bietet sich die Struktur mittels eines B-Baum-Index an [Gra11]. Ein B-Baum ist ein ausbalancierter Baum, welcher Daten sortiert nach Schlüssel speichert. Die Laufzeitkomplexität bei einer Suche liegt bei $\mathcal{O}(long(n))$.

GIN

Mit Hilfe des GIN-Index [pos] kann die Volltextsuche verbessert werden. Verwendet wird dieser Index in Kombination mit tsvector-Werten. Ein tsvector-Wert ist eine geordnete Liste, welche verschiedene Lexeme enthält. Dies sind normalisierte Wörter, d.h. verschiedene Varianten des Wortes sind auf dasselbe Wort zurückzuführen. Die Sortierung und Entfernung von Duplikaten werden vollautomatisch ausgeführt. Die Suche auf diesen Daten erfolgt via tsquery.

Tabelle 5.1 zeigt die in der implementierten Datenbank eingesetzten Indices.

Tabelle	Feld	Verwendeter Index
Tweet	text	GIN-Index
	user_id	Hash-Index
User	user_id	Hash-Index
UserMovement	user_id	Hash-Index
	conn_id	Hash-Index
Connection	conn_id	Hash-Index
TermMapping	term_id	Hash-Index
	conn_id	Hash-Index
Term	term	B-Baum
	term_id	Hash-Index

Tabelle 5.1: In der Datenbank eingesetzte Indices

Diese Indices resultieren aus den verschiedenen Suchanfragen, die während der Laufzeit des Programms getätigt werden.

5.2 Benutzeroberfläche

Dieses Kapitel widmet sich der Darstellung und Umsetzung der Konzepte in Form eines Java-Programms. Die Aufteilung erfolgt nach den einzelnen Komponenten, die auf der Benutzeroberfläche repräsentiert sind. Es wird zwischen zwei Benutzeroberflächen unterschieden: zum einen die Oberfläche, die zur Aggregation der Daten im Rahmen der Vorverarbeitung verwendet wird und zum anderen die Oberfläche, welche der Visualisierung und der Umsetzung der visuellen Analyse dient. Bevor auf die einzelnen Komponenten eingegangen wird und darauf, welche Algorithmen gegebenenfalls dahinter stecken, wird ein kurzer Überblick über beide Benutzeroberflächen gegeben.

5.2.1 Überblick

Es gibt zwei verschiedene Benutzeroberflächen: eine für die Aggregation und eine für die Darstellung der Daten. Abbildung 5.1 zeigt erstere.

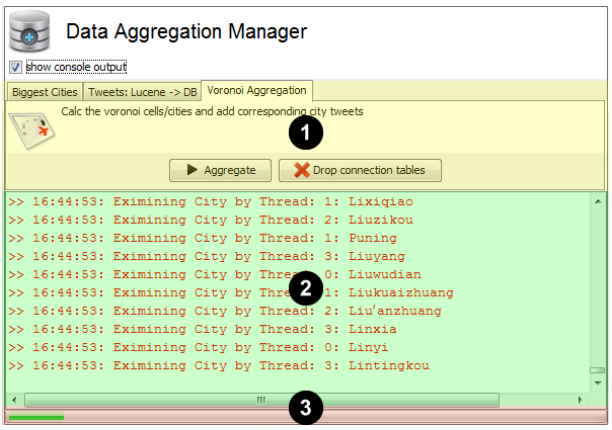

Abbildung 5.1: Benutzeroberfläche für die Datenaggregation

Die Oberfläche für die Aggregation der Daten besteht aus drei verschiedenen Bereichen.

1 Auswahl der verschiedenen **Aggregationsarten**. Jeder Reiter ist gleich aufgebaut und enthält jeweils eine Kurzbeschreibung darüber, was verarbeitet wird sowie entsprechende Buttons, um die Aktionen zu koordinieren. Der erste und zweite Reiter dienen der Verarbeitung und Vorbereitung der notwendigen Daten. Dazu gehört das Ablegen der Städte, Drehkreuze und Tweets sowie deren Benutzerprofile in der Datenbank. Auf Basis dieser Daten findet im dritten Reiter die eigentliche Aggregation statt, welche die Daten in ein effizientes Datenformat speichert und für die Visualisierung vorbeireitet.

2 Eine **Konsole**, die den Benutzer auf dem Laufenden hält und zeigt, was im Moment bearbeitet wird.

3 Ein **Fortschrittsbalken**, der anzeigt, wie weit die Bearbeitung bereits fortgeschritten ist.

Wurden alle notwendigen Daten vorverarbeitet, lassen sie sich visualisieren. Für diese Aufgabe dient die zweite Oberfläche, die in Abbildung 5.2 abgebildet ist.

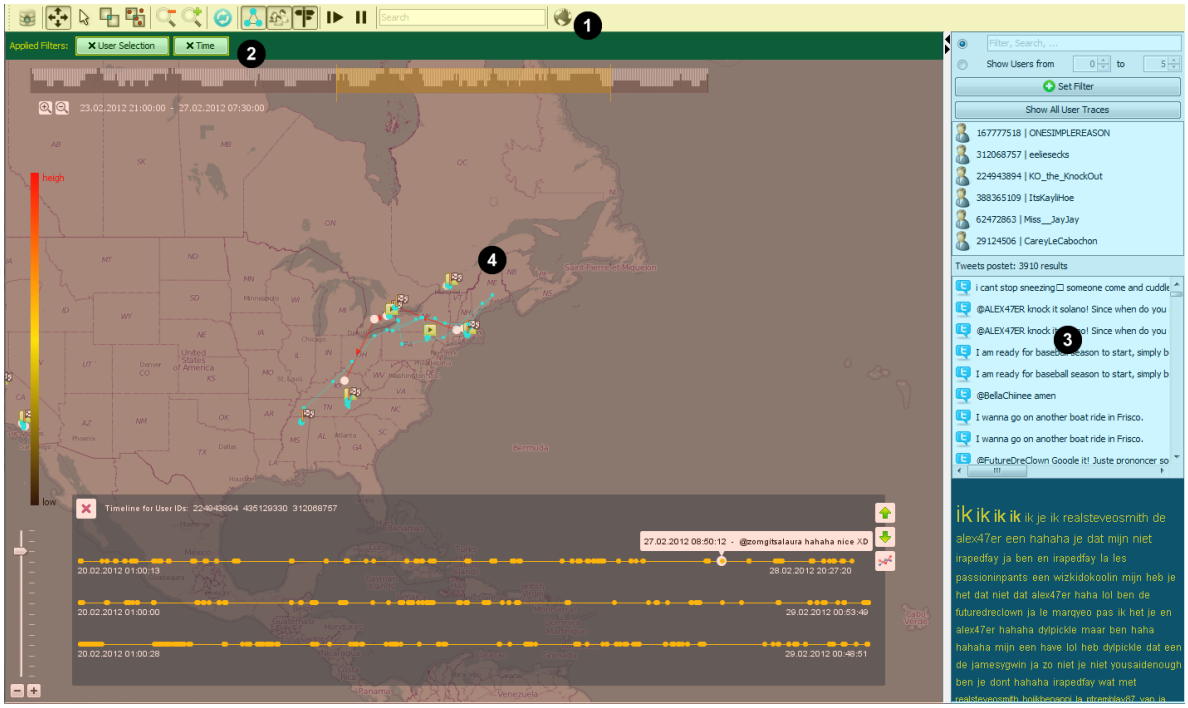

Abbildung 5.2: Übersicht über die Visualisierung der Ergebnisse

Aufgeteilt ist die Oberfläche zur Visualisierung in vier verschiedene Hauptkomponenten, welche jeweils einige Funktionen beinhalten:

1 Die **Werkzeugleiste**, welche verschiedene Optionen bietet. Diese Optionen sind in Gruppen unterteilt. Dazu gehören: Interaktionen mit der Karte, Anzeigeoptionen für die Karte sowie Filtermöglichkeiten.

2 Die **Filterverwaltung**. Hier werden alle gesetzten Filter angezeigt und können in beliebiger Reihenfolge auch wieder entfernt werden.

3 Die **Detailansicht**. Hier werden alle Benutzer und deren Tweets aufgelistet. Die aus den Tweets entstandene Schlagwortwolke wird ebenfalls dargestellt. Zusätzlich bietet diese Ansicht verschiedene Filteroptionen für die Benutzer an.

4 Die **Karte**. Sie ist das Herzstück der Oberfläche und visualisiert die aggregierten Ergebnisse der Vorverarbeitung. Zusätzlich beinhaltet sie Komponenten wie den Zeitfilter mit integriertem Histogramm und die Timeline-Anzeige.

5.2.2 Aggregation

Im Folgenden wird die Realisierung der Konzepte erläutert, die bei der Aggregation zum Einsatz kommen. Für die Umsetzung spielt die Nebenläufigkeit bei Ausführung der Algorithmen eine entscheidende Rolle, da sie die Aggregation durch eine parallele Ausführung bedeutend beschleunigen kann. Aus diesem Grund wird als Erstes die Realisierung der parallelen Ausführung vorgestellt. Darauffolgend wird auf Basis der Nebenläufigkeit dargestellt, wie die eigentliche Aggregation realisiert wird. Aufgeteilt ist die Umsetzung der Aggregation in zwei Hauptteile: die Aggregation der erforderlichen Daten und die Aggregation der Voronoi-Zellen auf der Basis der aggregierten erforderlichen Daten.

5.2.2.1 Nebenläufigkeit

In Kapitel 2.2.1.2 wurde bereits beschrieben, dass der Umfang der Daten immens ist. Diese Datenflut benötigt im Hintergrund spezielle Strukturen, um sie effizient verarbeiten zu können. Eine Möglichkeit, die hier verwendet wird, ist die Nebenläufigkeit im Programm über sogenannte Threads (oder auch Teil eines Prozesses genannt).

Die Realisierung der Nebenläufigkeit erfolgt über einen Thread Pool[2]. Ein Thread Pool besteht aus sogenannten Worker Threads, die er verwaltet. Um die zugrundeliegende Hardware optimal zu nutzen, wird pro vorhandenem Prozessor ein separater Thread erzeugt. Abbildung 5.3 zeigt die Architektur dahinter.

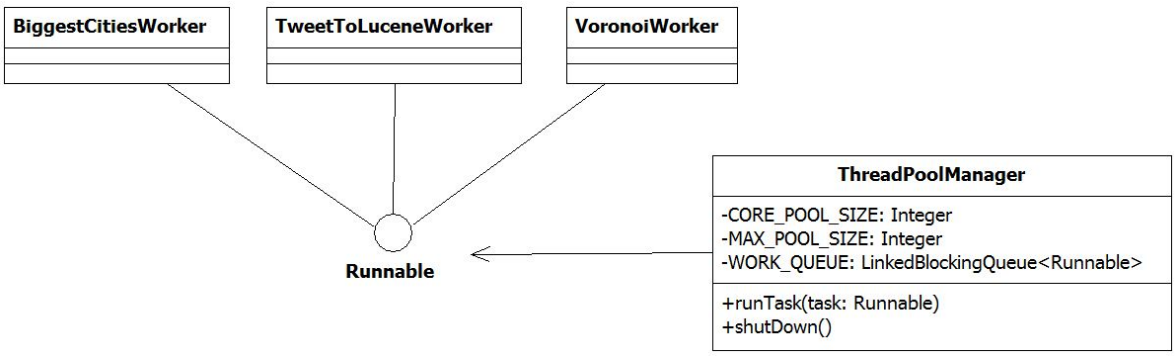

Abbildung 5.3: Klassendiagramm für den implementierten Thread Pool

[2]Thread Pools: http://docs.oracle.com/javase/tutorial/essential/concurrency/pools.html

Es gibt eine zentrale Klasse **ThreadPoolManager**, die die Threads verwaltet und für deren Ausführung sorgt. Zusätzlich gibt es das Java-eigene Interface **Runnable**, welches von der ausführenden Klasse implementiert werden muss. (Diese Klassen enden alle auf **Worker**.) Für jeden verfügbaren Prozessor wird ein **Worker** dem **ThreadPoolManager** über die Methode **runTask(...)** übergeben und ausgeführt.

Es bieten sich verschiedene Möglichkeiten für die Übertragung der Tweets aus den Lucene Repositories in die Datenbank an. Man kann z. B. für jedes Repository einen Thread starten. Es ist jedoch effizienter, alle Inhalte sukzessiv aufzuteilen, da so die Anzahl der Threads nicht von der Anzahl der Repositories abhängig ist. Dazu werden alle IDs der Tweets, die in den Repositories enthalten sind, in eine Liste abgespeichert. Der Tweet lässt sich über die eindeutige ID laden. Diese Liste wird parallel abgearbeitet bis alle Tweets übertragen wurden. Dazu muss eine Thread-sichere, globale Liste verwendet werden. Das heißt, dass alle ausgeführten Threads auf derselben Basis arbeiten und sich dabei nicht in die Quere kommen. Dieses Verfahren wird in dieser Arbeit für alle parallelen Tätigkeiten verwendet.

5.2.2.2 Aggregation der erforderlichen Daten

Auf Basis der parallelen Abarbeitung von Daten können nun die Daten aggregiert werden, die für die kontextbezogene Annotation und Aggregation via Voronoi-Zellen notwendig sind. Dazu gehören zum einen die relevanten Städte und Drehkreuze, zum anderen die erforderlichen Tweets mit zugehörigen Benutzern. Die Aggregation orientiert sich an der vorgegebenen Datenbankstruktur.

Im ersten Teil der Aggregation liegen die Städte sowie die Drehkreuze im .csv oder einem ähnlichen Format vor. Mittels einem Parser werden die relevanten Daten daraus extrahiert und in die Datenbank übertragen. Dazu gehören die ID der Stadt oder des Drehkreuzes, der Name, die exakten Geokoordinaten und die Population oder bei den Drehkreuzen der Passagierdurchlauf. Für die Städte gilt eine gewisse Einschränkung: Es werden diejenigen Städte übertragen, die eine Population von mehr als 1000 Einwohnern vorweisen. Dies wurde bereits in Kapitel 4 beschrieben, sodass ein Semantischer Zoom realisiert werden kann.

Im zweiten Teil werden die Tweets aus dem Lucene-Repository geladen und in das vorhandene Datenbankschema übertragen. Die Benutzerprofile werden in der Datenbank jedoch in eine separate Tabelle abgelegt und der eigentliche Tweet enthält einen Fremdschlüssel mit der Benutzer ID. Dies hat den Vorteil, dass auf einzelne Benutzerprofile viel schneller zugegriffen werden kann. Auf Basis der Benutzerprofile und den damit verbundenen Trajektorien findet im nachfolgenden Schritt die Voronoi-Zellen-Aggregation statt.

5.2.2.3 Aggregation der Voronoi-Zellen

Die Aggregation der Voronoi-Zellen beschreibt die Übertragung der Daten in eine Struktur, welche effizient repräsentiert werden kann. Diese wurde bereits in Kapitel 4 vorgestellt. Jegliche Interaktion des Analysten mit den Daten basiert auf dieser Datenstruktur.

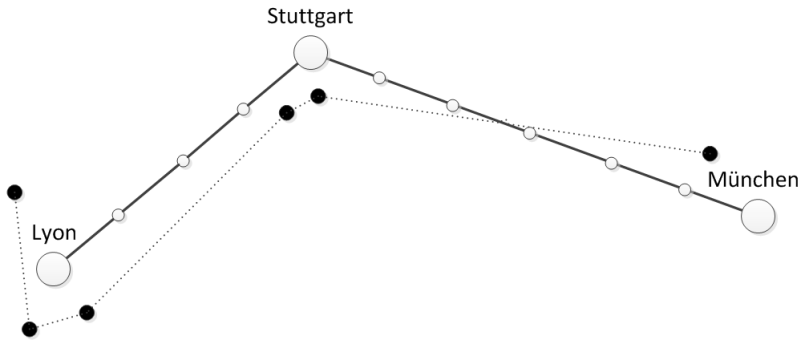

Abbildung 5.4: Beispiel der Aggregation einer Benutzertrajektorie

Abbildung 5.4 zeigt beispielhaft, wie eine Trajektorie mit zugehörigen Kontextinformationen auf Voronoi-Städte abgebildet wird. Die gestrichelte Linie ist der Originalweg des Benutzers. Seine Aufenthaltsorte mit zugehörigen, textuellen Inhalten werden immer der nächstgelegenen Voronoi-Stadt zugeteilt. Der Zeitraum, in welchem eine Verbindung stattfindet, startet mit dem letzten Tweet in einer Voronoi-Zelle und endet mit dem ersten Tweet nach Übertritt in eine andere Voronoi-Zelle. Der Zeitraum einer Verbindung wird in Zeitintervalle fester Länge unterteilt. Nehmen wir an, der Weg von Lyon nach Stuttgart dauert zwei Stunden, der Weg von Stuttgart nach München drei Stunden und ein Zeitintervall ist dreißig Minuten lang, so ergeben sich insgesamt zehn Zeitintervalle; vier für den Weg von Lyon nach Stuttgart und sechs für den Weg von Stuttgart nach München. Im Folgenden wird dieser Vorgang der Aggregation sowie die Übertragung der Daten in die Datenbank näher beschrieben.

Es gibt verschiedene Möglichkeiten die Trajektorien zu aggregieren; hier wurde aber bewusst die Variante über Voronoi-Zellen ausgewählt (siehe Kapitel 4.3.2). Der Algorithmus 5.1 zeigt exemplarisch in Pseudocode das Vorgehen der Aggregation. Der hier gezeigte Algorithmus ist stark gekürzt, sodass die relevanten Teile besser zum Vorschein kommen.

Ausgangssituation ist eine Liste mit allen vorhandenen Benutzern. Wie bereits in Kapitel 5.2.2.1 beschrieben, wird diese Liste parallel von mehreren Threads abgearbeitet, bis keine Benutzerprofile zum Abarbeiten mehr vorhanden sind. Im vorgestellten Algorithmus wird über jedes enthaltene Benutzerprofil iteriert und direkt zur Laufzeit aus der Datenbank all seine im zu betrachtenden Zeitraum veröffentlichten Tweets geladen. Da sichergestellt werden soll, dass der ausgewählte Benutzer wirklich gereist ist, wird gefordert, dass er mindestens zwei Tweets veröffentlicht hat. Falls dem so ist, geht es in der Bearbeitung weiter. Durch die Realisierung des Semantischen Zooms müssen alle relevanten Vergrößerungsstufen der Karte betrachtet werden, in denen eine andere Anzahl an Städten visualisiert wird. Dementsprechend müssen die Tweets für jede relevante Stufe neu aggregiert werden. Hierzu wird über alle Vergrößerungsstufen und deren Städte iteriert. Im ersten Schritt werden die veröffentlichten Tweets über die Distanzberechnung mittels des vorgestellten Haversine der nächstgelegenen Stadt aus der Liste der Städte, zugehörig zur Vergrößerungsstufe, zugeteilt. Falls am Ende dieser Berechnung daraus mindestens zwei verschiedene Städte

Algorithmus 5.1 Algorithmus zur Bestimmung der aggregierten Trajektorien

```
for all u ∈ users do
    trajectory ← hole alle Tweets für User u aus der Datenbank
    sortiereNachZeit(trajectory)
    if mehr als 2 Tweets in trajectory then
        if mehr als 1 Tweet pro Stunde veröffentlicht then
            Wahrscheinlich Spam, Rest der Schleife überspringen
        end if
        for all zs ∈ zoomStages do
            Erstelle Liste mit allen besuchten Voronoi-Städten, die in der Zoomstufe
            zs enthalten sind
            Dazu berechne mittels haversine die kleinste Distanz
            if mehr als 1 Stadt wurde besucht then
                if Der Benutzer reist schneller als 1000km pro Stunde then
                    Wahrscheinlich Spam, Rest der Schleife überspringen
                else
                    Erstelle Zeitintervalle. Start ist Datum des letzten vor der Reise
                    veröffentlichten Tweets. Ende das des ersten Tweets in der Zielstadt.
                    Termfrequenz aus den Tweets des Abreise und Ankuftstags einer
                    Verbindung zwischen zwei Städten berechnen und zusätzlich mit
                    Intervallgewichtung multiplizieren.
                end if
            end if
        end for
    end if
end for
```

resultieren, geht es mit der Aggregation weiter; ansonsten wird abgebrochen und mit dem nachfolgenden Benutzerprofil weitergemacht. Nun kommt ein zweiter Filter zum Einsatz, welcher Spam ausschließen soll. Dazu wird der zeitliche Unterschied zwischen den Voronoi-Städten betrachtet. Sind zwei Städte in einem Veröffentlichungszeitraum von einer Stunde mehr als 1000 Kilometer voneinander entfernt, so handelt es sich in vielen Fällen um einen Bot. Falls es kein Bot ist, so werden die Zeitintervalle über die Reise bestimmt sowie die Termfrequenz für die zugehörigen Tweets berechnet. Die Terme, die mit Hilfe des tf-Maß bestimmt werden, werden zusätzlich noch mit dem Gewicht des zugehörigen Zeitintervalls multipliziert. Damit wird die Relevanz einer Reise gespiegelt, falls mehrere Benutzer auf dasselbe Zeitintervall fallen (siehe Kapitel 4.3.2.2).

Nach der Bearbeitung eines jeden Nutzers wird die Liste mit allen berechneten Zeitintervallen und zugehörigen Termen in der Datenbank gespeichert. Das Vorgehen ist dabei wie folgt: Es wird über alle Zeitintervalle des Nutzers iteriert. Diese sind, entsprechend der Vergrößerungsstufe, dem Gewicht und der Start- und Endposition unterschiedlich. Wurde das Zeitintervall dem Benutzer in der Tabelle *UserMovement* noch nicht zugeteilt, so wird dieses neu angelegt. Es kann jedoch sein, dass das entsprechende Zeitintervall in der Tabelle

Connection bereits existiert. In diesem Fall wird lediglich das Gewicht durch Addition des neuen Gewichts des benutzerspezifischen Intervalls aktualisiert. Ansonsten muss das Intervall entsprechend dem Datenbankschema neu angelegt werden. Ein Intervall wird so bei häufigem Vorkommen bei der entsprechenden Vergrößerungsstufe und zugehöriger Start- und Endposition nicht doppelt oder mehrfach angelegt, sondern nur ein einziges Mal mit summierten Gewichten. Am Ende werden die Terme hinzugefügt. Hier ist das Vorgehen ähnlich wie bei den Zeitintervallen. Existiert noch keine Zuordnung von Zeitintervall zu Term, so wird in der Tabelle *TermMapping* diese Zuordnung erstellt. Enthalten ist auch die lokale Termfrequenz, multipliziert mit der Zeitintervall-Gewichtung. Falls die Verbindung schon existiert, wird lediglich die lokale Termfrequenz durch Addition des neuen Wertes mit dem existierenden aktualisiert. Ist anschließend der Term bereits in der Tabelle *Term* enthalten, so wird die globale Termfrequenz durch Addition aktualisiert. Ansonsten muss der Term neu angelegt werden.

Die Darstellung der Daten in der Datenbank ist recht effizient, jedoch zeigt sie auch Grenzen auf. Die Aggregation findet, wie bereits in Abschnitt 5.2.2.1 beschrieben, parallel statt. Dazu wird auch für jeden laufenden Thread eine eigene Verbindung zur Datenbank geöffnet. Durch Analyse der laufenden Aggregation kristallisieren sich zwei Teile heraus, die die meiste Rechenzeit beanspruchen. Einer davon ist die Abstandsberechnung des kürzesten Weges, da hier immer alle Voronoi-Städte betrachtet werden müssen und dies können je nach Vergrößerungsstufe Tausende sein. Dieser Teil ist jedoch nicht so rechenintensiv wie das Hinzufügen der Verbindungen zur Datenbank. Zwischen 95% und 97% der Rechenzeit wird dafür verwendet. Das Einfügen in die Datenbank ist dabei nicht das Problem, sondern das ständige Abfragen, ob schon etwas existiert. Als Beispiel sollen die Terme dienen. Angenommen, auf einer Reise werden im Durchschnitt davor fünf Tweets mit jeweils zehn unterschiedlichen Termen veröffentlicht und nach der Reise genauso viele. Eine Reise dauert im Durchschnitt zwei Stunden. Betrachtet man so eine Durchschnittsreise, dann fallen insgesamt 100 verschiedene Terme an. Zusätzlich entstehen vier Zeitintervalle á 30 Minuten. Auf jedes Intervall werden die Terme abgebildet. In der Tabelle *TermMapping* entstehen folglich für diese Verbindung 400 Einträge. Geht man davon aus, dass diese Reise auf allen Vergrößerungsstufen erhalten bleibt, so sind es schon 1600 Einträge. Betrachtet man mehrere und auch längere Reisen, so steigt dieser Wert gewaltig. Die Aggregation für den Datensatz einer beliebigen Woche ergibt etwa 90 Millionen Abbildungen von Zeitintervallen auf Terme bei den Trajektorien von ca. 150 000 Benutzern. Diese Dimensionen verlangsamen nicht nur die Aggregation, sondern auch das spätere Abfragen zur Visualisierung der Daten.

5.2.3 Visualisierung der Ergebnisse

In Abbildung 5.2 wurde bereits beschrieben, welche Komponenten in der Oberfläche enthalten sind. Interessant ist auch der Zusammenhang zwischen ihnen. Die Werkzeugleiste dient allgemeinen Optionen, welche direkt mit der Karte zu tun haben, wie beispielsweise das Aktivieren der Auswahlwerkzeuge usw. Die restlichen drei Komponenten hängen jedoch etwas enger zusammen. Bei der Filterverwaltung wirken sich die Änderungen direkt auf die Karte sowie die Detailansicht aus. Wird ein Filter entfernt, so werden die Daten, exklusive

dieses Filters, an die Karte und Detail-Ansicht propagiert. Beim Hinzufügen eines Filters verhält es sich etwas anders. In der Ansicht, in welcher der Filter gesetzt wurde, muss nichts mehr verändert werden. Dementsprechend wird die Änderung ausschließlich an die jeweils andere Ansicht propagiert. Die Karte und die Detailansicht kommunizieren fast einzig und allein über die Filter. Eine Ausnahme stellt die Anzeige der Tweets in der Detail-Ansicht dar, in welcher beim Anklicken eines Tweets dieser direkt auf der Karte angezeigt wird, ohne einen Filter zu setzen.

Im Folgenden wird näher auf die einzelnen Komponenten sowie deren Architektur eingegangen.

5.2.3.1 Weitergabe von Datenänderungen

In Kapitel 4.4.1 wurde bereits beschrieben, dass Datenänderungen über das Beobachter-Muster weitergegeben werden. In dieser Arbeit wird nicht der klassische Ansatz des Beobachter-Musters verwendet, in welchem sich Beobachter beim delegierten Subjekt registrieren müssen und über das Implementieren einer Schnittstelle Änderungen mitbekommen. Hier wird das als Grundidee aufgenommen und stattdessen ein erweiterter Ansatz über Listener gewählt [Ull07].

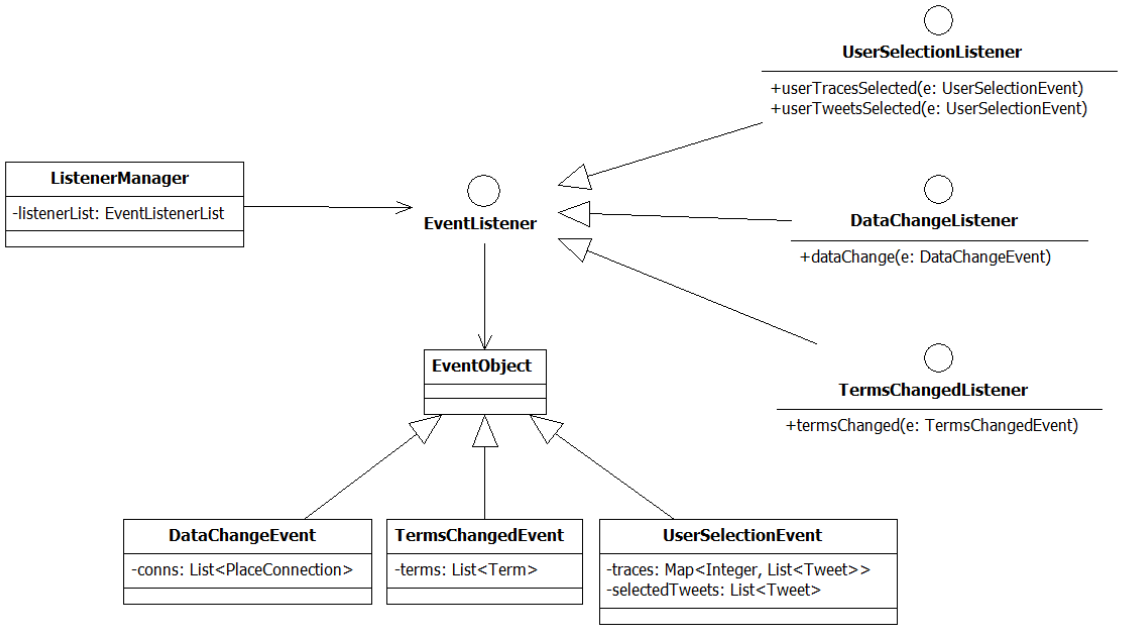

Abbildung 5.5: Umsetzung des Beobachter-Musters über Listener

Abbildung 5.5 zeigt den Aufbau des Beobachter-Musters über Listener. Die Beobachter implementieren die jeweilige Schnittstelle, die vom Java-eigenen **EventListener** abgeleitet

wird, und registrieren sich beim **ListenerManager**, damit sie entsprechend gehandhabt werden und Änderungen mitbekommen. Zusätzlich zu den Schnittstellen gibt es die Subjekte, die vom Java-eigenen **EventObject** abgeleitet werden. Wird ein Subjekt geändert, egal von wo aus, wird im **ListenerManager** dafür Sorge getragen, dass die Änderungen über die entsprechende Schnittstelle an alle registrierten Klassen propagiert werden. Es werden drei verschiedene Schnittstellen zur Implementierung angeboten. Jede dient einem anderen Zweck.

DataChangeListener Die visualisierten Verbindungen zwischen Städten werden entsprechend dem Gesamtzeitraum in Zeitintervalle aufgeteilt, wie in Kapitel 4.3.2 beschrieben wurde. Durch Filtern der Daten wird gegebenenfalls die Anzahl der abgebildeten Zeitintervalle reduziert. Je nachdem, ob nach Benutzern oder Zeit gefiltert wird oder die Daten beim Start des Programms direkt aus der Datenbank geladen werden, müssen die Daten weitergegeben werden. Dies geschieht über die Implementierung der Schnittstelle **DataChangeListener**.

UserSelectionListener Die Implementierung dieser Schnittstelle dient der Weitergabe von benutzerspezifischen Informationen. Wird beispielsweise ein Benutzerprofil ausgewählt, so muss an andere Programmteile, wie beispielsweise der Karte, mitgeteilt werden, welche Tweets geladen und welche Verbindungen oder nicht-aggregierte Trajektorien dargestellt werden sollen.

TermChangedListener Wird von allen Komponenten implementiert, die über Änderungen der Terme, welche beispielsweise in Form einer Schlagwortwolke dargestellt werden, informiert werden wollen.

Diese Form der Implementierung bringt viele Vorteile mit sich. Ein nennenswerter Vorteil ist die Möglichkeit der parallelen Propagierung der Datenänderungen. Dabei werden die Beobachter nicht sequenziell bearbeitet, sondern alle Beobachter können die Daten unmittelbar nach Änderung für ihre Zwecke verarbeiten. Dieser Ansatz ist nur beschränkt oder überhaupt nicht anwendbar, falls die Beobachter voneinander abhängig sind.

5.2.3.2 Filterverwaltung

Durch Anwendung der Filter können die visualisierten Daten effizient exploriert werden. Diese Filter können in beliebiger Reihenfolge gesetzt werden. In Kapitel 4.4.2 wurden bereits die verwendeten Filter und ihre Konzepte vorgestellt: Textfilter, Zeitfilter, Userfilter und Kartenfilter. Diese lassen sich in zwei Filtern zusammenfassen: den Zeitfilter und den Userfilter, wobei der Textfilter und die verschiedenen Kartenfilter unter den Userfilter fallen. Abbildung 5.6 zeigt das Klassendiagramm für die Filterverwaltung.

Es werden nicht nur die Ergebnisse der Filterung unmittelbar visualisiert, sondern auch die repräsentative Form des gesetzten Filters als Button. Wie in Abbildung 5.6 zu sehen ist, erbt jeder spezifizierte Filter (**TimeFilter**, **UserFilter** und **VoronoiFocusFilter**) von der abstrakten Klasse **Filter**. Sie gibt die abstrakte Methode **filter(...)** vor, die von allen abgeleiteten Klassen implementiert werden muss, stellt aber gleichzeitig auch einen Button dar.

Diese Filter werden von der Klasse **FilterManager** verwaltet. Das **FilterInterface** muss von Klassen implementiert werden, die den jeweiligen Filter setzen, also den Ausgangspunkt der Filterung.

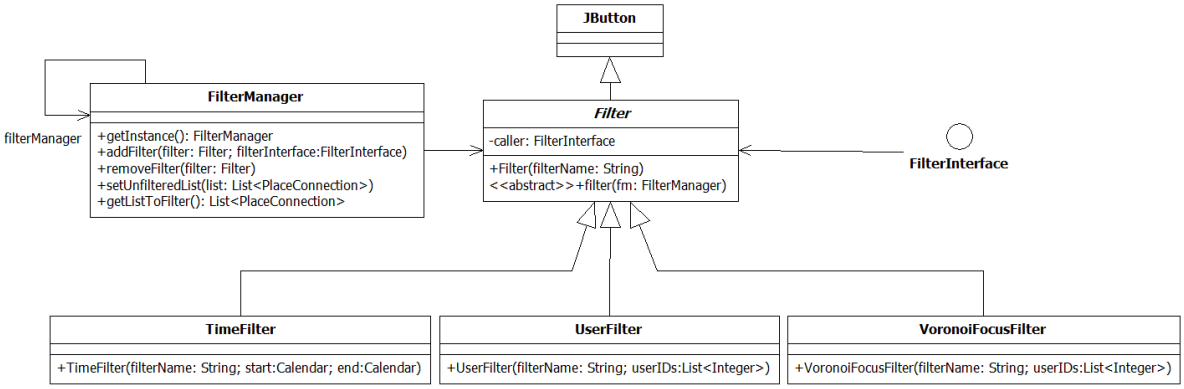

Abbildung 5.6: Klassendiagramm für die Umsetzung des Filterprinzips

In Kapitel 4.4.2 wurde bereits beschrieben, dass die Filter in der gesetzten Reihenfolge logisch miteinander Verknüpft werden. Dies wird ebenfalls in der Oberfläche gespiegelt. Die Filter werden, wie bereits erwähnt, durch einen Button mit entsprechender Beschriftung angezeigt. Durch das Klicken auf einen entsprechenden Button wird dieser Filter wiederum entfernt. Für die Umsetzung dieser Idee werden in der Klasse **FilterManager** drei verschiedene Listen gehalten. Anfangs wird eine ungefilterte Liste gesetzt, welche immer die initialen Daten enthält. Auf deren Basis wird der allererste Filter ausgeführt. Die zweite Liste wird durch die Filter durchgereicht und enthält die bereits gefilterten Daten. Diese Liste wird erstmals mit dem zuerst gesetzten Filter befüllt. Jeder weitere Filter filtert auf Basis dieser Liste. Die dritte Liste ist eine Liste für den Fall, dass ein Filter mehrmals in Folge gesetzt wird. Ein Beispiel dafür ist der Benutzerfilter. Durch eine Kartenselektion wird dieser Filter gesetzt und die Benutzeranzahl eingeschränkt. Innerhalb der gefilterten Daten kann nochmal ein Benutzerfilter angewendet werden. Anstatt einen komplett neuen Filter zu setzen, wird der zuletzt gesetzte Filter erweitert.

Falls ein Filter entfernt wird, so werden die restlichen Filter erneut in der vorher gesetzten Reihenfolge ausgeführt, damit die Daten, exklusive des entfernten Filters, gefiltert sind.

Ein kurzes Anwendungsbeispiel, welches die Umsetzung des dargestellten Klassendiagramms zeigt: Ein Analyst bekommt in der Oberfläche einen Datensatz über das Reiseverhalten in Europa visualisiert. Interessant sind für ihn lediglich die Daten in der siebten Kalenderwoche. Der Zeitfilter implementiert das **FilterInterface** und erstellt nach Setzen der passenden Zeit ein neues Objekt der Klasse **TimeFilter**. Dieses wird nun über den **FilterManager** entsprechend gesetzt, die Daten gefiltert und an alle restlichen Komponenten propagiert. Zusätzlich sind für den Analysten alle Reisen von Interesse, die in London starten oder enden. Dazu selektiert er über einen Kartenfilter, welcher auch das **FilterInterface**

implementiert, die gewünschte Region. Wie beim Zeitfilter wird ein neues Objekt erzeugt, gesetzt und die neuen Ergebnisse propagiert. Gefiltert werden die Daten, die beim Zeitfilter als Ergebnis verbreitet wurden. Entschließt sich der Analyst den Zeitfilter zu entfernen, dann drückt er auf den angezeigten Button. Da die Implementierung des Zeitfilters die Schnittstelle **FilterInterface** implementiert, wird die vorgegebene Methode **reset()** in der Anzeige des Zeitfilters aufgerufen, sodass diese zurückgesetzt werden kann. Nachdem die Anzeige zurückgesetzt wurde, werden die Daten neu, lediglich mit dem Kartenfilter, gefiltert und an die restlichen Komponenten propagiert. Somit ist die logische UND-Verknüpfung der Filter untereinander garantiert, sowohl beim Hinzufügen als auch beim Entfernen.

5.2.3.3 Werkzeugleiste

Die Werkzeugleise (Abbildung 5.7) bietet dem Analysten die Möglichkeit, verschiedene Optionen zur Interaktion und Filterung auszuwählen.

Abbildung 5.7: Anzeige der Werkzeugleiste. Aufteilung in sieben Bereiche, von links nach rechts.

Von links nach rechts ist diese Leiste in sieben Bereiche unterteilt. Der **erste Bereich** bietet die Option, die Oberfläche für die Aggregation zu öffnen.

Der **zweite** und **dritte Bereich** dienen ausschließlich der Karteninteraktion. Hier hat der Analyst die Möglichkeit auszuwählen, wie er mit der Karte interagieren will: die Karte verschieben, einzelne Trajektorien selektieren, Rechteckselektion ins Freie oder von Voronoi-Zellen und Vergrößerungsoptionen.

Der **vierte Bereich** ist zum Feststellen der angezeigten Daten. Es ist ein Semantischer Zoom realisiert. Beim Klicken dieses Buttons wird der Semantische Zoom aktiviert, d.h. auf verschiedenen Vergrößerungsstufen die Daten verfeinert oder vergröbert. Durch Lösen des Buttons bleiben die visualisierten Daten für jede beliebige Vergrößerungsstufe erhalten.

Zum **fünften Bereich** gehören drei verschiedene Anzeigeoptionen: das Ein- und Ausblenden der Voronoi-Verbindungen, der einzelnen Benutzertrajektorien und der Richtungen der Verbindungen.

Im **sechsten Bereich** der Werkzeugleiste bekommt der Analyst die Möglichkeit, Tweets animiert darstellen zu lassen. Durch den Button Play kann er starten und durch den Button Pause pausieren.

Der **siebte Bereich** dient der textuellen Filterung der Daten. Durch Eingabe mehrerer Stichwörter, getrennt durch Komma, werden alle Benutzer herausgefiltert, die diese Wörter nie in

ihren Tweets verwendet haben. Dies ist der Textfilter, der letztendlich in einen Benutzerfilter resultiert.

Listing 5.2 Textuelle Filterung der Daten mittels Datenbankabfrage

```
SELECT DISTINCT user_id
FROM tweet
WHERE to_tsvector('english', text) @@ to_tsquery('english', 'frankfurt & airport & strike');
```

Listing 5.2 zeigt die Datenbankabfrage für alle Benutzer, die im vorhandenen Datensatz über die Stichwörter „frankfurt“, „airport“ und „strike“ geschrieben haben. Da die meisten Tweets in Englisch verfasst werden, legen wir fest, dass lediglich im englischen Sprachraum nach Ähnlichkeiten gesucht wird und dass alle drei Wörter in einer Nachricht vorkommen müssen. Durch die Anwendung weiterer Indices (siehe Kapitel 5.1) lassen sich bei Bedarf auch andere Sprachen unterstützen. Als Ergebnis erhält man alle Benutzer, die solche Nachrichten jemals verfasst haben. Über einen Benutzerfilter werden die Ergebnisse propagiert.

5.2.3.4 Detail-Ansicht

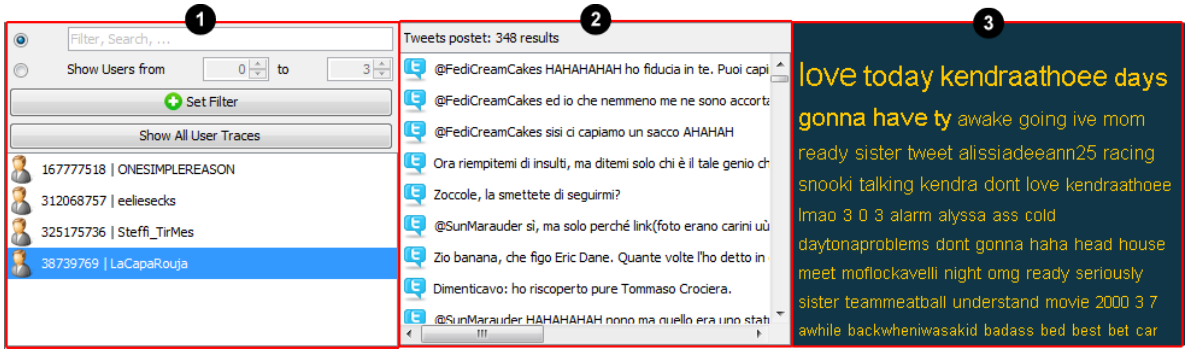

Abbildung 5.8: Aufteilung der Detail-Ansicht in drei Bereiche. Erstens: Darstellung aller visualisierten Benutzerprofile. Zweitens: Liste der Tweets aller ausgewählten Benutzerprofile. Drittens: Schlagwortwolke, generiert aus den ausgewählten Benutzerprofilen.

Die Detail-Ansicht besteht aus drei verschiedenen Hauptkomponenten, die zusammenhängend sind. Dazu gehört als Erstes die Ansicht mit allen auf der Karte visualisierten Benutzerprofilen. Es werden jeweils die ID und der Benutzername angezeigt. Zusätzlich enthält dieser Teil zwei verschiedene Benutzerfilter. Zum einen kann die Benutzerliste rein textuell und zum anderen über die Anzahl der Benutzerprofile gefiltert werden. Bei der textuellen Filterung wird nach IDs oder Benutzernamen gefiltert, welche die gesuchte Zeichenkette enthalten. Beim Filtern über die Anzahl der Benutzerprofile kann ausgewählt werden, bei welchem Benutzerprofil angefangen wird und bis zu welchem Benutzerprofil

in der Liste die Daten angezeigt werden. Dies ist ein legitimes Mittel, um die Größe des Datensatzes einzuschränken, ohne dabei Informationen zu verlieren, falls man Techniken wie das Paging anwendet.

Wählt man ein oder mehrere Benutzerprofile aus, werden die dahinter liegenden Daten nicht nur auf der Karte visualisiert. Zusätzlich werden in der Detail-Ansicht die Daten aktualisiert. Dazu gibt es eine Liste mit allen Tweets der selektierten Benutzerprofile, sortiert nach Zeit. Durch Anklicken der Tweets werden diese auf der Karte sowie in der Timeline-Anzeige dargestellt.

Zusätzlich zu der Tweet-Anzeige in einer Liste werden aus der Datenbank die in der Aggregationsphase bestimmten Terme, welche mittels Termfrequenz plus Zeitintervall-Gewichtung abgelegt wurden, geladen. Dargestellt werden diese Terme in einer nach Relevanz sortierten Schlagwortwolke. Wie im Grundlagenkapitel 2 bereits erläutert, wird für die Darstellung der Schlagwörter die Gewichtung auf eine Schriftgröße abgebildet. Dazu wird eine Mindestschriftgröße sowie eine maximale Schriftgröße bestimmt. Auf den Bereich dazwischen werden durch Normalisierung der Gewichte die entsprechenden Schriftgrößen ausgewählt. Die Terme sind absteigend nach Gewichtung sortiert, sodass der Term mit der höchsten Gewichtung die höchste Relevanz besitzt und somit auf die maximale Schriftgröße abgebildet wird. Die Terme beschreiben – wie bei der Aggregation (siehe Kapitel 5.2.2) festgelegt – lediglich die Tage der Ankunft oder der Abreise.

Listing 5.3 Datenbankabfrage zur Gewinnung der Terme für die Darstellung in einer Schlagwortwolke

```
SELECT DISTINCT (te.term), SUM(t.term_frequency) AS tfreq, MAX(te.global_term_frequency)
FROM usermovement u JOIN termmapping t ON u.conn_id = t.conn_id JOIN term te ON t.term_id =
    te.term_id
WHERE u.user_id = 38739769
GROUP BY te.term
ORDER BY tfreq DESC
```

Listing 5.3 zeigt die Datenbankabfrage, die die Terme samt Gewichtung aus der Datenbank lädt. Die Idee ist wie folgt: Nach dem Datenbankschema zur Aggregation, welches in Kapitel 4.3.4 erstellt wurde, werden den jeweiligen Zeitintervallen Terme zugeordnet. In dieser Ansicht geht es – entgegengesetzt der Kartendarstellung – nicht um die zeitliche Komponente, sodass die Terme entsprechend den zugehörigen Benutzerprofilen geladen werden können. In diesem Beispiel werden alle Terme für das Benutzerprofil mit der ID „38739769“ geladen. Hierbei wird ein Join über die Tabellen *usermovement*, *termmapping* und *term* ausgeführt. Diese Kombination resultiert aus der Datenbankstruktur, da über das Benutzerprofil die zugehörigen Zeitintervalle ausfindig gemacht werden können. Über diese Zeitintervalle können nachfolgend die Terme mit zugehöriger Gewichtung geladen werden. Da die Terme abhängig von der Benutzertrajektorie sind, spielt ausschließlich die lokale Termfrequenz eine Rolle. Die globale Termfrequenz spielt nur bei einem Gesamtüberblick eine Rolle. Da sich die Termfrequenzen von Intervall zu Intervall unterscheiden können, werden die einzelnen Frequenzen von gleichen Termen aufaddiert. Dies wird über die Gruppierung und die nachfolgende Summierung erreicht.

5.2.3.5 Karten-Ansicht

Die Karten-Ansicht bildet das Herzstück der Benutzeroberfläche. Als Basis dient eine Landkarte. Darüber können beliebig viele durchsichtige Überlagerungen gelegt werden, auf welchen verschiedene Komponenten abgebildet werden.

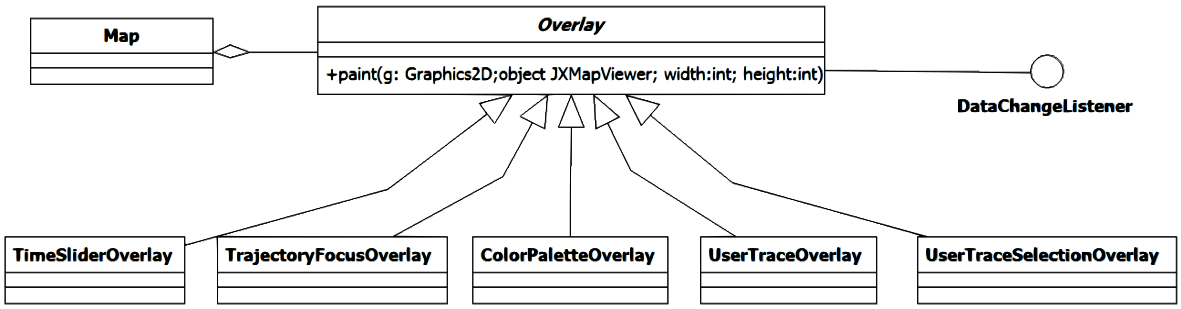

Abbildung 5.9: Klassendiagramm für die Realisierung mehrerer Überblendungen für die Karten-Ansicht

Abbildung 5.9 zeigt das Klassendiagramm für die Realisierung der Karten-Ansicht. Es gibt die Hauptkarte **Map**, welche von mehrere Lagen überblendet wird. Diese werden durch die Klasse **Overlay** spezifiziert. Durch Ableiten dieser Klasse und Überschreiben der spezifizierten Methode **paint(...)** ist es möglich, beliebige Überlagerungen zu erstellen.

Einbindung der Karte via Openstreet Maps

Die unterste Ebene der Karten-Ansicht besteht aus einer Weltkarte. Es gibt mehrere Möglichkeiten, um Karten in die Benutzeroberfläche einzubinden. Im Rahmen dieser Arbeit wird das Framework SwingX [jav] verwendet. Die beschriebene Technik mit mehreren Überlagerungen ist ohne Probleme anwendbar. Das Framework bringt zusätzliche Möglichkeiten zur Erweiterung mit.

In dieser Diplomarbeit wird für die Umsetzung die Landkarte via SwingX von Openstreet Maps über einen Webservice geladen. Dies hat den offensichtlichen Nachteil, dass eine Internetverbindung zur Laufzeit des Programms bestehen muss. Dies sollte im Zeitalter des Internets jedoch keinerlei Schwierigkeiten darstellen.

Semantischer Zoom

Der Semantische Zoom steuert den dargestellten Informationsgehalt anhand der Vergrößerungsstufen der Landkarte. Durch Nutzen der angelegten Datenstruktur, die in Kapitel 4.3

entwickelt wurde, werden für bestimmte Vergrößerungsstufen verschiedene Daten vorgehalten oder verfeinert. Es gibt insgesamt fünfzehn Stufen, von welchen für den Semantischen Zoom folgende von Bedeutung sind: fünfzehn, elf, acht und fünf. Bei diesen Stufen wird der dargestellte Datensatz gewechselt. Beim Vergrößern der Karte kommen mehr Informationen hinzu und beim Verkleinern nimmt der Informationsgehalt folglich ab.

Listing 5.4 Datenbankabfrage für die Bereitstellung aller notwendigen Daten für eine spezifizierte Vergrößerungsstufe

```
SELECT c.conn_id, c.time_start, c.time_end, c.start_voronoi_id, c.end_voronoi_id, c.weight,
    c.zoom_stage, u.user_id, u.user_location, u.user_mention, user_screenname
FROM connection c, the_user u, usermovement um
WHERE c.zoom_stage = 15
  AND c.conn_id = um.conn_id
  AND um.user_id = u.user_id
ORDER BY c.start_voronoi_id, c.end_voronoi_id, c.time_start
```

Listing 5.4 zeigt die verwendete Datenbankabfrage für die Daten, die bei einer gegebenen Vergrößerungsstufe dargestellt werden. In diesem Beispiel werden alle Daten für die Stufe fünfzehn geladen. Für die Abfrage werden drei verschiedene Tabellen benötigt, die mit einem Join miteinander verbunden werden: *Connection*, *User* und *UserMovement*. Dieses Vorgehen resultiert aus der gewünschten Repräsentation der Daten. Für die Darstellung in der Karten-Ansicht werden die Gesamtzeiträume der Verbindungen zwischen Städten als verkettete Zeitintervalle mit allen notwendigen Zusatzinformationen vorgehalten. Für ein einzelnes Zeitintervall werden die ID, die Startzeit, die Endzeit, das Gewicht sowie alle beteiligten Benutzer benötigt. Für die Darstellung der Verbindung braucht man die Startposition, die Endposition sowie das Gesamtgewicht der Verbindung.

Ein mögliches Vorgehen zur Bereitstellung der Daten für die Visualisierung wäre, alle Daten aus der Tabelle *Connection* zu laden und mit der gewonnenen ID alle Benutzerprofile über die Tabelle *UserMovement* zu laden. Ein effizienterer Ansatz als der, der hier zum Einsatz kommt, ist all diese Tabellen zu vereinigen. Darüber hinaus können die Daten über die Sortierung sequentiell verarbeitet werden, ohne unnötige Abfragen in Listen oder Tabellen, die zusätzliche Zeit und Rechenleistung kosten. Die Sortierung der Daten erfolgt zunächst über den geographischen Startpunkt, innerhalb dieser Sortierung nach dem geographischen Endpunkt und dann erst nach dem Zeitintervall. Die Ergebnisliste kann sequentiell abgearbeitet werden, wie im Algorithmus 5.2 beschrieben wird.

Der Algorithmus iteriert über alle Ergebniszeilen der Datenbankabfrage. Sind die geographischen Start- und Endpunkte identisch mit den Punkten der vorherigen Ergebniszeile, so wird immer noch dieselbe Verbindung zwischen den Städten bearbeitet. In diesem Fall muss die ID des Zeitintervalls überprüft werden. Falls diese sich nicht verändert hat, wird darüber hinaus auch noch dasselbe Zeitintervall bearbeitet. Dann wird nur der Benutzer des Zeitintervalls hinzugefügt; ansonsten muss ein neues Zeitintervall mit allen verfügbaren Informationen angelegt und der aktuellen Verbindung zwischen den Städten angehängt werden. Falls eine neue Verbindung erstellt werden soll, wird dies getan und direkt der Ergebnisliste hinzugefügt. Wird zur Laufzeit des Programms ein Vergrößerungsschwellwert

Algorithmus 5.2 Algorithmus zur Bereitstellung der Daten nach Laden aus der Datenbank

```
for all e ∈ rows do
    startCity_i ← e.startCity
    endCity_i ← e.endCity
    connectionID_i ← e.connectionID
    if startCity_i == startCity_{i−1}  &&  endCity_i == endCity_{i−1} then
        Gleiche Verbindung
        if connectionID_i == connectionID_{i−1} then
            Gleiches Zeitintervall.
            Füge lediglich den Benutzer zum aktuellen Zeitintervall hinzu
        else
            Neues Zeitintervall
            Erstelle neues Zeitintervall und füge es zur aktuellen Verbindung hinzu
        end if
    else
        Neue Verbindung
        Erstelle neue Verbindung und füge sie zur Ergebnisliste hinzu
    end if
    startCity_{i−1} ← startCity_i
    endCity_{i−1} ← endCity_i
    connectionID_{i−1} ← connectionID_i
end for
```

erreicht und wurden für diese Stufe die Daten noch nicht aus der Datenbank geladen, wird dieser Algorithmus durchgeführt. Anschließend werden die Daten im Speicher gehalten.

Verbindungen zwischen Städten

Nachdem die Verbindungen aus der Datenbank geladen wurden, können sie im nächsten Schritt visualisiert werden. Für die Visualisierung dieser Verbindungen gibt es eine extra Überblendung (**UserTraceOverlay**). Eine Verbindung besitzt einen Richtungspfeil, eine Gewichtung, die auf einen bestimmten Farbwert abgebildet wird sowie einen geographischen Start- und Endpunkt. Im weiteren Verlauf dieses Kapitels werden diese Eigenschaften noch beschrieben, an dieser Stelle ist lediglich der Prozess der Visualisierung von Bedeutung.

Zunächst müssen die bekannten Geokoordinaten von Start- und Endpunkt bei jedem Neuzeichnen der Oberfläche in Pixel umgerechnet werden. Das verwendete Framework bietet die notwendigen Funktionen dazu [jav]. Die Neuberechnung der exakten Koordinaten bei jedem Neuzeichnen resultiert aus der Verschiebung des Nullpunkts auf der Karte. Bei einer Vergrößerung, Verkleinerung oder Verschiebung der Karte wird automatisch der Nullpunkt in der linken, oberen Ecke auf andere Geokoordinaten abgebildet. Damit beispielsweise visualisierte Städte an der korrekten Position bleiben, muss aus ihren exakten Geokoordinaten die entsprechende Position in Pixeln bei jeder Neuzeichnung berechnet werden.

Eine Verbindung zwischen Städten wird unter Angabe der Richtung durch eine Linie repräsentiert. Die Richtung wird durch einen Pfeil signalisiert, der inmitten der Linie positioniert ist. Zusätzlich wird die Anzahl der Zeitintervalle, die zur Verbindung gehören, auf einen Farbwert abgebildet, der angibt, wie relevant die Verbindung nach der Anwendung von Filtern ist.

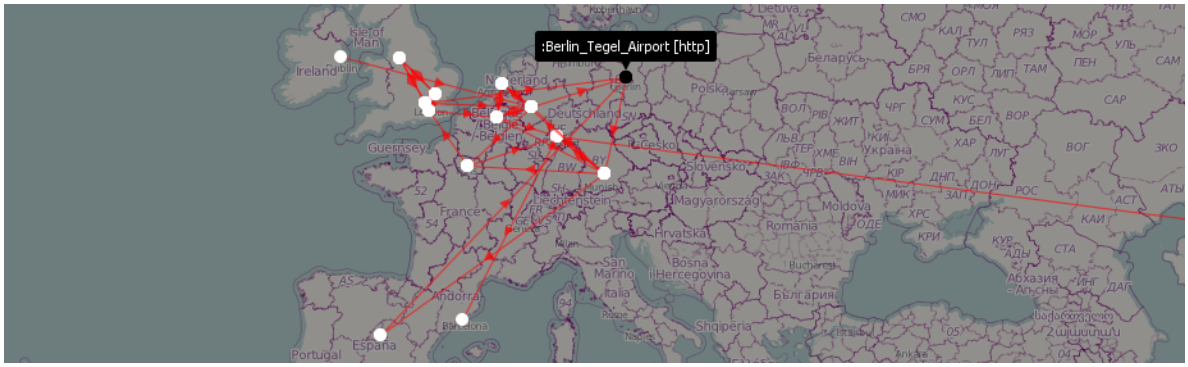

Abbildung 5.10: Verbindungsvisualisierung zwischen Städten

Abbildung 5.10 zeigt die Visualisierung von Verbindungen zwischen Städten. Die aus den Voronoi-Zellen resultierenden Städte werden durch weiße Punkte dargestellt. Beim Berühren einer Stadt mit der Maus wird der Name angezeigt. Bei der kleinsten Vergrößerungsstufe handelt es sich um Drehkreuze anstatt Städte, wie in der Abbildung zu sehen ist. Des Weiteren können durch Selektion einer Verbindung alle ihr zugeordneten Benutzerprofile mit ihren individuellen, unaggregierten Trajektorien, zusätzlich visualisiert werden. Die Erkennung, ob die Maus die entsprechende Stadt berührt, wird über die Koordinaten der Mausspitze getätigt. Es wird überprüft ob sich diese Koordinaten in dem Kreis, der die Stadt repräsentiert, befinden.

Für die effiziente Darstellung der Städteverbindungen wird ein Quadtree eingesetzt (siehe Kapitel 2.3.5). Verwendet wird diese Technik der räumlichen Abbildung auf einen Baum für Vergrößerungen und Verkleinerungen der angezeigten Karte. Ein Beispiel: Der Analyst vergrößert den europäischen Bereich. In diesem Fall interessiert eine Verbindung zwischen Villarrica und Buenos Aires nicht. Es ist demnach unnötig die Verbindung zu rendern. Für die Darstellung der Verbindungen wird immer der Sichtbereich der Karte betrachtet. Dieses Rechteck wird von den Pixelkoordinaten in die Geokoordinaten umgerechnet, mit dessen Hilfe die entsprechenden Städte aus dem Quadtree geholt werden. Dabei geht es nicht direkt um die Verbindung, sondern vielmehr um die Städte. Sind Start- und Endstadt einer Verbindung außerhalb des Sichtbereichs, so werden weder die Städte noch die Verbindungslinie gezeichnet. Dies führt bei extrem großen Datenmengen zu einer flüssigeren Darstellung während der Interaktion mit der Karte. Bei der Suche nach enthaltenen Städten im Quadtree werden die Eckpunkte des Rechtecks rekursiv in den Baum eingesunken. Als Ergebnis erhält man alle sichtbaren Städte.

Farbabbildung

Wie soeben beschrieben, wird eine Verbindung zwischen zwei Städten in Abhängigkeit der Anzahl und Gewichtung der dargestellten Zeitintervalle auf eine Farbe abgebildet. Die Farbpalette ist ein Gradient von Schwarz über Gelb zu Rot. Die Abbildung auf die Farbe erfolgt über normalisierte Daten auf eine Lookup-Tabelle.

Das Prefuse Toolkit[3], welches verschiedene Komponenten zur Informationsvisualisierung bereitstellt, bietet unter anderem auch fertige Farbpaletten und Möglichkeiten zur Generierung von Farbpaletten an. Der Gradient von Schwarz über Gelb zu Rot lässt sich in zwei Teilschritten erstellen, wobei das Farbintervall beliebig groß sein kann. Bei einem zu großen Farbintervall kommen gegebenenfalls die Unterschiede der Daten nicht zum Vorschein. Die Größe des Intervalls sollte immer abhängig von der Größe des Datensatzes gemacht werden. Ist das Intervall zu klein, führen geringfügige Unterschiede oder Einschränkungen in den Daten zu markanten Farbunterschieden. Für dieses Beispiel soll das Farbintervall vierzig Einträge besitzen. Mit Hilfe von Prefuse wird ein Array erstellt, das zwanzig Farbwerte des Gradienten von Schwarz zu Gelb enthält und ein weiteres mit zwanzig Werten des Gradienten von Gelb zu Rot. Diese generierten Farbwerte werden dann gemeinsam in einem Array gespeichert. Die Opazität wird hierbei nicht beachtet.

Die ausgewählte Farbe hängt von der Gewichtung der Zeitintervalle ab. Jede Verbindung zwischen zwei Städten besitzt ein Gesamtgewicht, welches sich aus der Summe aller Gewichte der Zeitintervalle auf der Verbindung zusammensetzt. Werden die Daten beispielsweise nach Zeit gefiltert, dann werden bestimmte Zeitintervalle und damit auch ihre Gewichte verworfen. Die Summe der Gewichte der übriggebliebenen Zeitintervalle wird summiert und in das Verhältnis zum bekannten Gesamtgewicht gesetzt. Ist die Summe der Gewichte gleich dem Gesamtgewicht, so werden die Informationen der Verbindung vollständig repräsentiert. Es ergibt sich ein Wert zwischen null und eins, der mit der Länge des Farbarrays multipliziert wird. Als Ergebnis bekommt man die zum Gewicht zugehörige Farbe. Folgende Gleichung drückt das soeben Beschriebene mathematisch aus:

$$i = \frac{w_{selected}}{w_{full}} \cdot palette.size() \tag{5.1}$$

$$color = palette\,[i] \tag{5.2}$$

$w_{selected}$ gibt das summierte Gewicht der übriggebliebenen Intervalle an. w_{full} gibt das volle ungefilterte Gewicht der Verbindung an. Da das Farbarray bei Index null startet, kann der Quotient der Gewichte mit der Größe des Farbarrays multipliziert werden. Als Ergebnis erhält man den Index des Farbarrays, auf welchem der entsprechende Farbwert abgebildet ist.

[3]Prefuse: http://prefuse.org/

Textuelle Darstellung

Eine weitere Selektionsmöglichkeit von Verbindungen ist das Berühren der Verbindung mit der Maus. Im Vergleich zum Berühren der Städtekreise muss hier ein größerer Bereich betrachtet werden, da dem Analysten nicht die Aufgabe gestellt werden soll, die Koordinaten der Mausspitze exakt auf der 1px dicken Verbindung zu positionieren. Aus diesem Grund wird um die Mausspitze herum ein zehn Pixel großer Rahmen gesetzt. Besitzt eine Verbindung einen Eintritts- sowie einen Austrittspunkt bei dem Rahmen, so wird diese Verbindung als berührt gekennzeichnet. Wird eine Verbindung berührt, werden in Form einer linear abgebildeten Schlagwortwolke entlang der Verbindung die textuellen Informationen der Tweets von Start- und Endstadt dargestellt. Abbildung 5.11 zeigt die Darstellung in der Benutzeroberfläche bei Berührung der Verbindung mit der Maus.

Abbildung 5.11: Darstellung der Schlagwortwolke in linearer Form zwischen zwei Städten

Inmitten der Verbindung wird der Term mit der höchsten Relevanz positioniert, danach abwechselnd links und rechts davon die restlichen Terme in absteigender Relevanz. Die Schriftgröße ist abhängig von dem Gewicht und der vorher bestimmten minimalen und maximalen Schriftgröße. Die Berechnung erfolgt wie im Grundlagenkapitel 2.4 zur Darstellung relevanter Textinformationen.

Es kann jedoch vorkommen, dass eine Verbindung in beide Richtungen vorhanden ist. In diesem Fall wird die Verbindung in zwei Teile unterteilt, um die aggregierte Textinformation für beide Richtungen darzustellen. Der Text wird gleichzeitig über und unter der Verbindungslinie angezeigt; über der Verbindungslinie für die Richtung von links nach rechts und unter der Verbindungslinie für die Richtung von rechts nach links. Dies hat den Vorteil, dass bei der Berührung der Maus mit der Verbindung die Informationen auf eine übersichtliche Art und Weise für beide Richtungen angezeigt werden.

Es gibt auch andere Möglichkeiten, wie beispielsweise den Text nur für eine Richtung anzuzeigen, indem eine Verbindung in einen linken und rechten Teil unterteilt wird und je nachdem, welcher Bereich mit der Maus berührt wird, entweder die textuelle Information für die Richtung von links nach rechts oder von rechts nach links angezeigt wird. Bei diesem Ansatz ist es beispielsweise jedoch nicht möglich, für eine beliebige Stadt die Information aller eingehenden und ausgehenden Kanten zur gleichen Zeit anzuzeigen.

Auswahlwerkzeuge

In Abschnitt 5.2.3.3 wurde bereits die in die Benutzeroberfläche integrierte Werkzeugleiste vorgestellt. In diesem Abschnitt liegt der Fokus auf der Anwendung der Auswahlwerkzeuge der Werkzeugleiste in der Karten-Ansicht. Dazu gehören die Einzelselektion sowie die Rechteckselektion.

Auf die Einzelselektion wurde an verschiedenen Punkten bereits eingegangen. So wird lediglich durch Berühren der Städte der entsprechende Name angezeigt, bei Verbindungen die textuellen Informationen. Wird eine Verbindung darüber hinaus angeklickt, so werden alle mit ihr verbundenen Benutzerprofile mit den aggregierten textuellen Informationen über die komplette Reise hinweg dargestellt, anstatt lediglich die Stadt-zu-Stadt-Verbindung. Außerdem werden alle getätigten Tweets in zeitlich korrekter Reihenfolge in der Timeline-Anzeige dargestellt.

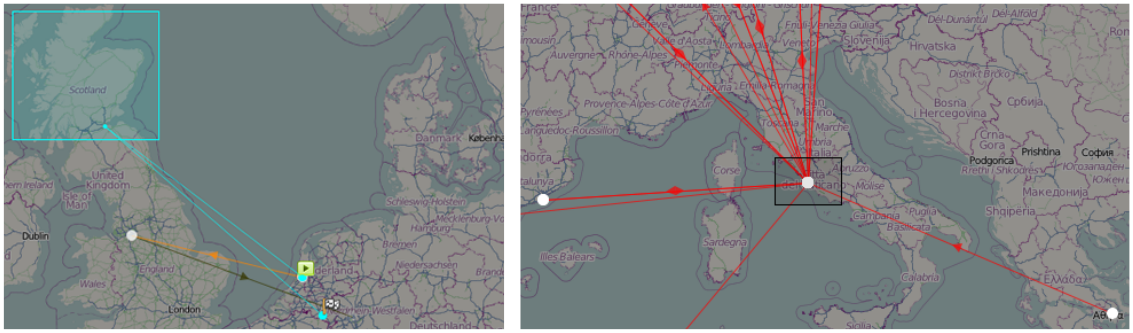

Abbildung 5.12: Unterschiedliche Möglichkeiten der Selektion über Auswahlrahmen. Links: Auswahl eines Bereichs, durch welchen keinerlei aggregierte Verbindungen laufen. Es wird in diesem Fall nach aggregierten Verbindungen gefiltert, deren zugeordnete Tweets im Auswahlrahmen enthalten sind. Rechts: Auswahl aggregierter Verbindungen

Abbildung 5.12 zeigt die zwei unterschiedlichen Wege der Selektion durch Auswahlrahmen. Das linke Bild beschreibt die erste Möglichkeit: Dieser Auswahlrahmen dient der Auswahl der individuellen Benutzertrajektorien, mit Start und Ziel gekennzeichnet. Zusätzlich werden aggregierte, zugehörige Städteverbindungen angezeigt. Es ist möglich, den Auswahlrahmen entweder über aggregierte Städte zu ziehen oder ihn ins Freie zu legen. Bei Ersterem werden alle individuellen Benutzertrajektorien angezeigt, die mit dieser aggregierten Stadt verbunden sind. Bei Letzterem werden alle individuellen Trajektorien angezeigt, die sich in diesem geographischen Bereich befinden. Der geographische Bereich wird durch die linke obere und rechte untere Ecke des Auswahlrahmens festgelegt. Aus der Datenbank werden alle Benutzer geladen, die jemals einen Tweet innerhalb dieser Geopositionen veröffentlicht haben. Über die jeweiligen Benutzerprofile kann der entsprechende Auswahlfilter auf der

Karte gesetzt werden, indem alle anderen dargestellten Benutzerprofile ausgeschlossen werden.

Listing 5.5 Datenbankabfrage für die Bereitstellung aller notwendigen Daten für eine spezifizierte Vergrößerungsstufe

```
SELECT user_id
FROM tweet
WHERE (lat > 56.08429756206141
  AND lng < -2.28515625
  AND lat < 58.83649009392136
  AND lng > -7.5146484375)
```

Listing 5.5 zeigt die Datenbankabfrage für den Auswahlrahmen über Schottland, wie es in der Abbildung dargestellt ist.

Das rechte Bild in Abbildung 5.12 zeigt die Selektion über einen Fokus der aggregierten Voronoi-Städte. Bei vielen visualisierten Daten fehlt oft der Überblick, sodass durch Verdeckung von Verbindungen für den Analysten nicht immer offensichtlich ist, zu welcher Stadt welche Verbindung gehört. Aus diesem Grund gibt es das Fokus-Auswahlwerkzeug. Damit ist es möglich, durch Auswahl bestimmter Städte und Verbindungen alle anderen auszublenden.

Bei beiden Selektionsarten gibt es die Möglichkeit, mehrere Auswahlrahmen zu erstellen. In diesem Fall werden die Ergebnisse miteinander vereinigt.

Zeitfilter

Der Zeitfilter (Abbildung 5.13) gehört streng genommen zu den Filtern, wird jedoch an dieser Stelle beschrieben, da er eine Komponente in der Karten-Ansicht darstellt. Mit seiner Hilfe können die Verbindungen zeitlich eingeschränkt werden. Aufgeteilt ist der Zeitfilter in Zeitintervalle, welche dieselbe Länge wie ein Verbindungszeitintervall besitzen. Somit können die Verbindungen direkt nach Zeitintervallen gefiltert werden. Jedes Zeitintervall beinhaltet in Form eines Histogramms die Anzahl an aktiven Verbindungen für diese Zeitspanne.

Abbildung 5.13: Zeitfilter

Falls der Zeitfilter eine sehr große Zeitspanne darstellt, ist die Auflösung der einzelnen Zeitintervalle sehr gering. Aus diesem Grund gibt es die Möglichkeit der Vergrößerung und

Verkleinerung. Bei der Vergrößerung wählt der Analyst einen Zeitbereich aus und klickt auf die Vergrößerungslupe, die unmittelbar darunter positioniert ist. Der ausgewählte Bereich wird nun über die volle Breite dargestellt und alle restlichen, nicht mit eingeschlossenen Zeitintervalle werden ausgeblendet. Für den Analysten ist so ersichtlicher, welche Zeitintervalle von ihm selektiert wurden. Dies wird zusätzlich durch die Zeitanzeige darunter unterstützt. Um zu der ursprünglichen Ansicht zurückzukehren, genügt ein Klick auf den Verkleinerungsbutton.

Durch Setzen der linken oder rechten Zeitgrenze wird unmittelbar danach der Filter gesetzt und die visualisierten Daten auf diese Zeit eingeschränkt. Der Zeitfilter wird als Überblendung der Karte angezeigt, um keinen Platz zu verschwenden. Das Herzstück der Benutzeroberfläche ist die Karte; diese soll auch den größten Teil des verfügbaren Platzes einnehmen.

Timeline-Anzeige von Trajektorien

Eine weitere ausblendbare Ebene der Karte ist die Timeline-Anzeige der individuellen Trajektorien. Diese wird aktiv durch die Selektion von Benutzerprofilen; die Menge spielt dabei keine Rolle. Die Selektion findet entweder über die direkte Benutzerauswahl in der dargestellten Liste oder über die Auswahl von Verbindungen auf der Karte statt.

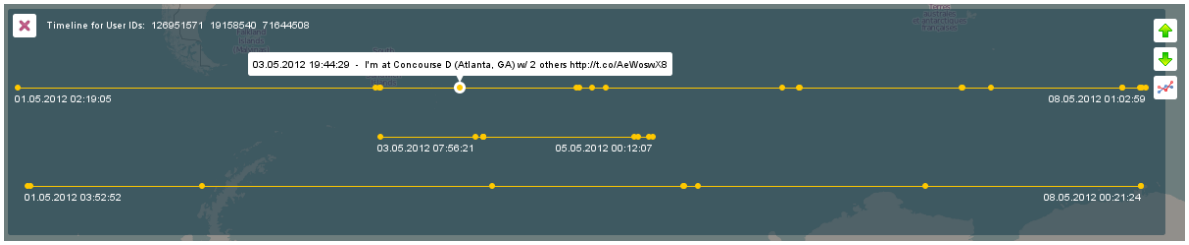

Abbildung 5.14: Timeline-Anzeige

Abbildung 5.14 zeigt einen Screenshot dieser Anzeige. Für jedes selektierte Benutzerprofil wird ein separater Zeitstrahl angezeigt. Dazu werden das kleinste und das größte Datum ausfindig gemacht, welche Start- und Endposition festlegen. Alle anderen Daten werden relativ dazu positioniert. Die Berechnung der relativen Position für Tweetposition t_i in Minuten verläuft wie folgt:

$$\Delta dur = t_n - t_0 \tag{5.3}$$

$$p_x = \frac{t_i}{\Delta dur} \tag{5.4}$$

Dabei bezeichnet t_n das letzte und t_0 das erste bekannte Datum der Benutzertrajektorie. Δdur ist somit die Zeitspanne zwischen größtem und kleinstem Datum in Minuten. Das Ergebnis p_x ist ein normalisierter Wert zwischen null und eins. Die Multiplikation des Abstands in Pixeln zwischen kleinstem und größtem Datum mit p_x ergibt die Position, an welcher der einzufügende Tweet gezeichnet wird.

Zusätzlich wird bei der Berührung von der Mausspitze mit einem Kreis, der einen spezifischen Tweet repräsentiert, der Tweet unmittelbar darüber mit genauer Zeitangabe angezeigt. Falls Benutzerprofile ausgewählt wurden und der Analyst in der Werkzeugleiste eine Animation startet, werden die Tweets nicht nur auf der Karte visualisiert, sondern auch in der Timeline-Anzeige in zeitlich korrekter Reihenfolge animiert, sodass immer der entsprechende Kreis, der den jeweiligen Tweet repräsentiert, hervorgehoben und der Textinhalt dargestellt wird.

Die Anzeige ist auf die gleichzeitige Visualisierung von drei Benutzertrajektorien beschränkt. Es ist aber möglich, durch Verwendung der zwei Pfeile am rechten Rand durch die Visualisierungen zu scrollen. Des Weiteren wird dem Analysten die Möglichkeit geboten, falls es beispielsweise um die Untersuchung von Ereignissen geht, alle ausgewählten Benutzertrajektorien auf einer Zeitachse mit unterschiedlichen Farbkodierungen anzuzeigen. Es lassen sich mit dieser Wahl sehr schnell Brennpunkte erkennen, da an bestimmten Punkten Benutzer gehäuft Tweets veröffentlichen. Diese Ansicht lässt sich mit dem Button am rechten Rand unter den Pfeilen aktivieren. Die Auswahl der Farbkodierung erfolgt sequentiell über den HSV-Farbraum. Dieser Farbraum wird über einen Kegel dargestellt und durch Angabe des Winkels, der Sättigung und der Helligkeit werden die Farben spezifiziert. Für diese Aufgabenstellung werden die höchste Helligkeit sowie die höchste Sättigung gewählt, sodass man sich auf dem HSV-Farbkreis bewegt. Nun wird über den Winkel in 60-Grad-Schritten iteriert, da immer an diesen Stellen der Farbwert wechselt. Wenn 360 Grad erreicht werden, fangen die Farben von vorne wieder an, also bei null Grad.

Am oberen Rand der Anzeige werden die IDs der Benutzerprofile angezeigt, deren Tweets visualisiert werden. Bei der Anzeige auf einer einzigen Zeitachse werden die IDs ebenfalls in den entsprechenden Farben dargestellt.

6 Fallstudie und Auswertung

Durch die Fallstudien wird verifiziert, inwiefern die entworfenen Konzepte eingesetzt werden und gegebenenfalls zum Erfolg führen können. Des Weiteren spiegeln die verwendeten Daten in den nachfolgenden Analysen nur einen kleinen Ausschnitt der Gesamtsituation. Lediglich ein Teil der Menschen verwendet soziale Medien und von denjenigen, die es tun, werden hier nicht alle Daten verwendet. Dies liegt an dem begrenzten Zugriff auf Twitterdaten, wie bereits in Kapitel 2.2.1.2 erläutert wurde. Die erzielten Ergebnisse stehen somit immer repräsentativ für den verwendeten Datensatz. Rückschlüsse auf den wirklichen Verlauf des Geschehens sind mit Vorsicht zu treffen.

In diesem Kapitel werden zwei Fallstudien durchgeführt: Die erste Fallstudie (Abschnitt 6.1 und 6.2) widmet sich der Exploration von Ereignissen; dazu gehören die re:publica in Berlin und die Comic-Con in San Diego. Mit der Analyse wird versucht, mit Hilfe der implementierten Visualisierungs- und Interaktionskonzepte nähere Informationen zu den Ereignissen und deren Besuchern herauszufinden. Die zweite Fallstudie (Abschnitt 6.3) führt einen Vergleich des Reiseverhaltens anhand von Verbindungsdaten durch.

6.1 Ereignis: re:publica 2012

Die re:publica (gesprochen: republica) ist eine Konferenz rund um das Thema Web 2.0 und die damit verbundenen Webblogs, sozialen Medien uvm. Sie findet jährlich in STATION-Berlin[1], in Kreuzberg, statt [rep]. STATION-Berlin ist ein ehemaliger Postbahnhof, welcher aktuell für verschiedene Veranstaltungen, wie etwa die re:publica, genutzt wird. Auf dieser Konferenz finden Diskussionen, Vorträge und Workshops statt. Für diese Fallstudie werden ausschließlich Twitterdaten verwendet und da diese Veranstaltung vor allem an Web 2.0-Begeisterte gerichtet ist, besteht die Möglichkeit, dass besonders viele Twitternutzer den sozialen Dienst während dieser Zeit auch genutzt haben. Im weiteren Verlauf des Kapitels wird die Konferenz anhand der aufgezeichneten Twitterdaten etwas genauer untersucht. Hierzu wird zunächst in der Übersicht gezeigt, welche allgemeinen Informationen über die Anreisenden herausgefunden werden können. Nachfolgend wird die Analyse verfeinert, sodass die Reise eines beliebigen Besuchers der re:publica untersucht wird. Vorab sei zu sagen, dass die Zahlen, die in den folgenden Unterkapiteln erwähnt werden, lediglich repräsentativ für die vorhandenen Daten stehen. Dies liegt vor allem daran, dass die meisten Menschen kein Twitter nutzen und falls doch, nicht unbedingt exzessiv.

[1] STATION-Berlin: http://www.station-berlin.de/

Da ein weltweiter Datensatz, der über längere Zeit verläuft, ein zu großes Volumen besitzt und es somit schwer wird diesen effizient zu visualisieren, werden die Daten im Vorfeld eingeschränkt. Die Konferenz fand in der Zeit vom 2.5.2012 bis 4.5.2012 statt, weshalb für die Visualisierung ein Datensatz ausgewählt wird, der den Zeitraum 1.5.2012 bis 8.5.2012 abdeckt. Zudem werden die Daten im Voraus bereits so aggregiert, dass nur Reisen enthalten sind, die mindestens einen Stopp in Deutschland enthalten.

6.1.1 Analyse der Konferenz

In der Visualisierung werden 4000 Benutzertrajektorien in der Übersichtskarte angezeigt, die in der Zeit vom 1.5.2012 bis 8.5.2012 mindestens einen Stopp in Deutschland hatten. Zwei Fakten sind zu Beginn der Analyse bekannt: Die Konferenz fand in Berlin-Kreuzberg statt und heißt re:publica. Durch Anwendung der Rechteckselektion wird Berlin markiert. Von den ursprünglich 4000 Benutzertrajektorien bleiben ca. 1100 übrig, die durch Berlin reisen. Durch die gewählte Datenbankstruktur sind 1100 Trajektorien zu viele, um aus diesen effizient eine Schlagwortwolke zu generieren, welche als Übersicht dient. Dies hängt mit dem in Kapitel 5.2.2.3 erläuterten Problem zusammen, dass die Terme auf einzelne Zeitintervalle abgebildet werden und bei entsprechend vielen Anfragen die Datenbanksuche verzögern. Bei 1100 Trajektorien müssten die Terme aller zugehörigen Zeitintervalle abgefragt werden. Aus diesem Grund bietet die Implementierung die Möglichkeit Teilmengen auszuwählen, d.h. man wählt z. B. nur zehn Trajektorien aus, wertet die Daten aus und geht dann zu den nächsten Trajektorien über. Abbildung 6.1 zeigt diesen ersten Schritt.

Abbildung 6.1: Darstellung der Rechteckselektion von Berlin und die damit generierte Schlagwortwolke für zehn individuelle Benutzertrajektorien

Schon bei den ersten Teilmengen erkennt man deutlich einen Trend: Die Terme „rp12“, „berlin“ und „republica“ besitzen die höchste Frequenz. Für die weitere Analyse werden alle gesetzten Filter zurückgesetzt und über die Textsuche alle Verbindungen, die den Term mit der höchsten Frequenz – „rp12“ – enthalten, gesucht und visualisiert. Die Anzahl der Trajektorien wird dadurch nochmals stark eingeschränkt und aus den knapp 1100 Benutzern, die durch Berlin reisten, bleiben noch 170 übrig. Diese Zahl erscheint sehr klein, jedoch ist sie

nach genauerer Betrachtung durchaus realistisch; die meisten Besucher reisten lokal an und gehören in diesem Sinne nicht zu einer Reise, wie sie in dieser Arbeit definiert ist. Ferner ist nicht gesagt, dass die Besucher, die Twitter verwenden und sich auf der Konferenz befanden, auch darüber schrieben oder „rp12“ als Hashtag verwendeten.

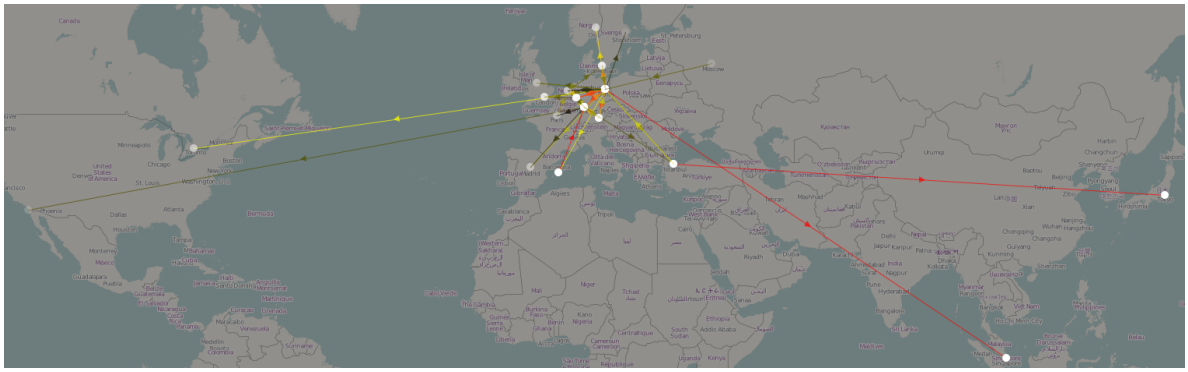

Abbildung 6.2: Übersicht über alle Reisen, die den Hashtag „rp12“ enthalten

Das textuelle Filtern nach „rp12“ schränkt die Daten folglich auch in der Visualisierung sehr ein. Abbildung 6.2 zeigt das Resultat der Filterung. Interessant ist, dass Berlin der zentrale Punkt dieser Filterung ist.

Abbildung 6.3: Sprachverteilung über Ländergrenzen hinweg

In der Übersichtskarte kann man nun deutlich erkennen, dass die meisten Besucher aus dem Raum Europa anreisten. Nur sehr wenige kamen aus dem internationalen Umfeld wie den USA, Singapur oder Japan. Dennoch ist interessant zu sehen, wie Abbildung 6.3 zeigt, dass

die Reisen über Länder hinweg durch ihre textuellen Kontextinformationen widergespiegelt werden.

Betrachtet man die Verbindung im oberen Teil der Abbildung 6.3 zwischen Manchester und Berlin, so fällt auf, dass englische Terme überwiegen. Dies lässt zwei mögliche Interpretationen zu: Die Reisenden auf diesen Verbindungen stammen entweder aus einem englischsprachigen Land wie etwa England oder sie stammen aus einer anderen Region, veröffentlichen ihre Nachrichten aber unter Verwendung der englischen Sprache. Ähnlich verhält es sich beim unteren Teil der Abbildung. Zu sehen ist die Verbindung zwischen Amsterdam und Berlin, wobei niederländische Terme überwiegen. Die Erklärung dafür ist dieselbe wie bei der Verbindung zwischen Manchester und Berlin: Entweder stammen die Reisenden aus den Niederlanden oder die Reisenden veröffentlichen ihre Nachrichten lieber auf niederländisch. Vergleicht man beide Verbindungen, so ist es bei der Verbindung zwischen Amsterdam und Berlin sehr wahrscheinlich, dass die Reisenden Niederländer sind. Niederländisch ist im Vergleich zu Englisch keine Weltsprache und es ist deshalb nicht zutreffend, dass Reisende aus anderen Ländern vorzugsweise in dieser Sprache Nachrichten veröffentlichen.

Anhand der Farbabbildung können auch die am stärksten bereisten Drehkreuze ausgemacht werden. Die Farbtabelle ist eine Interpolation von Schwarz über Gelb zu Rot, wobei Schwarz für am wenigsten bereist und Rot für am meisten bereist steht. Dementsprechend kann man in Abbildung 6.4 erkennen, dass die meisten Besucher über die Drehkreuze Frankfurt und Düsseldorf anreisten.

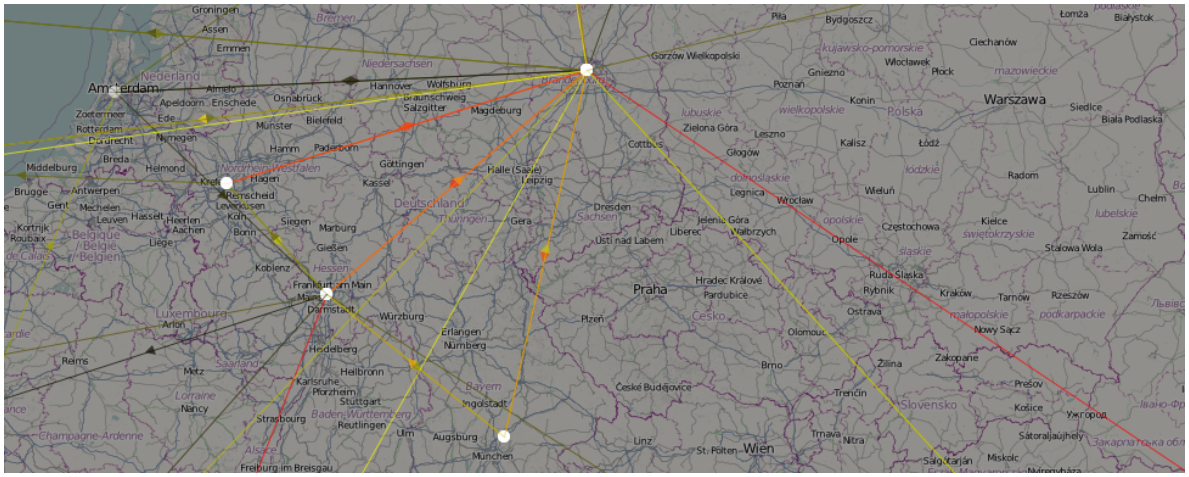

Abbildung 6.4: Farbabbildung der Reisen zur re:publica in Berlin

Durch Selektion der Verbindung zwischen Berlin und Düsseldorf wird eine neue Schlagwortwolke erstellt, die die Tweets der Reisenden auf dieser Verbindung repräsentiert. Das Resultat sind 58 der 170 dargestellten Trajektorien mit den relevanten Termen „rp12“, „berlin“, „republica“, „2012“, „hotel“, „airport“, „hauptbahnhof“. Zusätzlich zu der neu generierten

Schlagwortwolke werden der Zeitfilter sowie die Timeline-Anzeige aktualisiert. Die Timeline-Anzeige zeigt die Tweets der einzelnen Benutzer und ihren damit verbundenen Trajektorien in relativer, zeitlicher Reihenfolge zueinander. Da es in dieser Ansicht nicht möglich ist, alle 58 Einzelansichten der Trajektorien zu visualisieren, wird die farbkodierte Gesamtdarstellung gewählt. Durch die Option einzelne Nachrichten in der Timeline-Anzeige zu selektieren und somit den damit verbundenen Text und die Position auf der Karte anzeigen zu lassen, können detaillierte Informationen herausgezogen werden. Man kann feststellen, dass Besucher bereits am 1.5.2012 anreisten und in die Hotels einchecken. Des Weiteren wurden sehr oft die Schlagwörter „gate“ und „airport“ verwendet, was darauf schließen lässt, dass diese Besucher mit dem Flugzeug anreisten. Auch werden in den Tweets oft die Orte Hamburg, Düsseldorf, Berlin und Köln erwähnt, was zeigt, dass der meiste Verkehr über diese Orte verläuft. Abbildung 6.5 zeigt die Darstellung der Timeline-Anzeige. Erkennbar ist die re:publica als Ereignis; vor dem 2.5.2012 und nach dem 4.5.2012 sind Freiräume vorhanden, welche wiederum während der Veranstaltung überhaupt nicht vorhanden sind. Zu dieser Zeit waren die Besucher somit am aktivsten.

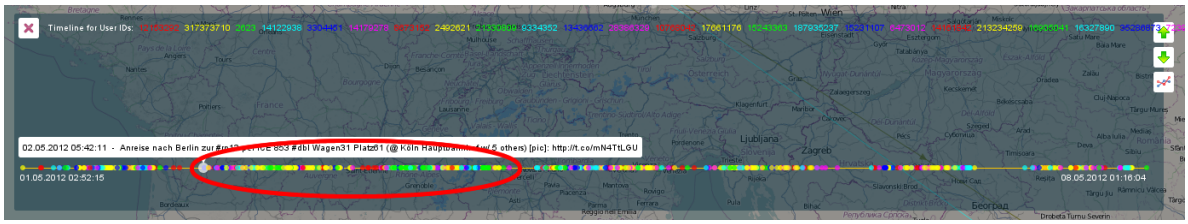

Abbildung 6.5: Darstellung der Trajektorien als Detailansicht in der Timeline-Anzeige. Zu erkennen ist die re:publica anhand des plötzlichen Anstiegs an Tweets. Unmittelbar davor und danach ist ein Leerraum von mehreren Stunden.

Auf der untersten Abstraktionsebene lassen sich einzelne Besucher analysieren. Dies wird im Folgenden mit dem Fokus auf eine Fernreise getan. Dazu wird die Verbindung nach Japan ausgewählt, und daraus ein beliebiger Benutzer ausgesucht. Abbildung 6.6 zeigt die Trajektorie und deren aggregierte Verbindungen eines einzelnen Benutzers. Auffällig ist die Abweichung der Städte, bedingt durch die Aggregation. So wird hier Dubai auf Instanbul abgebildet, weil Dubai nicht als eines der größten Drehkreuze gilt und somit in der Aggregation nicht beachtet wird. Die Reisereihenfolge ist hierbei wie folgt:

$$\text{Köln} \rightarrow \text{Berlin} \rightarrow \text{Köln} \rightarrow \text{Frankfurt} \rightarrow \text{Dubai} \rightarrow \text{Tokio}$$

Der Besucher startet in Köln seine Reise. Er betont immer wieder wie sehr Köln ihm gefällt. Daher ist Köln womöglich sein Wohnort. Als nächstes fliegt er mit dem Flugzeug nach Berlin. Dort checkt er in Kreuzberg ein und besucht die re:publica. Nach diesem Ereignis fliegt er zurück nach Köln, um am nächsten Tag mit dem Zug zum Flughafen Frankfurt zu fahren. Von dort aus geht es weiter mit dem Flugzeug über Dubai weiter nach Tokio.

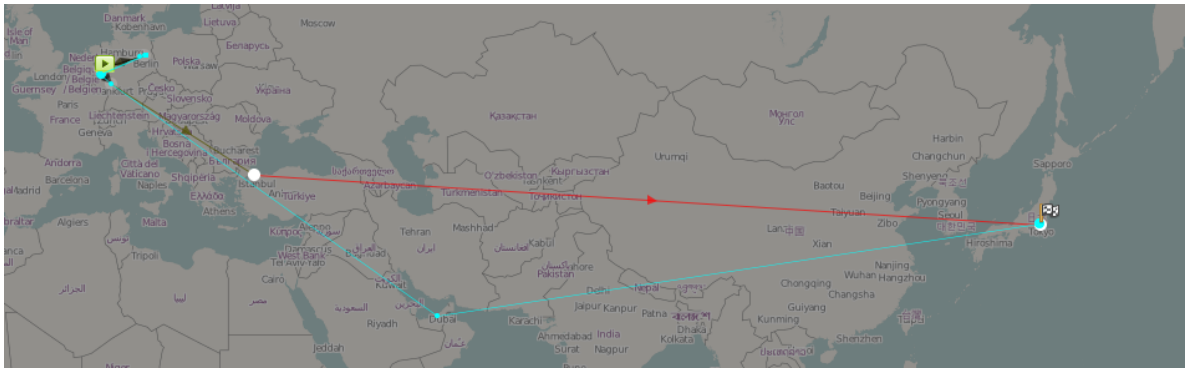

Abbildung 6.6: Darstellung einer einzelnen Benutzertrajektorie und deren aggregierte Verbindungen

Bisher wurden Reisen und Besucher zwischen den Drehkreuzen untersucht. Durch die Möglichkeit des Semantischen Zooms können auch kürzere Reisen untersucht werden. Dabei wird die Distanz der Reisen immer an die Auflösung der Anzeige angepasst. Abbildung 6.7 zeigt die kürzesten aggregierten Reisen der re:publica. Von links nach rechts: Berlin, Hamburg und Düsseldorf/Köln. Durch Vergrößern der Landkarte werden die Reisen mit einer großen Distanz ausgeblendet und diejenigen mit einer kleineren Distanz eingeblendet. Dazu wird auch die Liste der dargestellten Städte verfeinert.

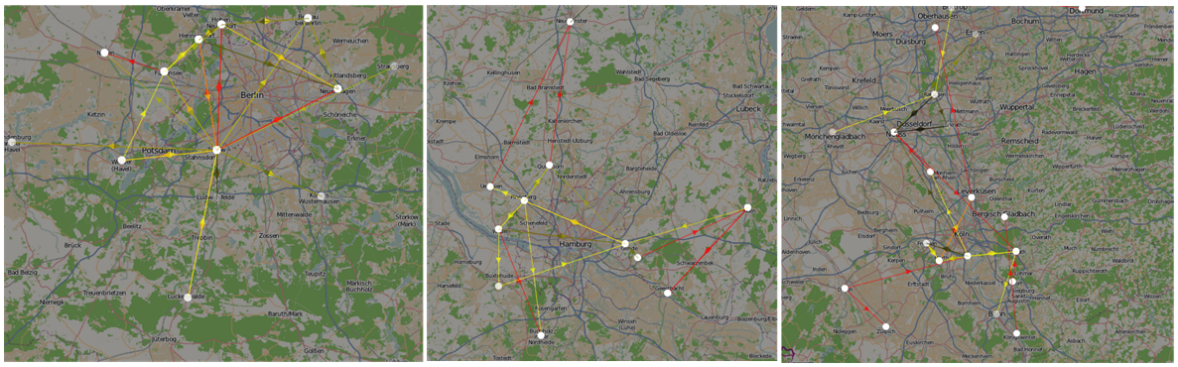

Abbildung 6.7: Anwendung des Semantischen Zooms zur Analyse von Reisen mit kürzerer Distanz. Links: Kurzreisen in Berlin. Mitte: Kurzreisen in Hamburg. Rechts: Kurzreisen im Raum Düsseldorf/Köln

Man erhält dadurch eine Verfeinerung der Reisen mit großen Distanzen. So kann das Verhalten mehrerer Benutzer etwa innerhalb Berlins analysiert werden, welche ebenfalls die re:publica besuchten: Dazu gehören unter anderem oft besuchte Stadtteile. Dadurch

kann gegebenenfalls herausgefunden werden, welche Stadtteile sehr populär und welche bei Touristen unbeliebt sind. Überdies werden so lokale Besucher sichtbar gemacht. Für dieses Ereignis spielen die lokal anreisenden Besucher nur bedingt eine Rolle, da auf der höchsten Vergrößerungsstufe es lediglich knapp 40 Benutzertrajektorien mehr sind. Anderseits kann durch Selektion der Benutzer und das Verkleinern der Landkarte von kleineren Reisen auf größere geschlossen werden. Es werden dann immer nur die Verbindungen angezeigt, die zu den ausgewählten Benutzern gehören. Interessant sind die Schlagwortwolken auf höchster und niedrigster Vergrößerungsstufe. In der niedrigsten Vergrößerungsstufe wird hauptsächlich über die re:publica geschrieben. Auf der höchsten Vergrößerungsstufe ist das Verhalten etwas anders: Lediglich die Verbindungen innerhalb Berlins besitzen noch die hoch frequentierten Terme wie „rp12“, „republica“ oder „berlin“. Andere Städte wie etwa Hamburg oder Düsseldorf besitzen diese Terme entweder überhaupt nicht oder mit einer niedrigen Frequenz. Dies hat auch mit dem Kontext zu tun. Zu dem Zeitpunkt, in welchem sich die Benutzer in einer anderen Stadt befinden, ist die Wahrscheinlichkeit, dass sie exzessiv über die re:publica schreiben, sehr niedrig.

6.1.2 Ergebnisse

Ein Ergebnis ist das Erkennen von Ereignissen anhand der aufbereiteten textuellen Information und des eingeschränkten Zeitraums. Vergleicht man beispielsweise den Datensatz rund um die re:publica mit einem Datensatz, der einen anderen Zeitraum beschreibt und kein bekanntes Ereignis enthält, so fällt auf, dass die Reisenden sehr viel über die zu besuchenden Orte schreiben und was sie dort vorhaben. Die Terme mit der höchsten Frequenz entsprechen fast immer den Städten, zwischen welchen sie reisen. Während das Vorhaben einer Reise in der Zeit der re:publica fast immer mit diesem Ereignis zu tun hatte, unterscheidet sich das Vorhaben in einer anderen Zeitspanne. Es wird immer noch viel über Flughäfen und Hotels veröffentlicht, jedoch haben die Vorhaben kein gemeinsames Ereignis. Auf einer Reise sind die Starbucks-Besuche sehr hoch frequentiert, während auf einer anderen Reise die Wahrzeichen einer Stadt im Fokus stehen.

Am Beispiel der re:publica sieht man auch, dass die Daten gegebenenfalls interessant für Eventplaner sein können. Schnell können Lokationen ausgemacht werden, die als gut besucht gelten. Auf so etwas kann in Folgejahren Rücksicht genommen werden. Da nur sehr wenige Menschen im Vergleich zu der Weltbevölkerung soziale Dienste wie Twitter benutzen, sind die Zahlen gegebenenfalls nicht aussagekräftig genug. Es ist aber durchaus denkbar, dass in Zukunft immer mehr diese Dienste verwenden und folglich auch ein besseres Profil von Ereignissen erstellt werden kann.

Durch die Verwendung der Timeline-Anzeige und der Schlagwortwolken können Ereignisse sehr gut sichtbar gemacht und somit auch erkannt werden. Darüber hinaus dient die Farbabbildung dem Sichtbarmachen von stark bereisten Orten. Auch kann dadurch und durch die Hinzunahme des Semantischen Zooms ausgemacht werden, wie viel Prozent der Besucher, die Twitter verwenden, aus dem Inland und wie viele aus dem Ausland angereist kommen.

Zwischen dem Filtern von Daten sind immer sehr hohe Wartezeiten, weswegen im Vorfeld die Daten auf Deutschland und den Zeitraum der re:publica eingeschränkt wurden. Besser wäre natürlich einen noch größeren Zeitraum zu betrachten, der auch mit anderen Reisen verbunden ist.

6.2 Ereignis: Comic-Con International: San Diego 2012

Die San Diego Comic-Con International ist die größte internationale Comic-Messe und ist auch für die Vorstellung neuer Filme und Serien im Comic-Genre bekannt. Im Jahr 2012 wurden etwa 125 000 Besucher aus internationalem sowie nationalem Umfeld erwartet. Vom 12.7.2012 bis zum 15.7.2012 fand die 43. Veranstaltung statt, die vor allem wegen ihres Ausmaßes für diese Diplomarbeit von Interesse ist. Des Weiteren handelt es sich um eine Veranstaltung, die sich im Gegensatz zur re:publica nicht den sozialen Medien widmet. Deshalb ist es interessant zu untersuchen, inwieweit das Ereignis mit Hilfe von Twitterdaten analysiert werden kann oder wie viel Information darüber in sozialen Medien veröffentlicht wird und öffentlich zugänglich ist. Der weitere Aufbau dieses Kapitels ist wie folgt: Über die Übersichtskarte wird versucht, Informationen herauszufinden, die in direktem Kontakt zur Comic-Con stehen. Auf dieser Basis werden die Zeiträume unmittelbar vor, nach und während der Veranstaltung analysiert, um gegebenenfalls Veränderungen in den aggregierten textuellen Informationen erkennen zu können. Darüber hinaus ist von Interesse, welche Informationen über die Umgebung während des Ereignisses in San Diego aus dem Datensatz extrahiert werden können. Wie bei der Analyse der re:publica zeigen die Daten lediglich einen kleinen Ausschnitt. Die erzielten Ergebnisse stehen rein beispielhaft für die gesammelten Daten.

Der verwendete Datensatz reicht vom 8.7.2012 bis zum 21.7.2012, sodass ausreichend Informationen aus der Zeit vor und nach dem Ereignis in die Analyse miteinbezogen werden können. Geographisch gesehen ist der aggregierte Datensatz ausschließlich auf die Geokoordinaten von San Diego eingegrenzt. Es werden nur Trajektorien zugelassen, die mindestens einen Stopp in San Diego haben. Wird ein größeres Gebiet gewählt, so wird die Analyse negativ beeinflusst, da zu viele Daten aggregiert und visualisiert werden müssen.

6.2.1 Analyse der Comic-Con

Um nähere Informationen zur Comic-Con zu erhalten, werden Schlagwörter benötigt, die im direkten Zusammenhang zum Ereignis stehen. In der Zeit vom 8.7.2012 bis 21.7.2012 sind etwa 2 Millionen Benutzerprofile im gesammelten Datensatz enthalten. Durch die Aggregation der Daten und der damit verbundenen geographischen Einschränkung bleiben auf der Übersichtskarte 1320 Benutzer übrig. Durch die Einschränkung des betrachteten Zeitraums auf die Veranstaltungstage werden auf der Übersichtskarte die aggregierten Trajektorien von 751 Benutzern angezeigt. Es kann nun auf zwei verschiedenen Wegen herausgefunden werden, welches die relevanten Schlagwörter sind: durch Generierung der Schlagwortwolke für alle Verbindungen, die über San Diego verlaufen, und durch

Generierung der Schlagwortwolke für die aktivste Verbindung, die durch die Farbabbildung zu erkennen ist.

Abbildung 6.8: Generierte Schlagwortwolken für die Analyse der Comic-Con. (a) Schlagwortwolke für alle Trajektorien, die mindestens einen Stopp in San Diego hatten. (b) Schlagwortwolke für die aktivste Verbindung nach San Diego.

Abbildung 6.8 zeigt beide generierten Schlagwortwolken. Die linke Schlagwortwolke (a) repräsentiert alle Trajektorien, die mindestens einen Stopp in San Diego zur Zeit der Veranstaltung hatten. Die rechte Schlagwortwolke (b) repräsentiert die aktivste Verbindung im Veranstaltungszeitraum. Deutlich zu sehen sind die Schlagwörter mit der höchsten Frequenz: „san diego“, „sdcc“, gefolgt von „comiccon“. Durch Einsatz des Textfilters wird die restliche Analyse dieses Ereignisses unter der Filterung des Schlagworts „comiccon“ und den damit verbundenen 102 Benutzern in der Übersichtskarte durchgeführt.

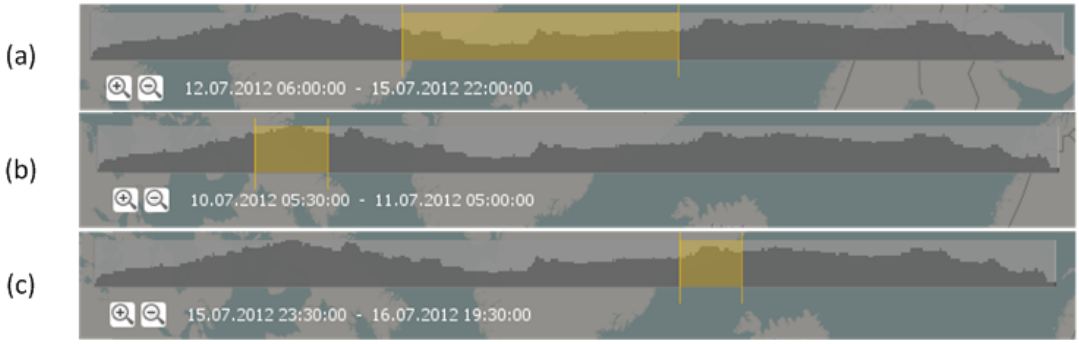

Abbildung 6.9: Zeitfilter und aktive Verbindungen über den aggregierten Zeitraum hinweg. In (a) ist der Zeitraum der Comic-Con selektiert, in (b) der Tag davor und in (c) der Tag unmittelbar danach. Das hinterlegte Histogramm zeigt die Anzahl der aktiven Verbindungen.

Auffällig ist, dass zum Zeitpunkt der Comic-Con die Anzahl der aktiven Verbindungen zurückgeht, jedoch am Tag davor und danach extrem hoch ist, wie in Abbildung 6.9 zu sehen

ist. (a) zeigt den Zeitraum des Ereignisses, (b) den Tag davor und (c) den Tag unmittelbar danach. Wir versuchen zu verstehen, was der Unterschied zwischen diesen Zeitbereichen ist und betrachten hierzu die Schlagwortwolken, welche direkt in die Karte auf die markierten Verbindungen gezeichnet werden. Dies hat den Vorteil, dass die Schlagwortwolke ausschließlich die Reise mit Start- und Zielort beschreibt, anstatt wie bei der Selektion eine komplette Trajektorie.

Wie in Abbildung 6.10 zu sehen ist, lassen nur wenige hoch frequentierte Terme auf das Stattfinden der Comic-Con schließen. Ferner kann man sehen, dass die Anzahl an Verbindungen in der Übersichtskarte deutlich reduziert ist. Abbildung 6.11 zeigt hingegen dasselbe Bild einen Tag vor und einen Tag nach der Comic-Con. Es gibt wesentlich mehr aktive Verbindungen und auch die Terme der Schlagwortwolken weisen deutlich auf das Ereignis hin. In den meisten Fällen ist am Tag vor der Comic-Con unter den relevanten Termen ein Hinweis auf die Comic-Con, einen Flughafen oder auf die Stadt San Diego. Am Tag nach dem Ereignis verhält es sich ähnlich, jedoch gehören Hinweise auf Hotels und Flughäfen nicht mehr zu den relevanten Termen. Dies lässt darauf schließen, dass über das Erlebte berichtet wurde, anstatt über das Einchecken im Hotel oder das Ankommen am Flughafen.

Abbildung 6.10: Schlagwortwolken, visualisiert auf den Verbindungen während der Comic-Con 2012

Es gibt viele unterschiedliche Erklärungen für die Verteilung des Informationsgehalts über die Zeiträume; hier werden nachfolgend zwei erläutert. Untersucht werden ausschließlich Ergebnisse, die Benutzertrajektorien enthalten, welche direkt die Comic-Con erwähnen. Daher ist eine mögliche Erklärung, dass sich die Benutzer während des Ereignisses auf der Comic-Con befanden, anstatt zu reisen. Dies erklärt auch die erhöhten Werte von aktiven Verbindungen unmittelbar vor und nach dem Ereignis: das sind die Zeiten der An- und Abreise.

Vergleicht man das Ereignis mit einem simplen Kinogang, ergibt sich eine weitere mögliche Erklärung. Unmittelbar vor der Filmvorführung beschäftigt man sich mit der Anreise und

freut sich auf das anstehende Ereignis. Man möchte seinen Freunden mitteilen, was man empfindet und was man erleben wird. Während man sich im Kino befindet, wird die Filmvorführung genossen und es existiert keinerlei Mitteilungsbedarf nach außen. Aus diesem Grund geht die Anzahl an Mitteilungen und Bewegungen zu diesem Zeitpunkt zurück. Nach der Filmvorführung wird durch persönliche Mitteilungen und Veröffentlichungen das Erlebte beispielsweise in Form einer Rezension reflektiert.

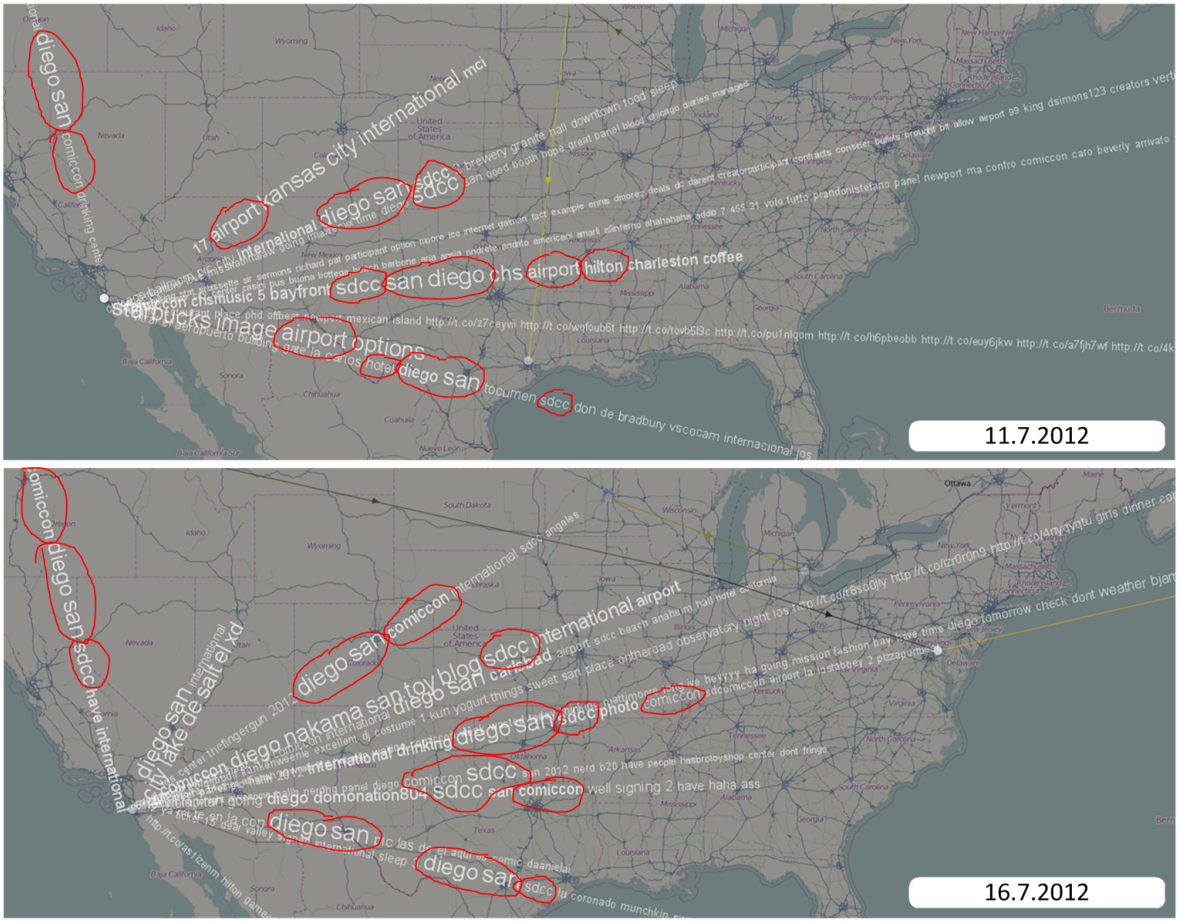

Abbildung 6.11: Schlagwortwolken, visualisiert auf den Verbindungen vor und nach der Comic-Con 2012

Von Interesse sind auch Informationen, die über Reisende herausgefunden werden können. So kann man sich fragen, ob es im Zeitraum der Comic-Con andere Städte außer San Diego gibt, die von potenziellen Besuchern stark bereist wurden. In der Übersichtskarte kann mit Hilfe der Farbabbildung erkannt werden, welche Verbindungen aktiver sind als andere, jedoch nicht, wie das Reiseverhalten innerhalb der Stadt ist. Mit Hilfe des Semantischen Zoom, also durch Hinzunahme von Städten und Stadtteilen, können stark bereiste Landteile

anhand der Städtedichte ausgemacht werden. Ferner wird die Reiseentfernung eingeschränkt; je weiter die Karte vergrößert wird, umso kürzere Reisen werden visualisiert. Abbildung 6.12 zeigt die Karten-Ansicht ab der Vergrößerungsstufe elf. Die Reduzierung der Reiseentfernung sowie die vermehrte Darstellung an Städten und Stadtteilen ist deutlich zu sehen. Man kann erkennen, dass hauptsächlich die Bewegung innerhalb der USA stattfindet, jedoch ist der Schwellwert für die Städte noch zu hoch, sodass auch bei dieser Vergrößerungsstufe nur ein Überblick erlangt werden kann.

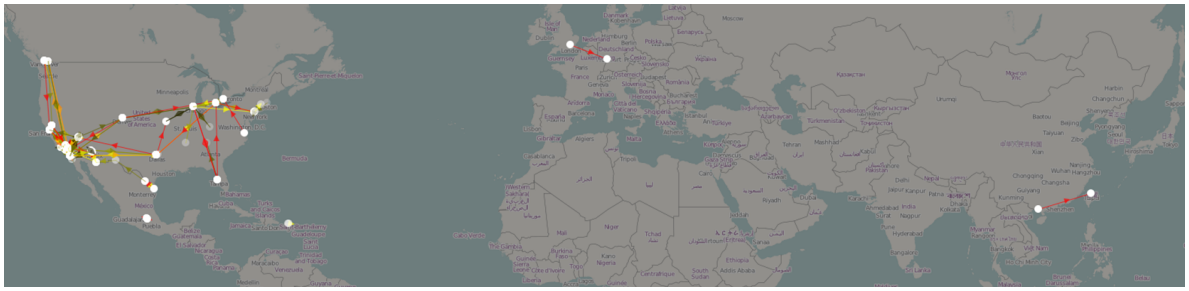

Abbildung 6.12: Visualisierung von Bewegungsdaten ab der Vergrößerungsstufe elf

Anders verhält es sich ab Vergrößerungsstufe acht (siehe Abbildung 6.13). Im Raum einiger Großstädte kann anhand der Dichte ein erhöhtes Reiseverhalten beobachtet werden. Da wie zuvor festgestellt fast die komplette Bewegung innerhalb der USA stattfindet, liegt hierauf auch der Fokus. Zu sehen ist, dass sich Städtecluster gebildet haben. Die Dichte an bereisten Städten und Stadtteilen ist besonders hoch bei:

San Diego, Tijuana, Los Angeles, San Francisco, Santa Clara, New York und Chicago.

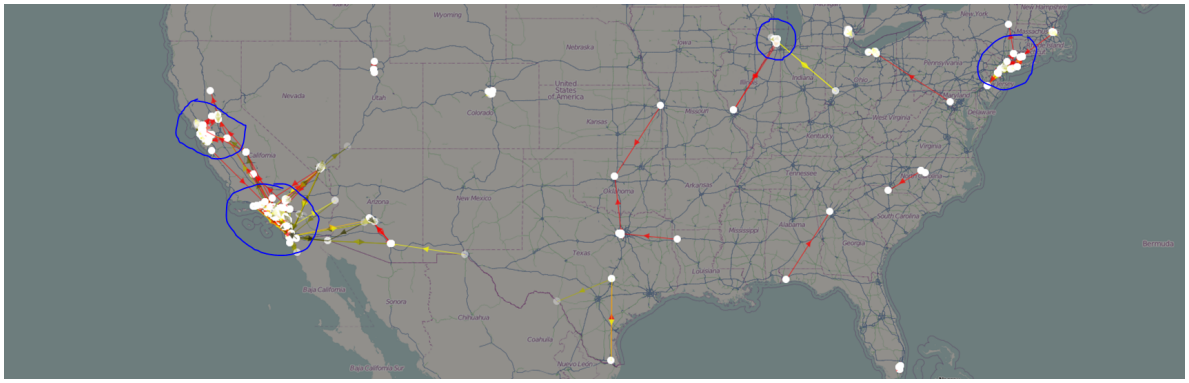

Abbildung 6.13: Visualisierung von Bewegungsdaten ab der Vergrößerungsstufe acht

6.2.2 Ergebnisse

Durch die erwarteten 125 000 Besucher der Comic-Con spekuliert man auf ein großes, internationales Publikum, das im Datensatz vorhanden ist. Besucher, die via Twitter über das Ereignis Daten veröffentlichten, reisten hauptsächlich aus den USA an. Durch Anwendung des Semantischen Zoom wird auch deutlich gemacht, welche Städte von genau diesen Besuchern sehr stark bereist wurden. Des Weiteren können so auch viel bereiste Sehenswürdigkeiten oder sehr beliebte Stadtteile ausgemacht werden.

Anhand von Schlagwortwolken, die direkt auf Verbindungen zwischen zwei Städten visualisiert werden, können nicht nur die aggregierten textuellen Informationen von ganzen Reisen betrachtet werden, sondern auch von Teilreisen, die nur einen festgelegten Abschnitt auf der Reise widerspiegeln. Werden Ereignisse in einer bestimmten Stadt betrachtet, so wird das Herausfinden von Kontextinformationen wesentlich durch diese Technik vereinfacht.

Durch die Anwendung des Zeitfilters in Kombination mit der Analyse durch Schlagwortwolken können Ereignisse zeitabhängig untersucht werden. Dies kann zu einem tieferen Verständnis führen, denn einerseits können die Kontextinformationen eines überblickbaren Zeitraums und anderseits eines spezifischeren, detaillierteren Zeitraums betrachtet werden.

6.3 Vergleich des Reiseverhaltens

In dieser Fallstudie wird untersucht, ob Bewegungsmuster erkennbar und erklärbar sind, ohne sich auf ein gewisses Ereignis zu konzentrieren. Hierzu werden normale Werktage mit Wochenendtagen verglichen. Ein normaler Werktag ist beispielsweise Dienstag oder Mittwoch. Dies resultiert aus der Tatsache, dass z. B. Freitag und Montag noch eng mit dem Wochenende zusammenhängen. Freitags hört man vielleicht ein bisschen früher auf zu arbeiten, um ins Wochenende zu starten und am Montag erscheint man vielleicht ein bisschen später, um das Wochenende ausklingen zu lassen. Aus diesem Grund werden hier hauptsächlich die Tage Dienstag/Mittwoch und Samstag/Sonntag gegenübergestellt.

Der hierzu verwendete Datensatz deckt den Zeitraum von Dienstag, den 14.8.2012 bis Montag, den 20.8.2012 ab. Die Daten sind geographisch eingeschränkt auf Besuche oder besser gesagt Stopps in Deutschland, d.h. Trajektorien, die keine Stopps in Deutschland aufweisen können, werden verworfen. Dies hat unter anderem den Vorteil, dass der Datensatz kontrollierbar ist im Sinne der Datenmenge. Falls der Datensatz textuell eingegrenzt werden soll, können bekannte deutsche Schlagwörter verwendet werden. Sollte beispielsweise nach Sehenswürdigkeiten eingegrenzt werden, so kennt sich ein einheimischer Deutscher bedeutend besser in Deutschland als in Mexiko aus.

6.3.1 Vergleich der Werktage zum Wochenende

Um ohne die Anwendung einer selektiven Filterung die Daten miteinander zu vergleichen, werden zunächst die Tage gegenübergestellt. Dazu werden die Daten über den Zeitfilter auf

die entsprechenden Tage gefiltert. Zudem werden die Daten über den Semantischen Zoom in der Vergrößerungsstufe acht betrachtet. Dies sind Reisen über Städte mit mehr als 50 000 Einwohnern. Zu beachten ist, dass die in diesem Kapitel gezeigten Histogramme innerhalb des Zeitfilters auf den jeweils gefilterten Datensatz normalisiert sind. Daher wird für einen Mengenvergleich immer die Anzahl an individuellen Trajektorien herangezogen.

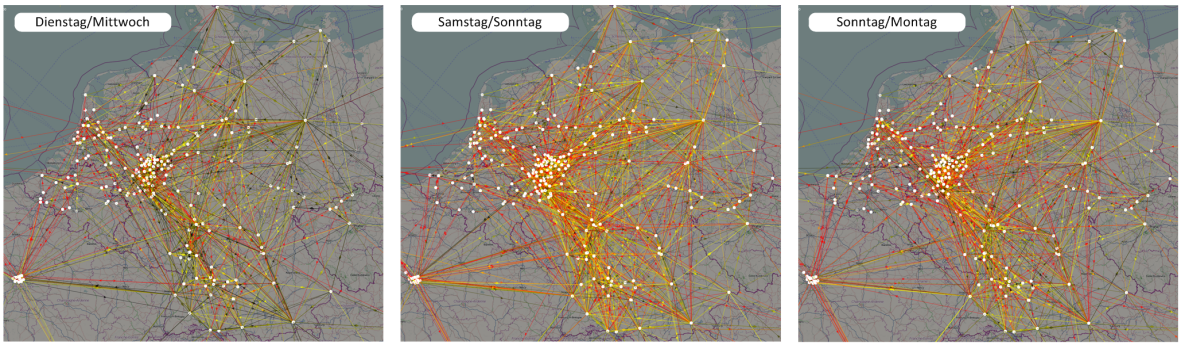

Abbildung 6.14: Vergleich der Reisemenge zwischen Werktagen und Wochenende

Ursprünglich besitzt der verwendete Datensatz 4493 Trajektorien. Abbildung 6.14 zeigt drei verschiedene Ausschnitte: Links sind die Daten für die Tage Dienstag und Mittwoch zu sehen mit insgesamt 2011 übriggebliebenen Trajektorien. In der Mitte werden die Tage Samstag und Sonntag mit insgesamt 3087 Trajektorien und rechts die Tage Sonntag und Montag mit 2518 Trajektorien visualisiert. Zunächst fällt durch die Farbabbildung auf, dass am Dienstag und Mittwoch weniger Reisen auftreten, was sich auch in den Zahlen widerspiegelt. Je näher die Farbe an der Farbe Rot ist, umso relevanter ist die Reise im Sinne der Anzahl an Menschen oder anhand der dargestellten Zeitspanne. Unter anderem ist auch zu sehen, dass zwischen Deutschland, den Niederlanden und Paris eine gehäufte Anzahl an Verbindungen vorhanden ist. Dieses Dreieck ist am Wochenende aktiver.

Der erhoffte Mehrwert bleibt bei dieser Analyse jedoch aus, da keine bedeutenden Unterschiede zu erkennen sind und sich demnach auch nicht beurteilen und erklären lassen. Aus diesem Grund wird nachfolgend der Datensatz einmal anhand von Schlagwörtern, welche bei vielen Deutschen ein Wochenende ausmachen, und einmal anhand von beliebten Kurzurlaubszielen textuell eingeschränkt.

Textuelle Einschränkung der Daten

Bei der textuellen Filterung werden ausschließlich Reisen angezeigt, die an einer beliebigen Stelle der verwendeten Tweets den gesuchten Text enthalten. Im ersten Teil werden die Reisen nach Schlagwörtern gefiltert, welche im deutschen Sprachgebrauch oft in Zusammenhang mit dem Wochenende verwendet werden. Repräsentative Schlagwörter, unter anderem auch

für einen Sommermonat wie August, sind beispielsweise: „wochenende“, „feierabend“, „party“, „kino“, „meer“, „auto“ oder „autobahn“.

Abbildung 6.15: Filterung des Datensatzes nach den Schlagwörtern „wochenende“ und „kino“

Abbildung 6.15 zeigt die Filterung für die Schlagwörter „wochenende“ und „kino“. Es bleiben 47 Trajektorien für Menschen, die das Wochenende in ihren Veröffentlichungen erwähnt haben, übrig. Der zugehörige Zeitfilter zeigt den vollen Zeitraum, also von Dienstag bis Montag. Bei Betrachtung des enthaltenen Histogramms bemerkt man, dass besonders viele aktive Reisen zwischen Mittwoch und Donnerstag sowie zwischen Freitagmittag und Sonntagnacht stattfinden. Diese Bereiche werden in der Abbildung in Orange hervorgehoben. Dass Menschen, die über das Wochenende schreiben, vermutlich auch am Wochenende aktiver sind, lässt sich logisch schlussfolgern, jedoch nicht wieso sie zwischen Mittwoch und Donnerstag aktiver sind. Ähnlich ist es bei dem Schlagwort „kino“. Es bleiben hier 30 Trajektorien nach der Filterung übrig. Deutlich zu sehen ist der Anstieg an aktiven Reisen im Zeitraum von Freitagmittag bis Samstagnacht. Der eigentliche, korrekte Zeitpunkt des Kinogangs bleibt jedoch verborgen und muss anhand einer Einzelanalyse der Benutzer durchgeführt werden. Anhand der Farbabbildung können in beiden dargestellten Landkarten die aktivsten Verbindungen ausgemacht werden.

Im nächsten Schritt wird ein Reiseverhalten einem beliebten Urlaubsziel im Sommer gegenübergestellt. Für dieses Reiseverhalten eignen sich die Schlagwörter „autobahn“ und „meer“. Die Autobahn ist ein beliebter Weg sein Ziel zu erreichen und im Sommer ist das Meer ein mögliches Ausflugsziel.

Abbildung 6.16: Gegenüberstellung von Reiseverhalten und Urlaubsort

Abbildung 6.16 zeigt diese Gegenüberstellung. Es sind 54 individuelle Reisen, die in irgendeiner Form das Meer beinhalten. Sie sind am aktivsten in der Zeit von Donnerstagmittag bis Sonntagabend. Im Gegensatz dazu bleiben 30 Trajektorien übrig, die die Autobahn in ihren Nachrichten erwähnen. Wie zu sehen ist, sind die Überschneidungen hauptsächlich Donnerstagmittag bis Freitagabend und Sonntagabend. Dies ist folgendermaßen interpretierbar: Donnerstag oder Freitag beschlossen die Kurzurlauber ans Meer zu fahren, dort bis Sonntag zu bleiben und anschließend zurückzureisen.

Außerdem fällt bei der Betrachtung der beiden Landkarten auf, dass die Ziele häufig an der holländischen Küste liegen. Anhand der Farbabbildung kann man obendrein sehen, dass dies gleichzeitig die aktivsten Verbindungen sind.

Berlin und Hamburg sind beliebte deutsche Kurzurlaubsziele im Bereich der Städtereisen. Im Folgenden wird versucht, durch Filterung nach diesen zwei Städten Näheres über das Reiseverhalten herauszufinden.

Abbildung 6.17 zeigt die gefilterten Daten für die Schlagwörter „berlin“ und „hamburg“. Bei beiden Landkarten ist die sehr aktive Anzahl an Verbindungen zwischen Berlin beziehungsweise Hamburg und dem Ruhrgebiet auffällig. Es bleiben außerdem 420 Trajektorien für „berlin“ und 220 Trajektorien für „hamburg“ nach der Filterung übrig. Am Wochenende steigt zwar die Anzahl an aktiven Verbindungen, wie der orangen Hervorhebung im Zeitfilter zu entnehmen ist, jedoch nicht bedeutend. Da viele Verbindungen trotz Filterung übriggeblieben sind, ist im Histogramm die Verteilung relativ konstant. Dies folgt aus dem bereits erwähnten Problem der Reiseverteilung und der hohen Anzahl an Verbindungen. Daher sollten diese Verbindungen auf der Landkarte analysiert werden.

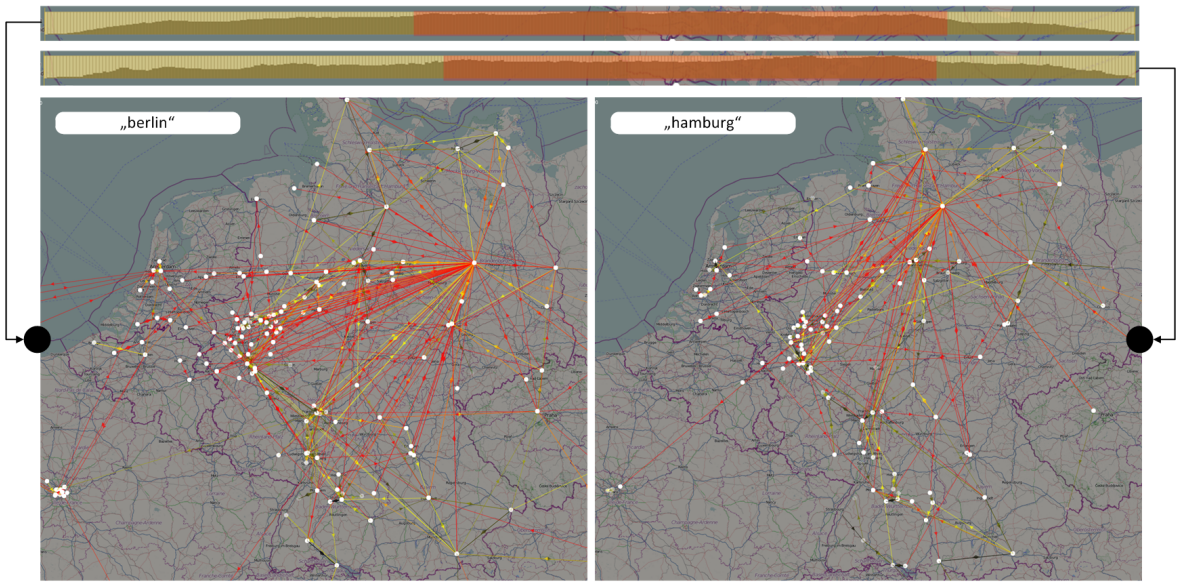

Abbildung 6.17: Beliebte deutsche Kurzurlaubsziele

6.3.2 Ergebnisse

In der Fallstudie wurde gezeigt, dass die Anwendung sich nicht für alle Szenarien eignet. Bei enormen Datenmengen sind Veränderungen des Reiseverhaltens aufgrund der gewählten Datenstruktur kaum oder überhaupt nicht zu erkennen. Dasselbe Problem entsteht, wenn trotz Filterung zu viele Trajektorien übrigbleiben. Der Grund dafür ist derselbe.

Bei einer Filterung mit einem entsprechenden Ergebnis können jedoch durchaus Bewegungsmuster erkannt werden, wie am Beispiel von einigen Schlagwörtern zu sehen war. So können bei Reisen in Verbindung mit bestimmten Schlagwörtern, wie beispielsweise „wochenende“ oder „kino“ , Unterschiede erkannt werden. Bei diesen Schlagwörtern steigt am Wochenende deutlich die Reiseaktivität.

6.4 Diskussion

Diese Fallstudie hat Stärken und Schwächen der implementierten Visualisierung zur Analyse von Bewegungsdaten aufgezeigt. Anhand der Ereignisse re:publica und Comic-Con wurde gezeigt, wie mit einer überlegten Filterung in Kombination mit annotierten Kontextinformationen in Form von verschiedenen Schlagwortwolken diese Ereignisse mit ihren zugehörigen Informationen schnell und einfach gefunden werden können. Anhand verschiedener Ansichten, wie die Timeline-Anzeige oder die Einzel-Trajektorien-Ansicht, konnten nähere Informationen zu den Besuchern herausgefunden und Rückschlüsse auf ihr Reiseverhalten

gezogen werden. Mit Hilfe des Zeitfilters wurden relevante Zeitpunkte identifiziert und auf dieser Basis die Daten miteinander verglichen. Durch Schlagwortwolken in globaler und lokaler Form können Einzelverbindungen oder vollständige Reisen mit Hilfe der textuellen Kontextinformation analysiert werden. So wurde sichtbar gemacht, dass Menschen aus einem anderen Land auch tatsächlich diese Sprache in ihren Veröffentlichungen verwenden. Ferner konnten allein anhand der Schlagwortwolke, die unmittelbar auf der Verbindung visualisiert wird, relevante Ereignisse sowie Städte, die bereist wurden, ausgemacht werden. Anhand der gewählten Datenstruktur über Zeitintervalle wurde auch der Unterschied zwischen den generierten Schlagwortwolken deutlich. So verändern sich beispielsweise mit der Verschiebung der Zeit auch die Schlagwörter auf einer Verbindung.

Außerdem wurde aufgezeigt, was nicht möglich ist: die zeitliche Analyse eines sehr großen Datensatzes ohne den Einsatz von Filtern sowie die zeitliche Analyse eines gefilterten Datensatzes, der zu groß ist. Wird die Filterung jedoch entsprechend zielgerichtet eingesetzt, so sind auch rein zeitliche Analysen möglich.

Man darf nicht vergessen, dass die verwendeten Daten lediglich einen kleinen Ausschnitt aus dem Twitter-Datensatz sowie aus der Menge an Menschen, die soziale Medien und Netzwerke verwenden, zeigen.

7 Zusammenfassung und Ausblick

Es gibt bereits einige Ansätze, Daten aus sozialen Medien automatisiert zu analysieren. In dieser Arbeit lag der Fokus jedoch nicht auf der Analyse einzelner Twitternachrichten, sondern auf der Analyse von Benutzertrajektorien, welche als Konkatenation mehrerer Twitternachrichten eines Benutzers in zeitlich korrekter Reihenfolge definiert sind. Die Realisierung der Problemstellung wurde in drei Teilen durchgeführt: die Vorverarbeitung der Daten, die Visualisierung der Daten unter Hinzunahme von verschiedenen Filter- und Selektionsmöglichkeiten und die Evaluation durch eine Fallstudie. Hintergrund war, mehr über die Semantik der Twitterdaten oder der damit verbundenen Reisen zu erfahren. Bei jeder der drei Teilaufgaben auf dem Weg zur Realisierung stand die Problemstellung im Vordergrund, ob sich Bewegungsmuster in den Daten mittels interaktiver Visualisierungen erkennen und durch visuelle Annotation basierend auf Kontextinformation beurteilen und erklären lassen.

Im Rahmen der Findung von geeigneten Datenstrukturen zur effizienten Aggregation und Visualisierung wurden für die Lösung die Zeitkomponente sowie die Darstellung der Daten in einem globalen wie auch lokalen Umfeld beachtet. Da die Landkarte für die Visualisierung in mehrere Vergrößerungsstufen unterteilt ist, wird für bestimmte Stufen eine Liste an Städten, abhängig von der Größe festgelegt, die visualisiert werden. Dazu wurden im ersten Schritt, der Vorverarbeitung der Daten, die Tweets eines Benutzers der nächstgelegenen Stadt aus dieser Liste zugeordnet. Anhand des Zeitstempels können so Reisen zwischen Städten aus diesen Listen ausgemacht werden, sobald ein Tweet einer anderen nächstgelegenen Stadt zugeordnet wird. Die Reise wird in gleichgroße Zeitintervalle unterteilt, wobei jedes Zeitintervall eine Gewichtung erhält. Diese Gewichtung sowie die Gesamtgewichtung der Reise vergrößern sich, falls mehrere Benutzer auf dieser Route während gleichen Zeitintervallen reisen. Mit diesem Ansatz wurden zwei Probleme behoben: Zum einen erscheinen und verschwinden Verbindungen nicht plötzlich, falls mehrere Benutzer auf derselben Route vertreten sind. Mit diesem Ansatz sind die Länge einer Reise und die in der Zeitfilterung enthaltenen Zeitintervalle bedeutend. Zum anderen werden Überdeckungen in der Visualisierung bedeutend eingeschränkt. Nicht nur die Tweets tragen zur Relevanz einer Reise bei, sondern auch die Benutzer. Für eine Reise sind alle Tweets, die am Abreisetag und am Ankunftstag veröffentlicht wurden, relevant. Mit Hilfe der Termfrequenz werden relevante Schlagwörter herausgefiltert und den einzelnen Zeitintervallen zugeordnet, was eine kontextabhängige Filterung zulässt.

Für die Ermöglichung geeigneter Strategien zur Exploration, Filterung und Selektion der annotierten Daten wurden verschiedene Konzepte realisiert. Unterteilt werden diese Konzepte in vier Kategorien: Textfilter, Zeitfilter, Userfilter und Kartenfilter. Mit Hilfe des Textfilters können die Daten anhand des textuellen Inhalts gefiltert werden. Nach der Filterung bleiben

die Reisen bestehen, die in ihren Tweets gegebenen Text enthalten. Durch Anwendung des Zeitfilters werden die Reisen nach Zeitintervallen gefiltert und durch die Farbabbildung entsprechend gekennzeichnet. Der Userfilter dient der Einschränkung der Benutzer, deren Reisen dargestellt werden. Filtern lassen sich die Benutzer entweder rein textuell oder zahlenmäßig. Der Kartenfilter enthält unter anderem auch verschiedene Möglichkeiten zur Selektion. Die visualisierten Bewegungsdaten können mit Hilfe verschiedener Kartenfilter eingeschränkt werden. Dazu gehören sowohl die Auswahlselektion als auch die Einzelselektion von Reisen. Durch das implementierte Filterframework können die soeben erwähnten Filter in beliebiger Reihenfolge auf den Datensatz angewendet oder entfernt werden. Zusätzlich wird die Analyse von Reisen durch die Anzeige von Schlagwortwolken, die entweder direkt auf eine Verbindung oder auf komplette Reisen bezogen sind, durch die Timeline-Anzeige zur Erkennung von Ereignissen und Einzeltrajektorien unterstützt.

Anhand der Evaluation der Realisierung durch eine Fallstudie wurde beschrieben, für welche Problemstellungen das Programm geeignet ist und für welche nicht. Durch die Analyse der re:publica und der Comic-Con wurde gezeigt, dass Ereignisse sich schnell finden lassen und nähere Informationen mit Hilfe der realisierten Interaktionskonzepte herausgefunden werden können. Zu den Grenzen des implementierten Programms gehört die Analyse von Reiseverhalten anhand der Gesamtübersicht, unter Verwendung des Histogramms, welches die Anzahl an aktiven Verbindungen pro Zeitintervall widerspiegelt. Dadurch, dass Reisen über ganze Zeiträume hinweg dargestellt werden, ist das punktuelle Auffinden von einem beispielsweise erhöhten Nachrichtenverkehr nicht möglich.

Die Analyse sowie die Aggregation der Daten ist sehr zeitintensiv, was jedoch nicht an der realisierten Struktur liegt, sondern vielmehr daran, dass eine relationale Datenbank und eine Festplatte mit SATA-Schnittstelle verwendet wurden. Darüber hinaus spielt die Größe des Datensatzes eine entscheidende Rolle. Aus diesem Grund wurden in der Fallstudie bereits gefilterte Daten verwendet, um die Visualisierung zu beschleunigen. Werden in einer relationalen Datenbank z. B. Indices zur Beschleunigung der Anfragen eingesetzt, so ist die Aggregation langsamer. Es gilt hier abzuwägen, was wichtiger ist. Die Zeitkomponente ist somit verbesserungswürdig, beispielsweise durch die Verwendung dokumentbasierter Datenbanken oder SSD-Festplatten. Dabei zieht ein Umstieg auf dokumentbasierte Datenbanken auch eine Umstellung der Datenstruktur nach sich. In Zukunft sollte sowohl dieses Problem gelöst als auch Erweiterungen vorgenommen werden. So könnte die Benutzerfreundlichkeit des Programms durch entsprechende Benutzerstudien verbessert werden. Ebenso könnten Bewegungsdaten noch besser durch Hinzunahme von veröffentlichten Fotos, Videos und den Freunden des Benutzers analysiert werden, um mehr über das Umfeld herauszufinden.

Literaturverzeichnis

[AA08] G. Andrienko, N. Andrienko. Exploration of Massive Movement Data: a Visual Analytics Approach. 2008. (Zitiert auf Seite 27)

[AFM+07] L. O. Alvares, J. A. Fernandes, D. Macedo, V. Bogorny, B. Moelans, B. Kuijpers, A. Vaisman. A model for enriching trajectories with semantic geographical information. In *ACM-GIS*. Press, 2007. (Zitiert auf Seite 37)

[BMZ10] J. Bollen, H. Mao, X.-J. Zeng. Twitter mood predicts the stock market. *CoRR*, abs/1010.3003, 2010. (Zitiert auf Seite 11)

[BS08] I. Bronstein, K. Semendjajew. *Taschenbuch der Mathematik*. Deutsch Harri GmbH, 2008. URL http://books.google.de/books?id=VnsL9p8hXfQC. (Zitiert auf Seite 20)

[Cal12] M. C. Calzolari. Analysis of Twitter followers of leading international companies. Quantitative and qualitative study of behaviours demonstrated by humans or by bots. Technischer Bericht, IULM University of Milan, 2012. (Zitiert auf Seite 16)

[Def07] D. of Defense. Global Positioning System Precise Positioning Service Performance Standard. Technischer Bericht, Department of Defense, 2007. URL http://www.gps.gov/technical/ps/2007-PPS-performance-standard.pdf. (Zitiert auf Seite 19)

[Def08] D. of Defense. Global Positioning System Standard Positioning Service Performance Standard. Technischer Bericht, Department of Defense, 2008. URL http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf. (Zitiert auf Seite 19)

[DH09] H. Dodel, D. Häupler. *Satellitennavigation*. Springer, 2009. URL http://books.google.de/books?id=l_OPEUEg2MwC. (Zitiert auf den Seiten 18 und 19)

[Ert10] T. Ertl. Visualization Course Slides, 2010. URL http://www.vis.uni-stuttgart.de/. (Zitiert auf den Seiten 7, 14, 15, 21 und 22)

[FB74] R. A. Finkel, J. L. Bentley. Quad Trees: A Data Structure for Retrieval on Composite Keys. *Acta Inf.*, 4:1–9, 1974. (Zitiert auf Seite 24)

[Fei] J. Feinberg. URL http://www.wordle.net/. (Zitiert auf den Seiten 7, 25 und 26)

[Fis04] J. Fisher. Visualizing the Connection Among Convex Hull, Voronoi Diagram and Delaunay Triangulation. Technischer Bericht, Michigan Technological University, 2004. (Zitiert auf den Seiten 7 und 23)

[geo] URL http://www.geonames.org/. (Zitiert auf den Seiten 17 und 34)

[GMSK08] B. Guc, M. May, Y. Saygin, C. Körner. Semantic Annotation of GPS Trajectories. In *Proc. of the 11th AGILE International Conference on Geographic Information Science (AGILE08)*. 2008. (Zitiert auf Seite 28)

[Gra11] G. Graefe. *Modern B-Tree Techniques*. Now Publishers, 2011. URL http://books.google.de/books?id=AiOj6n7-s_UC. (Zitiert auf Seite 50)

[HFBPL09] J. Hebeler, M. Fisher, R. Blace, A. Perez-Lopez. *Semantic Web Programming*. Wiley Publishing, Inc., 2009. (Zitiert auf Seite 34)

[HJ04] C. Hansen, C. Johnson. *Visualization Handbook*. Elsevier Science, 2004. URL http://books.google.de/books?id=mA8ih1AieaYC. (Zitiert auf Seite 21)

[HR08] M. A. Hearst, D. Rosner. Tag Clouds: Data Analysis Tool or Social Signaller? *Hawaii International Conference on System Sciences*, 0:160, 2008. (Zitiert auf Seite 26)

[HVW09] D. Holten, J. J. Van Wijk. Force-Directed Edge Bundling for Graph Visualization. *Computer Graphics Forum*, 28(3):983–990, 2009. doi:10.1111/j.1467-8659.2009.01450.x. URL http://dx.doi.org/10.1111/j.1467-8659.2009.01450.x. (Zitiert auf Seite 30)

[Jän] H. Jänicke. Visualisierung I Vorlesungsfolien. URL http://www.iwr.uni-heidelberg.de/groups/CoVis/. (Zitiert auf Seite 25)

[jav] URL http://swingx.java.net/. (Zitiert auf den Seiten 64 und 66)

[KKEE11] K. Kim, S. Ko, N. Elmqvist, D. Ebert. WordBridge: Using Composite Tag Clouds in Node-Link Diagrams for Visualizing Content and Relations in Text Corpora. In *Proceedings of the Hawaii International Conference on System Sciences*, S. . 2011. (Zitiert auf Seite 29)

[KKEM10] D. A. Keim, J. Kohlhammer, G. Ellis, F. Mansmann. *Mastering The Information Age - Solving Problems with Visual Analytics*. Eurographics, 2010. (Zitiert auf den Seiten 7, 13, 14, 22 und 23)

[KKR10] S. Kisilevich, D. A. Keim, L. Rokach. A novel approach to mining travel sequences using collections of geotagged photos. In *Proceedings of the 13th AGILE International Conference on Geographic Information Science*. 2010. (Zitiert auf Seite 30)

[Kla02] M. Klapp. *Analyse der Datumstransformation von Kugel – und Sphäroidalfunktionen zur Darstellung des terrestrischen Schwerefeldes*. Diplomarbeit, Universität Stuttgart, 2002. (Zitiert auf den Seiten 7 und 21)

[KMK10] S. Kisilevich, F. Mansmann, D. Keim. P-DBSCAN: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos. In *Proceedings of the 1st International Conference and Exhibition*

on Computing for Geospatial Research & Application, COM.Geo '10, S. 38:1–38:4. ACM, New York, NY, USA, 2010. doi:10.1145/1823854.1823897. URL http://doi.acm.org/10.1145/1823854.1823897. (Zitiert auf Seite 28)

[KMS+08] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, H. Ziegler. Visual Analytics: Scope and Challenges. In *Visual Data Mining*. 2008. (Zitiert auf den Seiten 13 und 15)

[KS09] G. Krüger, T. Stark. *Handbuch der Java-Programmierung*. Programmer's choice. Addison-Wesley, 2009. URL http://books.google.de/books?id=ukVrWWAtgFMC. (Zitiert auf Seite 43)

[LRKC10] B. Lee, N. H. Riche, A. K. Karlson, S. Carpendale. SparkClouds: Visualizing Trends in Tag Clouds. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1182–1189, 2010. doi:10.1109/TVCG.2010.194. URL http://dx.doi.org/10.1109/TVCG.2010.194. (Zitiert auf Seite 29)

[MPS09] C. D. Manning, Prabhakar, H. Schütze. *An Introduction to Information Retrieval*. Cambridge University Press, 2009. (Zitiert auf Seite 25)

[NS10] D.-Q. Nguyen, H. Schumann. Taggram: Exploring Geo-data on Maps through a Tag Cloud-Based Visualization. In *IV'10*, S. 322–328. 2010. (Zitiert auf Seite 30)

[PD08] G. Pomberger, H. Dobler. *Algorithmen und Datenstrukturen*. it informatik. Pearson Studium, 2008. URL http://books.google.de/books?id=SVtnGzSmGuUC. (Zitiert auf Seite 50)

[pos] URL http://www.postgresql.org. (Zitiert auf Seite 50)

[PXY+05] D. Phan, L. Xiao, R. Yeh, P. Hanrahan, T. Winograd. Flow Map Layout. In *IEEE Information Visualization (InfoVis)*, S. 219–224. 2005. URL http://vis.stanford.edu/papers/flow-map-layout. (Zitiert auf Seite 30)

[rep] URL http://re-publica.de. (Zitiert auf Seite 75)

[Shn96] B. Shneiderman. The eyes have it: A task by data type taxonomie for information. In *IEEE Symposium on Visual Languages*. 1996. (Zitiert auf Seite 14)

[Sin84] R. W. Sinnott. Virtues of the Haversine. *Sky and Telescope*, 68:159, 1984. URL http://daimi.au.dk/~dam/thesis/Sky_and_Telescope_1984.pdf. (Zitiert auf Seite 20)

[Tor02] W. Torge. *Geodäsie*. De Gruyter Lehrbuch. De Gruyter, 2002. URL http://books.google.de/books?id=aL9RPdUYuWQC. (Zitiert auf Seite 20)

[twi] URL http://twitter.com/. (Zitiert auf den Seiten 15, 16 und 17)

[Ull07] C. Ullenboom. *Java ist auch eine Insel*. Galileo Computing, Bonn, 6., aktualisierte und erweiterte auflage Auflage, 2007. URL http://www.galileocomputing.de/openbook/javainsel6/. (Zitiert auf Seite 58)

[Vin75] T. Vincenty. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review*, 22(176):88–93, 1975. (Zitiert auf Seite 20)

[Wäs10] K. Wäschle. *FolkTagCloud: Eine Social-Tagging-Komponente für das SemanticMediaWiki*. Fraunhofer Verlag, 2010. (Zitiert auf Seite 26)

[Zan09] P. A. Zandbergen. Accuracy of iPhone Locations: A Comparison of Asisted GPS, WiFi and Cellular Positioning. In *Transactions in GIS*. 2009. (Zitiert auf Seite 19)

[ZZXM09] Y. Zheng, L. Zhang, X. Xie, W.-Y. Ma. Mining Interesting Locations and Travel Sequences from GPS Trajectories, 2009. (Zitiert auf Seite 19)

Alle URLs wurden zuletzt am 30.09.2012 geprüft.

Erklärung

Hiermit versichere ich, diese Arbeit selbständig verfasst und nur die angegebenen Quellen benutzt zu haben.

(Dominik Jäckle)