media/rep/

Repositorium für die Medienwissenschaft

Fabian Pfaffenberger

Twitter als Basis wissenschaftlicher Studien. Eine Bewertung gängiger Erhebungs- und Analysemethoden der Twitter-Forschung

2016

https://doi.org/10.25969/mediarep/722

Veröffentlichungsversion / published version

Buch / book

Empfohlene Zitierung / Suggested Citation:

Pfaffenberger, Fabian: *Twitter als Basis wissenschaftlicher Studien. Eine Bewertung gängiger Erhebungs- und Analysemethoden der Twitter-Forschung*. Wiesbaden: Springer Fachmedien 2016. DOI: https://doi.org/10.25969/mediarep/722.

Erstmalig hier erschienen / Initial publication here:

https://doi.org/10.1007/978-3-658-14414-2

Nutzungsbedingungen:

Dieser Text wird unter einer Creative Commons - Namensnennung - Nicht kommerziell 4.0 Lizenz zur Verfügung gestellt. Nähere Auskünfte zu dieser Lizenz finden Sie hier: https://creativecommons.org/licenses/by-nc/4.0

Terms of use:

This document is made available under a creative commons - Attribution - Non Commercial 4.0 License. For more information see: https://creativecommons.org/licenses/by-nc/4.0

DFG Deutsche Forschungsgemeinschaft

Philipps Universität Marburg

Fabian Pfaffenberger

Twitter als Basis wissenschaftlicher Studien

Eine Bewertung gängiger Erhebungs- und Analysemethoden der Twitter-Forschung

OPEN

Springer VS

Twitter als Basis wissenschaftlicher Studien

Fabian Pfaffenberger

Twitter als Basis wissenschaftlicher Studien

Eine Bewertung gängiger Erhebungs- und Analysemethoden der Twitter-Forschung

OPEN

Springer VS

Fabian Pfaffenberger
Nürnberg, Deutschland

Mit dem vorliegenden Werk hat Fabian Pfaffenberger die Ausschreibung BestMasters Medien 2015 gewonnen. Die Open-Access-Publikation wurde von der Springer Fachmedien Wiesbaden GmbH gefördert. Wir bedanken uns bei der Experten-Jury, die Prüfung und Auswahl des Manuskripts vorgenommen hat: Prof. Dr. Christoph Bläsi ist Professor für Buchwissenschaft und Book Studies an der Johannes Gutenberg-Universität Mainz. Prof. Dr. Gabriele Hooffacker ist Professorin für den Lehrbereich „Medienadäquate Inhalteaufbereitung“ an der HTWK Leipzig. Steffen Meier ist Experte für digitales Publizieren der Firma Readbox, 360°epublishing. Dr. Niels Peter Thomas ist Executive Vice President, German Language Science Publishing beim Wissenschaftsverlag Springer Nature.

ISBN 978-3-658-14413-5 ISBN 978-3-658-14414-2 (eBook)
DOI 10.1007/978-3-658-14414-2

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Springer VS
Der/die Herausgeber bzw. der/die Autor(en) 2016. Dieses Buch ist eine Open-Access-Publikation.
Open Access Dieses Buch wird unter der Creative Commons Namensnennung - Nicht kommerziell 4.0 International Lizenz (http://creativecommons.org/licenses/by-nc/4.0/deed.de) veröffentlicht, welche für nicht kommerzielle Zwecke die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.
Die in diesem Buch enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist auch für die oben aufgeführten nicht-kommerziellen Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.
Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede kommerzielle Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.
Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.
Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen.

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier

Springer VS ist Teil von Springer Nature
Die eingetragene Gesellschaft ist Springer Fachmedien Wiesbaden GmbH
Die Anschrift der Gesellschaft ist: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Inhalt

Abbildungsverzeichnis

Tabellenverzeichnis

Listingverzeichnis

Typografische Konventionen

Diese Masterarbeit verwendet einige typografische Konventionen, die das Verständnis des Inhalts unterstützen sollen.

Kursiv stehen bei erstmaliger Nennung:

- Begriffe, die definiert werden,
- Unternehmens- und Produktbezeichnungen,
- Software-Applikationen und -bibliotheken sowie
- selten gebrauchte, fremdsprachige Wörter.

In `nichtproportionaler Schrift` stehen:

- Dateinamen,
- Quelltext oder Codefragmente und
- Befehle.

Alle Auszüge aus Konfigurations-, Quelltext- und JSON-Dateien sowie Shell-Befehle/Kommandozeilen werden in markierten Listings abgebildet. Innerhalb dieser Listings ist **`<Wert>`** ein zu definierender/ersetzender Wert, `# Text` ein Zeilen-Kommentar und `[ ... ]` der Hinweis für nicht angezeigte Codefragmente.

1 Twitter in Gesellschaft und Forschung

Abbildung 1: Ursprünglicher, mittlerweile gelöschter Tweet von Justine Sacco auf Twitter. Bildquelle: Pilkington (2013).

„Going to Africa. Hope I don't get AIDS. Just kidding. I'm white!" (Sacco, 2013). Mit diesem Tweet verabschiedete sich Justine Sacco, Leiterin der Unternehmenskommunikation des New Yorker Medienunternehmens *InterActiveCorp,* am 20. Dezember 2013 vor ihrem Flug von London nach Kapstadt. Während sie sich wohl der Reichweite ihrer Äußerung nicht bewusst war, entstand binnen kürzester Zeit eine Welle der Empörung im Internet – ein sogenannter *Shitstorm*. Das Hashtag *#HasJustineLandedYet* war ein populärer Begriff auf Twitter (Ronson, 2015).

© Der/die Autor(en) 2016
F. Pfaffenberger, *Twitter als Basis wissenschaftlicher Studien*,
DOI 10.1007/978-3-658-14414-2_1

Doch nicht nur auf Twitter wurde diese Meldung diskutiert – auch etablierte Printmedien thematisierten den Tweet in ihren Online-Ausgaben, wie die *New York Times* (Southall, 2013), der *Guardian* (Pilkington, 2013), *Die Welt* (Neumann, 2013) und der *Stern* (Noffke, 2013). Nachdem der öffentliche Druck auf Sacco stieg, erhielt sie noch am selben Tag die Kündigung. InterActiveCorp entschuldigte sich außerdem in einer Stellungnahme für das Verhalten seiner Mitarbeiterin (Stelter, 2013).

Dieser Fall zeigt, welche Bedeutung Twitter in der Gesellschaft einnehmen kann. Tweets sind mittlerweile ein beliebtes Mittel zum Verbreiten von Meldungen in Echtzeit: Beispielsweise kündigte der ehemalige griechische Finanzminister Yanis Varoufakis (2015) seinen Rücktritt auf Twitter an, ebenso wie *ProSieben* (2015) das Karriereende seines Entertainers Stefan Raab. Julia Klöckner informierte bei der Wahl zum Bundespräsidenten 2009 noch vor der offiziellen Bekanntmachung die Öffentlichkeit über den Wahlausgang (Boie, 2011). Der bisher weitverbreitetste Tweet wurde während der Oscar-Verleihung 2014 von der Moderatorin Ellen DeGeneres geschrieben und über 3,2 Millionen Mal[1] geteilt (DeGeneres, 2014). Er zeigte ein Gruppenfoto mehrerer Hollywood-Stars und wurde live – als Teil der Show – aufgenommen.

Twitter dient jedoch häufig nicht nur als Kanal zur einseitigen Bekanntgabe von Meldungen, sondern vor allem als Netzwerk zur *gegenseitigen* Information und Interaktion. Besonders im Umfeld der politischen Umwälzungen in Nordafrika und dem Nahen Osten spielte Twitter eine wichtige Rolle bei der Weitergabe von Informationen und der Koordination von Protesten (Bruns, Highfield, & Burgess, 2013; Lotan et al., 2011). Ägypten und die Türkei sperren beispielsweise gelegentlich einzelne Nutzer oder das ganze Portal – etwa bei politischen Ausschreitungen, kritischer Berichterstattung und neuerdings auch nach Anschlägen (Gadde, 2014; Kazim, 2015). Diese Maßnahmen sollen – aus der Sichtweise der jeweiligen politischen Regime – die Koordination von Protestbewegungen und die Weitergabe sensibler oder kritischer Informationen unterbinden.

Ähnlich reagierten Brüsseler Behörden: Während Razzien im Zuge der Pariser Terror-Anschläge im Dezember 2015 bat die Brüsseler Polizei die Twitter-User um eine selbst auferlegte Funkstille, damit potentielle Zielpersonen nicht gewarnt werden können (Police Fédérale, 2015). Viele Nutzer folgten der Bitte und veröffentlichten stattdessen Katzenbilder, um kompromittierende Tweets in der Masse untergehen zu lassen (Rogers, 2015). Bei den Terroranschlägen in Paris im November 2015 kamen viele Meldungen, Bilder und Videos zunächst nicht von der

[1] Stand: Februar 2016

Presse, sondern von Privatpersonen in sozialen Medien (Wendling, 2015). Diese Beispiele verdeutlichen die Präsenz und Bedeutung, die Twitter vor allem bei Katastrophen und Anschlägen, aber auch bei sportlichen und kulturellen Großereignissen haben kann.

Soziale Medien im Internet sind in den letzten Jahren rasant gewachsen. Die weite Verbreitung des Kurznachrichtendienstes Twitter mit seinen über 302 Millionen monatlich aktiven Nutzer/-innen und etwa 500 Millionen gesendeten Tweets pro Tag (Twitter, Inc., 2015k) steht nur beispielhaft für die Bedeutung moderner, internetbasierter Kommunikationskanäle. Statistiken zeigen, dass die knapp 56 Millionen deutschen Internetnutzer durchschnittlich 166 Minuten am Tag im Internet sind – etwa 39 Prozent nutzen dabei Online-Communities (ARD-Werbung Sales & Services GmbH, 2015). In diesem Zusammenhang sehen auch Politik, Medien und Unternehmen einen großen Nutzen in sozialen Medien: Sie eignen sich zur schnellen, einfachen, kostengünstigen und zeitunabhängigen Kommunikation mit den Wählern, Lesern oder Kunden.

Twitter bietet ein ausführliches, klar strukturiertes und frei zugängliches Datenset, welches sich sowohl für detaillierte, als auch für breit angelegte Datenanalysen gut eignet. So umfasst ein Datensatz rund 150 Metadaten − neben Tweet und Benutzerkonto auch Standortdaten, gewählte Sprache, Zeitzone oder die Vernetzung von Nutzer/-innen. In Abhängigkeit von der Tweetfrequenz ergibt sich ein ständig wachsender Datenpool. Aufgrund der standardisierten Struktur und der hohen Verfügbarkeit der Daten wird Twitter mittlerweile häufig als Datengrundlage für die wissenschaftliche Forschung in den unterschiedlichsten Disziplinen verwendet. Zwischen Januar 2007 und August 2015 finden sich auf *Scopus* allein 737 Forschungsarbeiten der Sozialwissenschaften, deren Titel den Begriff *Twitter* beinhalten, wobei die Anzahl an Artikeln in einem Jahr kontinuierlich ansteigt[2]. Daneben befassen sich auch andere Forschungszweige, wie die Medizin, Informatik, Ingenieurswissenschaften oder Psychologie mit Twitter (siehe auch Kapitel 2).

Je nach wissenschaftlicher Disziplin, Themensetzung, Datengrundlage, Rechenleistung und Budget ergeben sich dabei viele unterschiedliche Ansätze für die wissenschaftliche Verwendung von Twitter-Daten. Die Erhebung kann durch direktes Abgreifen von Tweets in Echtzeit, die Nutzung kostenfreier oder kostenpflichtiger Online-Aggregatoren oder durch den Erwerb vollständiger Datenbanken bei Datenhändlern vollzogen werden. Bei der Analyse besteht die Wahl zwischen der Nutzung (kostenpflichtiger) Online-Dienste und der Eigenauswertung mit Hilfe von Verarbeitungsprogrammen. Letztere können wiederum Lösungen

[2] Eigene Abfrage auf scopus.com. Stand: August 2015. Suchterm: TITLE (twitter) AND (LIMIT-TO (SUBJAREA , "SOCI"))

out of the box oder Eigenentwicklung auf Basis freier Skriptsprachen sein, wie beispielsweise *Python*. Python ist eine frei verfügbare, umfangreiche und dennoch übersichtliche und leicht anwendbare Programmiersprache mit vielen Erweiterungsmöglichkeiten (s. Kapitel 3.2).

Trotz der Vielzahl an Studien und respektive an Ansätzen fehlt eine genauere Betrachtung und vor allem Bewertung typischer wissenschaftlicher Vorgehensweisen. Ziel dieser Arbeit ist deshalb eine Darstellung mehrerer methodischer Ansätze zur Erhebung, Speicherung und Analyse von Twitter-Daten auf Basis von Python. In der Arbeit sollen die verschiedenen Erhebungs-, Speicher- und Auswertungsverfahren näher betrachtet und anhand von Praxisbeispielen hinsichtlich ihrer Leistungsfähigkeit und ihres Nutzens für die Forschung bewertet werden. Darauf aufbauend wird die Eignung von Twitter als Quelle und Forschungsobjekt wissenschaftlicher Analysen erörtert, indem auch Einschränkungen und Herausforderungen aufgezeigt werden. Dabei liegt der Fokus nicht auf Vollständigkeit, sondern auf der Darstellung der Praktikabilität von Methoden anhand ausgewählter freier, kostenloser Programme.

Die Skizzierung einiger wissenschaftlicher Arbeiten unterschiedlicher Forschungsbereiche in Kapitel 2 soll nicht nur einen Überblick möglicher Verwendungszwecke für Twitter-Daten geben, sondern auch den Bedarf einer vergleichenden Darstellung möglicher Verfahren hervorheben. Kapitel 3 dient der Vermittlung von Grundlagen, indem Twitter, die spezielle Kommunikation auf diesem Portal und die Datenstruktur angesprochen werden. Zudem erfolgt eine kurze Einführung in die Programmiersprache Python. Kapitel 4 befasst sich schließlich mit der methodischen Abhandlung. Zuerst werden drei Ansätze zum Sammeln von Tweets vorgestellt: Abfragen über die beiden Programmschnittstellen[3] (APIs) *Streaming API* und die *REST APIs* sowie die Beschaffung von Twitter-Daten über Drittanbieter. Diese drei Möglichkeiten werden schließlich in Kapitel 4.1.4 gegenübergestellt und hinsichtlich ihrer unterschiedlichen Eignung für die Datensammlung bewertet.

Anschließend vergleicht Kapitel 4.2 zwei gegensätzliche Konzepte zum Speichern von Tweets: Das Speichern in Einzeldateien und das Einspeisen in Datenbanken, wobei ein Hauptaugenmerk auf dem Datenbanksystem *MongoDB* liegt. Daran anknüpfend erfolgt eine Betrachtung verschiedener Ansätze zum Auswerten der gesammelten Tweets (Kapitel 4.3). Zunächst thematisiert Kapitel 4.3.1

[3] Schnittstellen (Application Programming Interfaces, APIs) dienen beispielsweise der Kommunikation zwischen Datenbank und Nutzer. Über sie werden An- und Abfragen oder – allgemein gesagt – Befehle übermittelt und verwaltet. Die Kommunikation zwischen Schnittstelle und Endpunkt erfolgt dabei immer auf Quellcode-Ebene.

sinnvolle Vorverarbeitungsschritte zur Verbesserung der Datenqualität und Optimierung der Datenstruktur für eine automatisierte Analyse. Danach werden grundlegende, bereits in MongoDB integrierte Verfahren präsentiert und verglichen. Darüber hinaus gibt Kapitel 4.3.3 einen Einblick in die computergestützte Textanalyse, stellt grundlegende computerlinguistische Untersuchungen vor und schließt mit der Durchführung einer einfachen semantischen Analyse ab.

Schließlich beleuchtet Kapitel 5 die Eignung von Twitter als Quelle wissenschaftlicher Arbeiten, indem beispielsweise die Qualität und Zweckmäßigkeit der zur Verfügung gestellten Daten kritisch hinterfragt wird. Darüber hinaus sind auch rechtliche Bestimmungen zum Datenschutz und das Fehlen anerkannter Metriken von Bedeutung.

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung - Nicht kommerziell 4.0 International Lizenz (http://creativecommons.org/licenses/by-nc/4.0/deed.de) veröffentlicht, welche für nicht kommerzielle Zwecke die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.
Etwaige Abbildungen oder sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende oder der Quellreferenz nichts anderes ergibt. Sofern solches Drittmaterial nicht unter der genannten Creative Commons Lizenz steht, ist eine Vervielfältigung, Bearbeitung oder öffentliche Wiedergabe nur mit vorheriger Zustimmung des betreffenden Rechteinhabers oder auf der Grundlage einschlägiger gesetzlicher Erlaubnisvorschriften zulässig.

2 Forschungsstand

Zahlreiche Studien unterschiedlicher wissenschaftlicher Disziplinen, wie der Kommunikationsforschung, Soziologie, Geografie oder Politikwissenschaften, beschäftigten sich bereits mit *Twitter*. In den letzten Jahren stieg vor allem das Interesse an der Auswertung geospezifischer Daten, die Rückschlüsse auf Standorte und Bewegungsmuster liefern können (z.B. Gerätestandort, Zeitzone, Sprache). Beispielsweise nutzten Hawelka et al. (2014) die in Tweets gespeicherten Standortdaten zur Erstellung von Bewegungsmustern, indem sie die über einen längeren Zeitraum gesammelten Geo-Daten (GPS-Daten und IP-Adressen) von Nutzerprofilen verknüpften. Jedoch erkannten Graham, Hale und Gaffney (2014) bei einem Vergleich von benutzerdefinierten Profil-Standorten mit GPS-basierten Gerätestandorten und Tweet-spezifischen Zeitzonen-Angaben, dass diese Informationen häufig voneinander abweichen. Nutzer/-innen geben in ihrem Profil oft falsche Wohn- beziehungsweise Aufenthaltsorte an.

Cheng, Caverlee und Lee (2010) versuchten eine zuverlässigere Lokalisierung von Tweets, die unabhängig etwaiger Standortangaben durch die Nutzer ist. Die geolinguistische Analyse von Tweets erfolgte hier mithilfe vorhandener Algorithmen *(Google Geocoding API*, *Yahoo PlaceFinder*), um über den Tweet-Inhalt den momentanen Aufenthaltsort abzuleiten. Letztlich erwies sich jedoch eine rein technische Auswertung im Vergleich zu einer menschlichen, manuellen Codierung als nur bedingt verlässlich.

Carter, Tsagkias und Weerkamp (2011), Gottron und Lipka (2010) sowie Bifet und Frank (2010) verweisen hierbei auf elementare Unterschiede zum normalem Fließtext, auf den jedoch viele Linguistik-Programme ausgelegt sind: Twitter-Meldungen sind auf 140 Zeichen begrenzt, enthalten häufig Abkürzungen, Neologismen, mehrsprachige Inhalte und selten eine klare Satzstruktur, wie dies bei normalen Fließtexten der Fall ist. Dementsprechend können Inhalte leicht fehlgedeutet werden, weshalb eine programmgestützte Inhaltsanalyse immer mit Vorsicht genutzt werden sollte.

© Der/die Autor(en) 2016
F. Pfaffenberger, *Twitter als Basis wissenschaftlicher Studien*,
DOI 10.1007/978-3-658-14414-2_2

Das Problem der eingeschränkten Analysierbarkeit von Tweets durch Programme betrifft auch andere Forschungszweige, die auf eine automatisierte inhaltliche Auswertung angewiesen sind, wie die Sentiment-Forschung. Mithilfe mehrstufiger Filter- und Verarbeitungsprozesse (Bollen, Pepe, & Mao, 2011; Pak & Paroubek, 2010), Einbezug zusätzlicher Informationen wie Nutzerdaten (Sriram, Fuhry, Demir, Ferhatosmanoglu, & Demirbas, 2010) oder verwandter/ähnlicher Tweets (Jiang, Yu, Zhou, Liu, & Zhao, 2007) sowie der Verwendung lernfähiger Analyseprogramme (Bifet & Frank, 2010; Carter et al., 2011; Tumasjan, Sprenger, Sandner, & Welpe, 2010) können die oben genannten Probleme jedoch einigermaßen kontrolliert werden. Allerdings sollte eine korrekte Einordnung der Stimmung auch den Kontext des Tweets berücksichtigen, was in bisherigen Studien nicht gemacht wurde. Dies könnte an der Datenstruktur der Tweets liegen: Tweets werden einzeln nach Veröffentlichungszeitpunkt und nicht als zusammenhängende Konversationen übermittelt. Folglich müssten Unterhaltungen erst manuell zusammengefügt werden.

Studien der Sentiment-Forschung untersuchten beispielsweise den Zusammenhang der aggregierten Stimmung beobachteter Nutzer (= Sentiment) mit Kursschwankungen, Ölpreisen und medialer Großereignissen (Bollen, Pepe et al., 2011), die Möglichkeit, mithilfe des aktuellen Stimmungsbildes auf Twitter Aktienkurse vorherzusagen (Bollen, Mao, & Zeng, 2011; Zhang, Fuehres, & Gloor, 2011).

Echtzeit-Daten aus Twitter dienten zuletzt auch als Grundlage zur Erkennung von Epidemien, wie Influenza (Signorini, Segre, & Polgreen, 2011), beziehungsweise des allgemeinen Gesundheitszustandes der Bevölkerung (Paul & Dredze, 2011; Scanfeld, Scanfeld, & Larson, 2010). Aramaki, Maskawa und Morita (2011) zeigten, dass mithilfe automatisierter Erhebung und Analyseverfahren, wie dem Natural Language Processing, Vorhersagen über das Auftreten und den Verlauf von Grippe-Wellen machen lassen. Des Weiteren wurden Tweets zur Früherkennung von Erdbeben (Earle, Bowden, & Guy, 2012; Sakai, Okazaki, & Matsuo, 2010), oder zur Analyse der Kommunikation während Krisen verarbeitet (Acar & Muraki, 2011; Heverin & Zach, 2010; Mendoza, Poblete, & Castillo, 2010; Vieweg, Hughes, Starbird, & Palen, 2010). Jedoch besteht auch hier das Problem, dass Tweets aufgrund der unkonventionellen Sprache nur schwer automatisiert analysiert werden können. Ähnlich, wie bei der Sentiment-Erkennung, gehen durch den fehlenden Miteinbezug des Twitter-Kontextes womöglich viele relevante Informationen verloren.

Ein weiterer Schwerpunkt der Forschung liegt in der Analyse der politischen Kommunikation auf beziehungsweise über Twitter. Dabei wurden sowohl Tweets

über Politiker/-innen als auch deren Nachrichten und Interaktion mit anderen Nutzern auf Twitter betrachtet. So dienten Echtzeitdaten für eine Bewertung von Politikern während TV-Debatten (Diakopoulos & Shamma, 2010) oder zur Analyse der Stimmung während der US-Präsidentschaftswahl 2012 (Wang, Can, Kazemzadeh, Bar, & Narayanan, 2012).

Umstritten ist die Möglichkeit, mit Hilfe von Twitter-Daten den Ausgang von Wahlen zu prognostizieren. Es gibt einige Kontroversen hinsichtlich der Aussagekraft von Tweets und der Zuverlässigkeit der Schätzung. Tumasjan et al. (2010) sowie Sang und Bos (2012) sehen in Twitter trotz der eingeschränkten Repräsentativität der Daten ein relativ zuverlässiges Instrument zur Wahlprognose. Jungherr, Jürgens und Schoen (2012) zeigen jedoch, dass sowohl das Datenmaterial, als auch Erhebungszeitpunkt und Auswahl der Parteien keine valide Erhebungsmethode darstellen und somit keine verlässlichen Rückschlüsse auf Wahlergebnisse erlauben. Aufgrund der unterschiedlichen Wähler-Zielgruppen gibt es eine Verzerrung hinsichtlich Twitter-Nutzungsverhalten und somit der Tweet-Häufigkeit der jeweiligen Partei-Anhängerschaft. Beispielsweise liegt es nahe, dass Unterstützer der Piraten-Partei deutlich häufiger twittern als Anhänger der großen Volksparteien.

Conover et al. (2011) beobachteten während der US-amerikanischen *Midterm Elections* 2010 eine hohe Polarisierung der politisch aktiven Twitter-Nutzer zwischen linkem und rechten Lager und einer geringen Interaktion zwischen diesen Gruppen. Weitere Studien ergaben, dass der Grad politischer Inhalte auf Twitter stark von medialen Ereignissen, wie Diskussionsrunden oder Wahlveranstaltungen, abhängt (Dusch et al., 2015; Larsson & Moe, 2012). Auch ist der Interaktionsgrad zwischen Politikern und „normalen" Usern eher gering: Politiker verwenden Twitter meist eher als Werbeplattform für politische Veranstaltungen (Dusch et al., 2015; Thimm, Einspänner, & Dang-Anh, 2012) beziehungsweise zur Verbreitung ihrer Standpunkte, als zur direkten Interaktion mit ihren Kontakten (Elter, 2013; Grant, Moon, & Busby Grant, 2010; Parmelee & Bichard, 2012).

Eine deutlich aktivere politische Partizipation und Kommunikation auf Twitter findet dagegen während politischer Proteste und Aufständen statt: Besonders bei der Ägyptischen Revolution sowie der *Grünen Revolution* im Iran spielte Twitter eine zentrale Rolle bei der Organisation und Informationsweitergabe (Bruns et al., 2013). Bei Ereignissen, in denen klassische Massenmedien aufgrund der Rasanz der Entwicklungen oder Abwesenheit von Journalisten nicht zeitnah reagieren konnten, etablierte sich Twitter als wichtige Plattform für die Produktion und Distribution von Nachrichten (Papacharissi & de Fatima Oliveira, 2012). Der Kurznachrichtendienst diente, aus Mangel an zuverlässigen staatlichen Medien und

aufgrund der Unterdrückung freier, kritischer Meinungsäußerungen, hierbei auch als Nachrichtenquelle für westliche Medien (Khondker, 2011; Lotan et al., 2011). Schließlich befassten sich auch zahlreiche Arbeiten mit der Kommunikation auf Twitter an sich: Chen (2011) begründete die Art und Stärke des Nutzungsverhalten von Twitter-Usern mit dem *Uses and Gratification* Ansatz. Je länger die Nutzung (hinsichtlich des Zeitraums), desto belohnender wird eine Vernetzung mit anderen Nutzern wahrgenommen, wobei die Zahl eigener Tweets und Replies die Stärke des Effekts beeinflusst. Liu, Cheung und Lee (2010) sehen vier Dimensionen innerhalb des Ansatzes, die belohnend auf die Nutzung von Twitter wirken: *Content* (Möglichkeit zur Informationsaufnahme- und Verbreitung), *Technology* (bequeme und unmittelbare Kommunikation), *Social* (Soziale Interaktion und Vernetzung) sowie *Process* (Unterhaltung, Zeitvertreib), wobei die letzten beiden eine geringere Bedeutung haben. Dies wird damit erklärt, dass Twitter ursprünglich nur zum Austausch von Informationen konzipiert war und soziale Interaktionsmöglichkeiten erst später implementiert wurden. Auch haben sich Kommunikationsmöglichkeiten erst mithilfe von Konventionen, wie Retweets oder direkte User-Verweise in Tweets (*@username*) durchgesetzt (boyd, Golder, & Lotan, 2010; Honeycutt & Herring, 2009).

Cha, Haddadi, Benevenuto und Gummadi (2010) betrachteten die soziale und informationelle Komponente hinsichtlich des Einflusses populärer Twitterer. Die Wahrscheinlichkeit, dass ein Tweet eine hohe Aufmerksamkeit erhält ist demnach weniger von der Vernetzung eines Users, im Sinne von Followern, sondern vom Inhalt der Nachricht abhängig. Diese These, dass Twitter eher zum Informationsaustausch verwendet wird, als zum Knüpfen sozialer Kontakte, unterstützen auch Huberman, Romero und Wu (2008), Java, Song, Finin und Tseng (2007) sowie Johnson und Yang (2009).

So unterschiedlich die Forschungsabsichten und Verwendungszwecke bezüglich Twitter sind, so verschieden sind auch die Methoden zur Datengewinnung und Auswertung. Die Erfassung von Twitter-Daten lässt sich dabei in drei Komplexe zusammenfassen: Abfragen historischer Daten über die Programmschnittstelle *Search API*, Erhebungen von gesampelten Echtzeitdaten über die *Streaming API* und die Verwendung von Programmen und Datensätzen Dritter (siehe Kapitel 4.1). Die hier erwähnten Studien gingen wie folgt vor: boyd, Golder und Lotan (2010), Cheng et al. (2010), Diakopoulos und Shamma (2010), Grant et al. (2010), Sakai et al. (2010) und Vieweg et al. (2010) nutzen die frei zugängige Search API von Twitter, um über Suchabfragen historische Daten eines eingeschränkten Zeitraums zu erhalten. Bifet und Frank (2010), Graham et al. (2014), Hawelka et al. (2014), Sang und Bos (2012) sowie Signorini et al. (2011) griffen dagegen mit Hilfe der

Streaming API (gesampelte) Daten in Echtzeit ab. Neben diesen beiden populären, da kostenlosen, Methoden der Datenerhebung auf Twitter, gibt es noch eine Vielzahl von Drittanbietern, die Twitter-Daten gebührenpflichtig oder gratis zur Verfügung stellen. Conover et al. (2011) erhielten Zugriff auf die sogenannte *Gardenhose*[4] wogegen Wang et al. (2012) Daten vom Dienstleister *Gnip* kauften und Dusch et al. (2015) sowie Larsson und Moe (2012) Online-Dienste zur Datenerfassung nutzen.

Hinsichtlich der Auswertung von Twitter-Daten ergibt sich ein ähnlich differenziertes Bild: Twitter-bezogene Studien fokussieren sich nicht nur auf reine Inhaltsanalysen, sondern beziehen sich (zusätzlich) auch auf Befragungen oder Experimente. Dennoch ist die Inhaltsanalyse nach einer Meta-Studie von Williams, Terras und Warwick (2013) eine dominierende Methode, was sich auch auf das umfangreiche Datenangebot durch Twitter zurückführen lässt. Die Nutzung der bereits vorhandenen, leicht zugänglichen, stark strukturierten und ausführlichen Daten ist einfacher und schneller als die Durchführung von Befragungen oder Experimenten. Interviews werden häufig nur ergänzend durchgeführt, etwa, um Ergebnisse der Inhaltsanalyse durch ermittelte Einstellungen und Verhalten der Nutzer zu erklären.

Dennoch fehlen methodische Standards, da die Twitter-Forschung noch sehr jung ist (Bruns & Liang, 2012). Ein weiterer Forschungsschwerpunkt liegt deshalb in der Konzeption neuer Methoden und Algorithmen zur Analyse der Daten (Williams et al., 2013). Einige Forschende entwickeln für ihre Forschungszwecke eigene Ansätze beziehungsweise Programme zur Twitter-Analyse. Das eigentliche methodische Vorgehen, insbesondere die Datengewinnung, wird dabei selten detailliert präsentiert (Weller, 2014). Trotz der hier dargestellten großen Bandbreite an Ansätzen und Verfahren der Twitter-Analyse, gibt es kaum wissenschaftliche Arbeiten, die sich mit den Methoden der Datengewinnung und Auswertung befassen. Wenn überhaupt, wurden nur einzelne Vorgehensweisen angesprochen.

So zeigen Perera, Anand, Subbalakshmi und Chandramouli (2010), wie mit der Programmiersprache *Python* Twitter-Daten gesammelt und in einem *MySQL*-Datensystem verarbeitet werden können. Tugores und Colet (2013) vergleichen für eine Mobilitätsanalyse zwei Varianten von Datenbanksystemen (*SQL* und *noSQL*) im Kontext einer Twitter-Analyse mit Python. Bruns und Liang (2012) präsentierten mehrere Programme zum Erfassen und Analysieren von Tweets

[4] Die Daten der öffentlichen Streaming API und REST APIs sind hinsichtlich Datenvolumen und Abfragehäufigkeit limitiert. Innerhalb der Streaming API bietet die Gardenhose einen größeren Datenumfang als der allgemeine Datenzugang Spritzer, wogegen die Firehose einen Echtzeit-Zugriff auf alle Daten ermöglicht (siehe Kapitel 4.1 für eine ausführliche Erläuterung).

während Naturkatastrophen. Dennoch findet sich nirgends eine detaillierte Bewertung der Ansätze. Aufgrund der unterschiedlichen Disziplinen und somit auch Forschungsschwerpunkte fehlte auch die spezifische Bewertung dieser Möglichkeiten im Hinblick auf die Analyse der Twitter-Kommunikation.

Kumar, Morstatter und Liu (2014) liefern bisher die umfassendste Übersicht, mit welchen Ansätzen Twitter-Daten gesammelt und verarbeitet werden können. Jedoch werden auch hier nur ausgewählte Aspekte der Datensammlung und -analyse betrachtet und die einzelnen Methoden weder miteinander verglichen, noch hinsichtlich ihrer Praktikabilität bei wissenschaftlichen Erhebungen bewertet. Eine ähnliche Zielsetzung verfolgt Russell (2013): Anhand zahlreicher fallspezifischer Beispiele erhält der Leser einen guten Überblick über die Möglichkeiten der (nicht nur) Twitter-bezogenen Datenerhebung und Auswertung mittels Python, MongoDB und NLTK. Jedoch ist auch hier eine vergleichende und wertende Betrachtung – besonders im Hinblick auf die Nützlichkeit für die Forschung – nicht vorhanden.

Es gibt folglich bereits eine Vielzahl an Studien, die sich mit der Kommunikation auf Twitter beziehungsweise der wissenschaftlichen Auswertung der generierten Nutzer-Daten befasst haben. Was fehlt, ist ein vergleichender Überblick über Verfahren der Twitter-Analyse für die Sozialwissenschaften. Williams et al. (2013) befanden in ihrer Meta-Analyse, dass sich etwa 80% der analysierten Beiträge auf den Inhalt der Tweets sowie die Nutzer und deren Kommunikationsweise konzentrierten. Dabei wurde eine Vielzahl unterschiedlichster Methoden zur Erfassung und Analyse von Twitter-Daten angewendet, oftmals sogar mehrere Ansätze in einer Arbeit. Demgegenüber war die rein technische Betrachtung von Twitter am stärksten unterrepräsentiert. Die Autoren verwiesen hier nicht nur auf eine geringere Beimessung an Bedeutung, sondern auch auf mögliche technische Barrieren und Verständnisprobleme (Williams et al., 2013, S. 402).

Aktuelle Verfahren zur Messung der Nutzung von Twitter (und anderen sozialen Medien) sind weder standardisiert, noch unabhängig bestätigt, sondern funktionieren eher als eine Art „Black Box“ (Weller, Bruns, Burgess, Mahrt, & Puschmann, 2014, S. xxxii), deren Ergebnisse Forschende vertrauen müssen. Deshalb sollen nach einer theoretischen Einführung in den Dienst Twitter und die hier verwendete Programmiersprache Python einige gängige Ansätze genauer betrachtet und anhand von Fallbeispielen hinsichtlich Praktikabilität und Anwendungsweise verglichen werden.

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung - Nicht kommerziell 4.0 International Lizenz (http://creativecommons.org/licenses/by-nc/4.0/deed.de) veröffentlicht, welche für nicht kommerzielle Zwecke die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.
Etwaige Abbildungen oder sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende oder der Quellreferenz nichts anderes ergibt. Sofern solches Drittmaterial nicht unter der genannten Creative Commons Lizenz steht, ist eine Vervielfältigung, Bearbeitung oder öffentliche Wiedergabe nur mit vorheriger Zustimmung des betreffenden Rechteinhabers oder auf der Grundlage einschlägiger gesetzlicher Erlaubnisvorschriften zulässig.

3 Grundlagen

Dieses Kapitel dient der Vermittlung technischer Grundlagen und soll dem Leser einen Einblick in den Mikroblogging-Dienst Twitter. Dafür wird zuerst Twitter vorgestellt (Kapitel 3.1), indem auch die Kommunikation auf Twitter charakterisiert sowie Konventionen der Interaktion und allgemeine Begrifflichkeiten erläutert werden. Anschließend folgen ein Überblick der Datenstruktur und eine Skizzierung der darin enthaltenen wesentlichen Informationen. Da die vorliegende Arbeit die Programmiersprache Python zur Datensammlung und -analyse verwendet, stellt Kapitel 3.2 diese kurz vor und erläutert deren Vorteile hinsichtlich Anwendung und Verständlichkeit.

3.1 Post, Reply, Retweet – der Internet-Dienst Twitter

Twitter ist in erster Linie ein Echtzeit-Internetdienst zum Teilen von auf 140 Zeichen limitierten Text-Nachrichten (*Tweets*) in einem personalisierten, öffentlichen Nachrichtenstrom (Jürgens & Jungherr, 2011, S. 203). Dieser Nachrichtenfeed kann von anderen Twitterern abonniert werden, um dadurch jeder neuen Nachricht eines Nutzers automatisch zu folgen. Der abonnierende Nutzer wird als *Follower* bezeichnet und ist als dieser öffentlich gekennzeichnet (siehe Kapitel 3.1.2). Die Stärke des Dienstes liegt in der schnellen und ungefilterten Verbreitung von Informationen (Parmelee & Bichard, 2012, S. 216). Durch die Begrenzung auf 140 Zeichen muss der Nachrichteninhalt auf das Wesentliche konzentriert werden, die geringe Länge fördert auch eine gute und schnelle Lesbarkeit. Während in der Frühphase der Entwicklung der reine Informationsaustausch im Fokus stand, folgten in mehreren Entwicklungsschritten weitere Funktionen zur sozialen Interaktion. So ermöglicht die Plattform mittlerweile auch das Weiterleiten von Tweets (*Retweet*), eine explizite Nennung und Verknüpfung anderer Nutzer/-innen in Nachrichten (*Mention*), das Teilen von Fotos, Links und Videos sowie das Schreiben privater Nachrichten (*Direct Message*) zu einzelnen Personen oder Gruppen (Stone, 2009; Weil, 2014).

© Der/die Autor(en) 2016
F. Pfaffenberger, *Twitter als Basis wissenschaftlicher Studien*,
DOI 10.1007/978-3-658-14414-2_3

Twitter ist mittlerweile ein weit verbreiteter Kommunikationskanal mit einer Fülle von Anwendungsmöglichkeiten. Beispielweise nutzt die Politik Twitter zur Interaktion mit (potentiellen) Wählern, Journalist/-innen zur Verbreitung von Informationen (wie Eilmeldungen), Fernsehanstalten als weiteren Kommunikationskanal während TV-Sendungen (für Kommentare und Feedback) oder Unternehmen als Werbekanal mit Hinweisen zu Aktionen oder Produkten (Bruns & Stieglitz, 2012; Grant et al., 2010; Jansen, Zhang, Sobel, & Chowdury, 2009; Jungherr, 2015; Lasorsa, Lewis, & Holton, 2012; Mamic & Almaraz, 2014). Hinzu kommt eine vielfältige Anwendung als Kommunikationskanal zwischen sich gegenseitig bekannten oder unbekannten Personen – von der „normalen" Nutzung im Alltag bis zur Interaktion während politischer Krisen, wie der des sogenannten *Arabischen Frühlings* (boyd et al., 2010b; Christensen, 2011; Lotan et al., 2011).

Laut Twitter (2015k) gab es im März 2015 etwa 302 Millionen monatlich aktive Nutzer, was im Vergleich zu März 2010 mit 30 Millionen (Twitter, Inc., 2015b) eine Verzehnfachung bedeutet. Twitter definiert aktive Nutzer/-innen als Personen, die pro Monat mindestens einmal auf der Plattform aktiv waren (z.B. durch Anmelden im Account). Von den aktiven Usern nutzen etwa 80 Prozent den Dienst über das mobile Internet, insgesamt werden pro Tag 500 Millionen Tweets verfasst (Twitter, Inc., 2015k). Wissenschaftler, die Twitter-Daten beziehen, steht somit ein sehr großes potentielles Datenset zur Verfügung.

Zur Betrachtung der Staaten mit den meisten Twitter-Nutzern sollten gewichtete Daten verwendet werden, um Verzerrungen durch die Einwohnerzahl zu vermeiden. Da Twitter keine offiziellen Zahlen zur Herkunft seiner Nutzer/-innen veröffentlicht, führten Mocanu et al. (2013) eine Lokalisierung anhand von Sprache und Standort durch. Nach Anzahl der Accounts je 1000 Einwohner eines Staates ergab sich folgendes Bild: Kuwait (1 Prozent), die Niederlande (0,39 Prozent), Brunei (0,31), Großbritannien (0,3) und die USA (0,25) belegten Platz eins bis fünf. Deutschland wies in etwa einen Anteil von 0,04 Prozent an Twitter-Nutzern auf (Ebda). In absoluten Zahlen weisen die USA zwar den größten Anteil an Nutzern auf – dies bestätigen unter anderem Analysen des Twitter-Volumens (SimilarWeb, 2014) – relativ zur Einwohnerzahl belegt US-Amerika jedoch nur Platz fünf.

Allerdings sind diese Angaben alle nur eingeschränkt zuverlässig: Der Tweet-Standort erlaubt noch keinen verlässlichen Rückschluss auf die Nationalität des Nutzers. So könnten unter anderem Verzerrungen durch Reisen in andere Länder auftreten. Bei der Erhebung durch Mocanu et al. (2013) wurden zumindest Hauptreisezeiten berücksichtigt. Die Problematik der Lokalisierung von Tweets ist folglich auch hier präsent.

Interessant ist auch die Altersstruktur der monatlich aktiven Nutzer: Während bei Facebook die Altersgruppe der 25- bis 34-Jährigen mit 29 Prozent dominiert und knapp 24 Prozent der Nutzer älter als 45 sind (GlobalWebIndex, 2015), ist die Altersverteilung der männlichen und weiblichen Twitter-Nutzer gleichmäßiger (siehe Abbildung 2:). Zwar haben auch hier die 25- bis 34-Jährigen mit 22 Prozent den größten Anteil, jedoch sind die Abstände zu anderen Altersgruppen deutlich geringer. Nutzer ab 45 Jahren haben sogar einen Anteil von 38 Prozent, wovon die knappe Mehrheit über 55 Jahre alt ist (comScore, 2015).

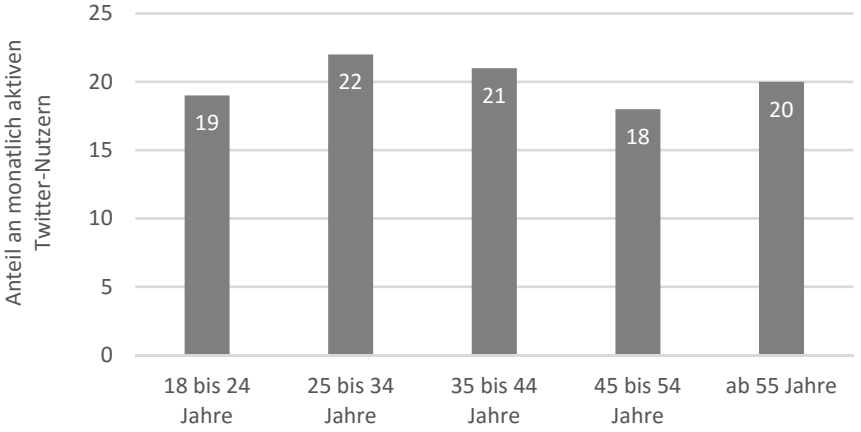

Abbildung 2: Altersverteilung aktiver Twitter-Nutzer im Dezember 2014. Quelle: comScore (2015), eigene Darstellung.

Der Online-Dienst Twitter wird, je nach Perspektive und Nutzung, mal als soziales Netzwerk, mal als reiner Kurznachrichtendienst bezeichnet. Diese Diskussion um die Definition von Twitter soll zunächst in Kapitel 3.1.1 aufgegriffen werden. Dabei wird auch auf aktuelle Statistiken über Nutzer und Nutzung eingegangen. Anschließend folgt eine genauere Betrachtung der Twitter-Nutzung und der Datenstruktur von Tweets.

3.1.1 Einordnung in die Social Media Landschaft

Aufgrund der mittlerweile umfassenden sozialen Kommunikationsmöglichkeiten ist eine klare Einordnung des Dienstes innerhalb der Social Media Landschaft nicht mehr möglich (Parmelee & Bichard, 2012, S. 38). Einerseits teilt Twitter viele Eigenschaften sozialer Netzwerke (wie *Facebook* oder *LinkedIn*): halb-öffentliche Profile, Interaktivität, einen sozialen Charakter der Interaktion, Vernetzung mit Nutzerlisten (boyd & Ellison, 2007). Andererseits wird Twitter auch als Mikroblogging-Plattform gesehen (Ebersbach, Glaser & Heigl, 2008) und ist mit seinen Funktionen und Eigenschaften immer noch näher an Blogs als an sozialen Netzwerken. Ross, Terras, Warwick und Welsh (2011, S. 217) definieren Mikroblogging wie folgt:

> „Microblogging is a variant of blogging, which allows users to quickly post short updates, providing an innovative communication method that can be seen as a hybrid of blogging, instant messaging, social networking and status notifications. The word's origin suggests that it shares the majority of elements with blogging, therefore it can potentially be described using blogging's three key concepts (Karger and Quan, 2004): the contents are short postings, these postings are kept together by a common content author who controls publication, and individual blog entries can be easily aggregated together."

Im Vergleich zu anderen Netzwerken wie *Facebook*, *Google+* oder *MySpace* basiert die soziale Vernetzung/Freundschaft durch Followers nicht auf Reziprozität (Kwak, Lee, Park, & Moon, 2010, S. 591). Der ursprüngliche, informationelle Zweck ist immer noch ein wichtiger Grund für die Twitter-Nutzung. Nach Parmelee und Bichard (2012, S. 64) sind *Information Seeking* und *Guidance*, also im weitere Sinne die Informationssuche zur Erleichterung von Entscheidungen und der Meinungsbildung, neben Unterhaltungs-Aspekten immer noch zentrale Nutzungsmotive.

Dass eine klare Abgrenzung nicht möglich ist, verdeutlicht auch die Tatsache, dass mittlerweile eine eigene Ökosphäre von Zusatzprogrammen und Diensten rund um Twitter entstanden ist (siehe Abbildung 3: auf der nächsten Seite), wie Kurzlink-Generatoren, Storytelling-Plattformen für Tweets oder Aggregatoren. Tweets sind zudem häufig der Ausgangspunkt für weitere Informationen, die über Links, Fotos und Videos vermittelt werden. Dadurch entsteht unter Umständen auch der Charakter eines Content Networks, auf welchem Inhalte geteilt werden.

Dennoch entwickelt sich Twitter immer mehr zu einem sozialen Netzwerk. Dies zeigt sich vor allem in der Implementierung zusätzlicher Funktionen: Während zu Beginn der Plattform nur ein Schreiben reiner, auf 140 Zeichen begrenzter,

Textnachrichten möglich war, wuchs Twitter nach und nach um soziale Funktionen, wie das Beantworten oder Teilen von Tweets. Wie stark sich Twitter von der ursprünglichen Idee der rein öffentlichen Informationsvermittlung distanziert hat, zeigt die jüngste Ankündigung von Twitter, Inc. Seit Juli 2015 sind private Nachrichten (*Direct Messages*) nicht mehr auf 140 Zeichen begrenzt (Twitter, 2015), sodass ausführliche, private Interaktionen ermöglicht werden. Diese und weitere Möglichkeiten sowie Konventionen der Kommunikation auf Twitter soll das folgende Kapitel betrachten.

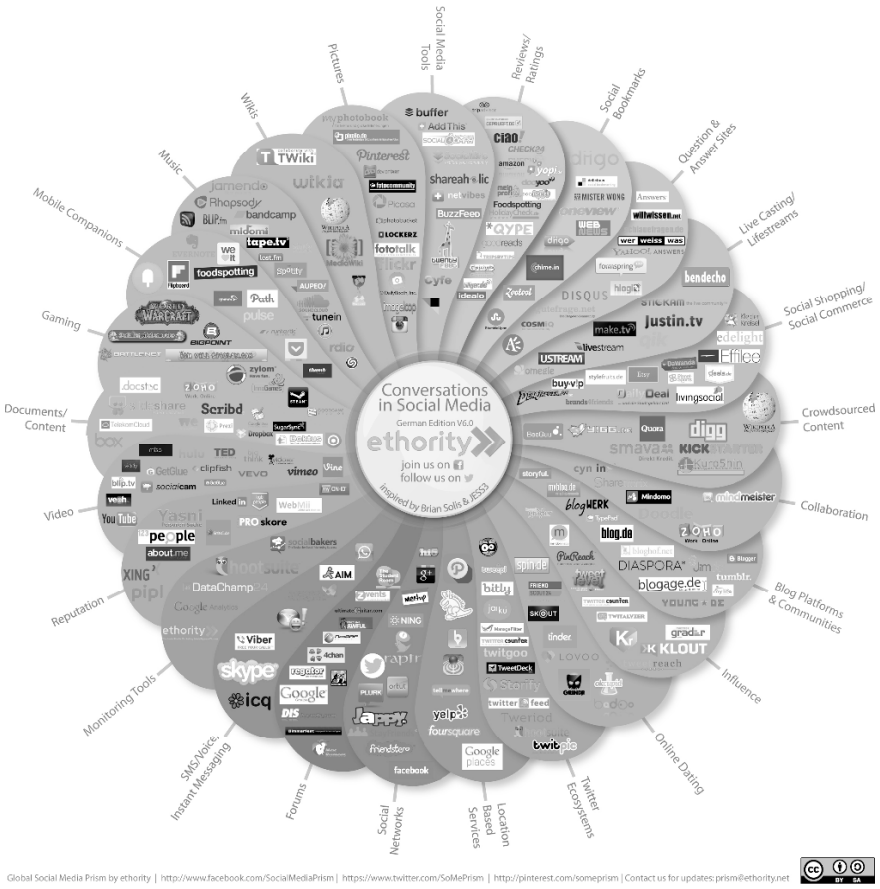

Abbildung 3: Social Media Prisma. Quelle: Ethority (2014).

3.1.2 Konventionen und Struktur der Kommunikation

Twitter bedient sich mehrerer Mechanismen zur Vereinfachung der Kommunikation: Mithilfe eines vorangestellten @ an einem existierenden Benutzernamen können einzelne *Twitter*-Nutzer in einem Tweet direkt adressiert werden. Man spricht hierbei von *Mentions* (*@Username*). Die direkte Beantwortung eines Tweets durch eine andere Nachricht heißt *Reply*. Der Unterschied zu einer reinen Erwähnung in einem Tweet besteht darin, dass bei einem Reply das Mention immer vorangestellt wird (z.B. „@Mustermann: Ich stimme dir zu!"). *Retweets* sind der Kern-Mechanismus auf Twitter: Damit können einzelne Tweets direkt zitiert oder mit anderen Nutzern geteilt werden (Suh, Hong, Pirolli, & Chi, 2010). Ein Retweet ist eine Weiterleitung einer Meldung, früher ersichtlich durch ein „RT @Username" im Fließtext, mittlerweile nur durch eine spezielle Markierung des Tweets (Halavais, 2014, S. 35). Dabei können Nutzer durch Retweets nicht nur Informationen teilen, sondern beispielweise auch Follower unterhalten (durch Teilen unterhaltsamer Tweets) oder mit beigefügten Kommentaren die eigene Zustimmung oder Ablehnung eines Tweets äußern (boyd et al., 2010a). Seit Juni 2014 besteht die Möglichkeit, zu einem Retweet nochmal zusätzlich einen bis zu 140 Zeichen langen Kommentar zu schreiben (Perez, 2014).

Hashtags (*hash*, engl. für „Raute", und *tag*, engl. für „Markierung") sind Wörter oder Abkürzungen, die durch ein vorangestelltes #-Symbol markiert werden. Diese Stichwörter sind nicht moderiert (jeder Nutzer kann eigene Hashtags erstellen) und dienen zur thematischen Vernetzung mit anderen Tweets beziehungsweise gleichen Themen (Parmelee & Bichard, 2012, S. 4). Über die portaleigene Suche oder andere Webdienste können auch nicht registrierte Personen gezielt nach bestimmten Hashtags suchen. Hashtags sind ein nützlicher und sehr wichtiger Mechanismus zur Verbreitung und Verknüpfung von Informationen auf Twitter (Bruns & Moe, 2014, S. 164). Nur so besteht die Möglichkeit, thematisch ähnliche Tweets miteinander zu assoziieren.

Des Weiteren gibt es einen Mechanismus, Tweets von anderen Nutzern zu favorisieren. Diese *Favorites* werden jedoch, im Vergleich zu Retweets seltener eingesetzt (Suh et al., 2010). Die Darstellung der Favorites eines Users erfolgt nicht, wie bei Retweets, auf der eigenen Profilseite im Twitter-Verlauf. Diese sind nur beim jeweiligen favorisierten Tweet aufgelistet. Dennoch ist ein Favorite ein wichtiges Kennzeichen für die Verbreitung einer Nachricht. In der Funktionsweise ist es vergleichbar mit dem *Like* auf Facebook. Tabelle 1 listet nochmal alle Konventionen auf und Abbildung 4 stellt deren Verwendung und Darstellung in einem ausgewählten Tweet dar.

Tabelle 1: Konventionen/Begriffe der Kommunikation auf Twitter

KONVENTION, BEGRIFF	BESCHREIBUNG	BEISPIEL/HINWEIS
TWEET	Kurznachricht auf Twitter, limitiert auf 140 Zeichen. Kann Links, Fotos und Videos enthalten.	Wann scheint endlich die #Sonne! Dann eben #kino… http://t.co/123458abc
MENTION	Erwähnung eines Nutzers in einem Tweet, bzw. Verknüpfung einer Nachricht mit einem Twitter-Nutzer. Vorangestelltes „@“-Zeichen bei Benutzernamen.	Im #Kino mit @musteruser :-)
REPLY	Direkte Antwort auf einen Tweet. Beginnt mit Nennung des kommentierten Nutzers.	@musteruser: Viel Spaß im Kino!
RETWEET	Teilen eines fremden Tweets durch den eignen Nutzeraccount. Nachricht enthält in der Regel „RT@username“.	RT@musteruser: Im #Kino mit musterfrau :-)
HASHTAG	Wörter oder Abkürzungen, die durch ein vorangestelltes „#“-Zeichen markiert werden. Hashtags können gesucht werden und dienen zur Verknüpfung von Themen.	Hätte Lust auf #kino #zeitvertreib #langeweile
FAVORITE	Markierung eines Tweets durch einen Nutzer, dass ihm der Tweet gefällt. Entspricht dem „Like“ auf Facebook.	*Zahl der Favorites wird unterhalb eines Tweets angezeigt (Zahl neben dem Sternchen).*
FOLLOWER	Twitter-Nutzer, der alle Tweets eines anderen Nutzers abonniert hat.	*Follower werden in der Account-Übersicht angezeigt.*
FOLLOWEE	Twitter-Nutzer, dem gefolgt wird/der abonniert wurde.	
FRIEND	Reziproke Follower-Followee-Beziehung.	*Zwei Nutzer sind gegenseitige Follower.*
DIRECT MESSAGE	Private Nachricht, die an eine Person oder Gruppe geschickt wird. Direct Messages werden nicht öffentlich angezeigt.	
LIST	Durch Nutzer verwaltete, öffentliche Liste, von anderen Accounts. Kann abonniert werden.	*Liste mit Accounts von Nachrichtenagenturen.*

Abbildung 4: Konventionen auf Twitter anhand eines Tweets durch den Regierungssprecher. Steffen Seibert (@RegSprecher) retweetet am 11. März 2015 eine Nachricht des Auswärtigen Amtes (@GermanyDiplo) anlässlich des Jahrestags der Naturkatastrophe in Japan 2011. Verwendet werden unter anderem die Hashtags #Japan, #Quake und #Tsunami. Zum Zeitpunkt der Erhebung hatte dieser Tweet 14 Retweets und 32 Favorites. Quelle: Seifert (2015).

Die Interaktion auf Twitter kann unterschiedlich typisiert werden: Anhand der Kommunikationsrichtung, der Kommunikationsebene und Kommunikationsbeziehung. Hinsichtlich der *Richtung* findet bei Twitter sowohl eine unidirektionale, als auch eine bidirektionale Kommunikation statt. Ursprünglich war der Mikroblogging-Dienst primär als Verteiler von Informationen/Neuigkeiten konstruiert (Rogers, 2014), indem Wissen unidirektional von einem Nutzer zu anderen vermittelt und multipliziert werden sollte. Abonniert ein Nutzer beispielsweise einen anderen Nutzer, werden diesem Follower nun automatisch alle Tweets des abonnierten *Followees* angezeigt. Durch die spätere Implementierung weiterer sozialer Interaktions-Funktionen haben sich die Kommunikationsmöglichkeiten jedoch ausgeweitet. Wird ein Tweet kommentiert oder eine private (direkte) Nachricht verschickt, findet eine zweiseitige Kommunikation statt.

Des Weiteren lässt sich die Kommunikation auf Twitter nach Bruns (2014) in drei Ebenen einordnen. Auf der *Mikroebene* findet die auf zwei Nutzer begrenzte, interpersonellen Kommunikation statt: Replies, Mentions und Direct Messages, wobei letztere als einzige Interaktionsform nicht öffentlich ist und somit der Mikroebene am ehesten entspricht. Die *Mesoebene* bildet alle Interaktionen zwischen einem Followee und dessen Followern ab. Diese Kommunikation ist somit auf eine spezifische, relativ konstante und abgrenzbare Nutzer-Gruppe ausgerichtet. Bruns (2014, S. 16) argumentiert, dass Tweets primär von den eigenen Followern gelesen, kommentiert und geteilt würden. Es entstünde somit eine „personal public" (Ebda, S. 17), also persönliche Öffentlichkeit eines Followees. Dieser Effekt der Zielgruppenbegrenzung verstärkt sich durch die Tatsache, dass pro Sekunde durchschnittlich etwa 5.800 Tweets veröffentlicht werden (Twitter, Inc., 2015k) und somit die Wahrscheinlichkeit gering ist, dass der Tweet von Nutzern gelesen wird, die den Verfasser nicht abonniert haben. Bei großen medialen Ereignissen, wie dem Finale der Fußball-Weltmeisterschaft am 13. Juli 2014 mit insgesamt 31,2 Millionen Tweets zum Finale, können es mehr als 600.000 Tweets pro Minute sein (Wiltshire, 2014). Zu der *Makroebene* gehört der Großteil der Kommunikation auf Twitter. Da grundsätzlich jeder Tweet durch die Öffentlichkeit gelesen, durch Hashtags gezielt gesucht und mit Themen verknüpft werden könne, seien Tweets meist Teil eines großen Kommunikationsflusses von in der Popularität schnell steigenden und fallenden Themen/Begriffen (Bruns, 2014, S. 19-20).

Die drei genannten Ebenen sollten nicht als isolierte Strukturen der Twitter-Kommunikation betrachtet werden, sondern als sich teils kreuzende oder überschneidende Kommunikationsstränge: Replies und Retweets können beispielsweise Teil einer übergeordneten Ad-hoc-Diskussion bezüglich eines Themas (verknüpft durch ein gemeinsam genutztes Hashtag) sein.

Schließlich kann die Twitter-Kommunikation noch, wie Tabelle 2 dargestellt, anhand der Beziehung typisiert werden. In Anlehnung an Konert und Hermanns (2002, S. 416) wird einerseits eine Einordnung anhand der Anzahl und Organisation der beteiligten Akteure vorgenommen (von *One-to-One* bis *Many-to-Many*), andererseits nach der Chronologie der Kommunikation (synchron oder asynchron). Die Interaktion auf Twitter erfolgt in der Regel nicht zeitgleich (synchron), wie bei einer Unterhaltung oder einem Telefonat, sondern wird unter Umständen stark zeitverzögert (asynchron) fortgesetzt. Twitter-User können zu einem beliebigen Zeitpunkt Tweets versenden, Nachrichten anderer Nutzer teilen oder kommentieren. Aufgrund der unterschiedlichen Interaktionsmöglichkeiten auf Twitter

sind auch mehrere Interaktionsbeziehungen möglich: *One-to-One* (private Nachrichten, direkte Antworten), *One-to-Few* (private Nachricht an Gruppe) beziehungsweise *One-to-Many* (normaler Tweet) sowie *Many-to-Many* (Tweet innerhalb einer per Hashtag verknüpften Ad-hoc-Öffentlichkeit) möglich.

Tabelle 2: Typisierung der Kommunikations-Beziehungen im Internet, in Anlehnung an Konert und Hermanns (2002, S. 416).

	SYNCHRONE KOMMUNIKATION (NAHEZU SIMULTAN)	**ASYNCHRONE KOMMUNIKATION (ZEITUNABHÄNGIG, VERZÖGERT)**
ONE-TO-ONE	Private Chats/Instant Messaging, Video-Chat (z.B. *Skype*)	E-Mails, Twitter: Direct Message, Reply
ONE-TO-FEW/MANY, FEW/MANY-TO-ONE	Live-Streaming, Newsticker	Webseiten, Blogs, E-Mails Twitter: Direct Message an Gruppe, Tweets an Follower
MANY-TO-MANY	Video-Konferenzen (z.B. *Google+ Hangout*), öffentliche Chat-Rooms	Foren Twitter: Hashtag-verknüpfte Unterhaltung

Charakteristisch für Online-Plattformen ist die non-konforme Textstruktur von Tweets. Wie bei Chatnachrichten oder SMS achten viele (vor allem nicht-kommerziell ausgerichtete) Twitter-Nutzer selten auf Grammatik oder Rechtschreibung. Häufig werden nur Kleinbuchstaben, Abkürzungen oder Neologismen verwendet. Auch Dialekte oder die Vermischung von Sprachen, wie zum Beispiel deutscher Fließtext mit englischen Hashtags, sowie eine fehlende Interpunktion oder eine unkonventionelle Verwendung von Sonderzeichen erschweren die Analyse von Tweets. Hashtags, Mentions und Links werden teilweise in die Satzstruktur integriert.

Abbildung 5 auf der folgenden Seite zeigt beispielhaft Tweets, die für die spätere Analyse (Kapitel 4.3) erfasst wurden. Bei einer automatisierten Analyse durch Computer müssen diese Besonderheiten berücksichtigt und − sofern möglich − bereinigt werden. Mit diesem Problem beschäftigt sich Kapitel 4.3.1 im hinteren Teil dieser Arbeit.

Abbildung 5: Typische Sprache auf Twitter anhand zweier Tweets von Jokolove (2015) und Fahrstuhlprofi (2015).

3.1.3 Datenstruktur von Tweets

Jeder Tweet besteht nicht nur aus dem ersichtlichen Tweet-Text, sondern aus einem Bündel an Meta-Daten, die sich hinsichtlich Inhalt und Umfang nach Tweet und Nutzer unterscheiden[5]. Twitter verwendet hierfür eine ungeordnete Datenstruktur im *JSON*-Format (*JavaScript Object Notation*), welches sich durch eine kompakte, leicht lesbare und schnell zu verarbeitende Textform auszeichnet. Jede Abfrage liefert Datensätze in diesem Format. Die verschachtelte Struktur ermöglicht eine einfache Zuordnung spezifischer Werte zu übergeordneten Wertegruppen. Die einzelnen Informationen werden relativ unsortiert übermittelt, sind jedoch in vier logische Objektgruppen gegliedert. Diese bündeln jeweils spezifische Informationen über User, Tweet, Informationsobjekte und – sofern angegeben – den Ort. Anhang A beschreibt die wichtigsten Felder der einzelnen Objekte.

Abbildung 6 auf der übernächsten Seite liefert einen beispielhaften Überblick über die Datenstruktur eines Tweets. Ein typischer Datensatz bezieht sich immer auf einen einzelnen Tweet, unabhängig ob es ein originärer Tweet, Retweet oder ein Reply ist. Je nach Typ werden dabei unterschiedliche Informationen zur Verfügung gestellt. Ein originärer Tweet, wie in Abbildung 4, umfasst Daten über den Tweet-Inhalt, Sprache, Zeit (und Ort) sowie den Verfasser. Bei Retweets ist zusätzlich der weitergeleitete Tweet, dessen Metadaten (wie unter anderem Favorites

[5] Anmerkung: Häufig wird irrtümlich die reine Textnachricht als Tweet bezeichnet. Streng genommen ist die Nachricht aber nur eines von vielen Merkmalen eines Tweets.

und Retweets) sowie dessen Verfasser ersichtlich, wogegen bei Replies der Umfang begrenzter ist: Hier sind nur die IDs des beantworteten Tweets und des damit verbundenen Twitter-Nutzers einsehbar.

Für inhaltliche Analysen sind vor allem die extrahierten Entities interessant. Hashtags, Hashtag-Trends, Mentions, URLs, Symbole, Bilder und Videos werden automatisch erkannt und in diesem Daten-Array aufgelistet. Zusätzlich sind Informationen über die genaue Position eines Objektes innerhalb des Tweets ersichtlich: Indizes liefern Werte über die Position des ersten und letzten Zeichens eines Objektes und damit auch über die Länge. Bei dem in Abbildung 6 dargestellten Tweet beginnt das Hashtag „#merkel“ mit Zeichen 10 und endet bei Zeichen 17, während die URL bei Zeichen 11 beginnt.

Der hohe Informationsgehalt und die Mehrebenen-Struktur von Twitter-Daten sind mit einfachen Daten-Verwaltungsprogrammen, wie *Microsoft Excel* und *Access* nur schwer zu bewältigen, weshalb Datenbank-Systeme wie SQL oder NoSQL sinnvoll sind. Vor der Datenspeicherung oder spätestens vor der eigentlichen Analyse sollten die gesammelten Daten für eine bessere Übersicht umstrukturiert werden, indem unwichtige Parameter gefiltert und bedeutende Bestandteile umcodiert werden. Es bedarf somit an Programmen oder Programmiersprachen, die die jeweiligen Operationen zur Restrukturierung des Datensatzes ermöglichen und die aufbereiteten Daten dann in Datenbanken schreiben. Eine sehr geeignete, da einfache und übersichtliche, Sprache ist Python, welche das folgende Kapitel kurz vorstellt.

Abbildung 6: Datenstruktur eines Tweets. Eigene, gekürzte Darstellung in Anlehnung an Krikorian (2010).

3.2 Programmiersprache Python

Entwickelt im Jahr 1990 durch Guido van Rossum, etablierte sich die Programmiersprache *Python* mittlerweile als Standard für deskriptive, computergestützte Studien (Millman & Aivazis, 2011, S. 9). Python ist eine interpretierte, interaktive, objekt-orientierte Programmiersprache, die eine sehr einfache und übersichtliche Syntax aufweist (Sanner, 1999, S. 3). Während die Sprache ursprünglich nicht für wissenschaftliche Zwecke gestaltet war, entstanden im Lauf der Zeit mit zunehmendem Interesse durch die Wissenschaft mehrere spezialisierte Module, wie *SciPy*, *matplotlib* und *NumPy*. Diese Pakete beinhalten etwa Funktionen zur Darstellung von Plots oder zur Ausführung einfacher, numerischer Funktionen bis hin zu komplexen Berechnungen (Millman & Aivazis, 2011, S. 10). Eines dieser Pakete ist auch *Tweepy*.

Tweepy[6] ist ein Python-Modul, das speziell zur Interaktion mit den Twitter APIs entwickelt wurde. Es unterstützt Anwender bei der Autorisierung und Durchführung von Abfragen. Zusätzlich zu den normalen Abfrage-Methoden über die REST und Streaming API stehen weitere nützliche Funktionen zur Verfügung. So berücksichtigt Tweepy bei Bedarf die Bandbreiten-Limitierung der REST API und plant beziehungsweise pausiert die definierten Requests. Tweepy wird in dieser Arbeit zur Datensammlung verwendet (siehe Listing 8, Kapitel 4.1.2).

Python weist mehrere Eigenschaften auf, die für eine wissenschaftliche Nutzung ohne vorhandene, fortgeschrittene Programmierkenntnisse von Vorteil sind: Eine intuitive, klar strukturierte Syntax, die eine gute Lesbarkeit[7] und somit auch einfachere Programmierung fördert (Russell, 2013, S. xv). Die Lernkurve ist dadurch hoch und die Einarbeitungszeit kurz. Der Programmcode ist plattformunabhängig und kann auf nahezu jedem Betriebssystem (wie *Windows*, *Mac OS* oder *Linux*) ausgeführt werden.

[6] http://www.tweepy.org/
[7] Der Programmcode liest sich wie ein Text.

Listing 1: Beispiel für ein Python-Skript

```
# Laden von Modulen
import tweepy
from slistener import tweepylistener

#Definition von Parametern
list_terms = ["#obama"]
listen = tweepylistener(api)
stream = tweepy.Stream(auth, listen, timeout=600.0)

#Definition einer Suchabfrage
while True:
    try:
        stream.filter(track=list_terms, async=False)
        break
    #Vorgehen bei Fehlern: Abbruch
    except Exception, e:
        sys.exit(1)
```

Listing 1 zeigt ein sehr einfaches Skript, das mit Python zum Sammeln von Tweets zu einem bestimmten Begriff geschrieben wurde. Das Programm bedient sich dabei an vordefinierten Klassen, die hier bereits im Modul `Tweepy` integriert sind oder vorher manuell erstellt wurden (Klasse `slistener`). Aufgrund dieser modularen Struktur können auf einfache und übersichtliche Weise Befehle ausgeführt werden, für die vorher nur wenige Parameter definiert werden müssen. In diesem Fall wurde mit `list_terms` nur ein Suchterm angegeben.

Der geschriebene Programmcode wird unmittelbar interpretiert, es entfällt also das aufwändige Kompilieren vor der Ausführung. Als objekt-orientierte Sprache erlaubt Python eine Modularisierung von Programmteilen, die zu einem anderen Zeitpunkt im Code wiederverwendet werden können, als dynamische Sprache können Paramater während der Ausführung hinzugefügt oder geändert werden (Bird, Klein, & Loper, 2009, S. xiii). Python ist zudem, im Vergleich zu ähnlichen Sprachen wie *MatLab*, als Open Source lizenziert und damit kostenlos. Idealerweise ist die Kern-Datenstruktur von Python im JSON-Format – dem gleichen Format, in dem auch Twitter-Daten zur Verfügung gestellt werden. Es gibt mittlerweile ein sehr großes Ökosystem mit vielen Programmpaketen für unterschiedlichste Zwecke – von Schnittstellen zu Datenbanken oder anderen Programmen, bis hin zu eigenständigen Bibliotheken zur Visualisierung von Daten. Besonders aufgrund der hohen Lesbarkeit und der großen Popularität bei Twitter-basierten Analysen, wird Python in dieser Arbeit als Programmiersprache verwendet.

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung - Nicht kommerziell 4.0 International Lizenz (http://creativecommons.org/licenses/by-nc/4.0/deed.de) veröffentlicht, welche für nicht kommerzielle Zwecke die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.
Etwaige Abbildungen oder sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende oder der Quellreferenz nichts anderes ergibt. Sofern solches Drittmaterial nicht unter der genannten Creative Commons Lizenz steht, ist eine Vervielfältigung, Bearbeitung oder öffentliche Wiedergabe nur mit vorheriger Zustimmung des betreffenden Rechteinhabers oder auf der Grundlage einschlägiger gesetzlicher Erlaubnisvorschriften zulässig.

4 Methoden zur Erfassung, Verwaltung und Auswertung von Tweets

Die in Kapitel 2 vorgestellten Studien verwendeten unterschiedliche Herangehensweisen zur Erhebung und Analyse von Daten auf Twitter: Je nach zeitlicher Fokussierung wurde die Streaming API oder die REST API, in wenigen Fällen auch ein kostenpflichtiger Datensatz eines Datenhändlers verwendet. Zudem wurden die erhobenen Daten in Datenbanken oder einfachen Textdateien gespeichert, die quantitativen oder qualitativen Analysen erfolgten manuell oder computergestützt.

Welche Möglichkeiten der Datenerhebung auf Twitter bestehen und wie diese eingesetzt werden, soll im nächsten Kapitel genauer analysiert werden. In Anlehnung an den üblichen Prozess der Analyse von Online-Daten (Datenerhebung – Datenspeicherung – Datenauswertung) gliedert sich dieser Teil der Arbeit in drei Kapitel. Kapitel 4.1 befasst sich zunächst mit den technischen Aspekten der Datenerhebung und stellt hierfür die beiden kostenlosen Twitter-Schnittstellen gegenüber. Zudem wird auch kurz auf alternative, kostenpflichtige Methoden eingegangen sowie die Vorteile und Grenzen der einzelnen Ansätze anhand praktischer Beispiele dargestellt. Kapitel 4.2 geht schließlich auf den Prozess der Datenspeicherung und -verwaltung ein. Die Wahl einer geeigneten Speichermethode ist entscheidend für die weitere Datenverwaltung, Datenpflege und letztlich auch die Analyse. Neben der einfachen Speicherung in einzelnen Dateien werden unterschiedliche Datenbank-Konzepte angesprochen. Das abschließende Kapitel 4.3 befasst sich mit Analyseverfahren. Für die Auswertung umfangreicher und detaillierter Daten sozialer Netzwerke sind manuelle Verfahren, wie das Kodieren einzelner Tweets, nicht praktikabel, beziehungsweise nur unter großem Aufwand realisierbar. Deswegen steht eine Vielzahl computergestützter Verfahren zur Verfügung, die den Prozess vereinfachen: Von der automatisierten Inhaltsanalyse mit *Bag of Words* Repräsentation bis hin zu vollständigen semantischen Analysen.

© Der/die Autor(en) 2016
F. Pfaffenberger, *Twitter als Basis wissenschaftlicher Studien*,
DOI 10.1007/978-3-658-14414-2_4

4.1 Möglichkeiten der Datensammlung

Twitter-Daten können mit unterschiedlichen Methoden erhoben werden. In der Praxis haben sich, je nach Ausgangslage, mehrere Verfahren etabliert. Twitter ermöglicht seit 2014 sechs von 1.300 beworbenen Forschungsprojekten umfassenden Datenzugriff (Krikorian, 2014). Neben diesen *Data Grants* besitzen noch mehrere Institutionen (wie das *MIT* oder die *Library of Congress* in den USA) und Geschäftspartner (z.B. *IBM*, *Brandwatch*) privilegierte Rechte. Die meisten Interessenten für Twitter-Daten müssen jedoch andere, hinsichtlich Umfang und Informationsgehalt deutlich beschränkte, Datenzugänge wählen.

Für „normale“ Twitter-Nutzer besteht die Möglichkeit, eigene Tweets als lokale Kopie zu archivieren, wobei sich der Datenexport aufgrund des Formats und der Einschränkung auf eigene Tweets nicht für Forschungszwecke eignet. Für eine detaillierte Datenabfrage stellt Twitter zwei Typen von Schnittstellen zur Verfügung: die Streaming API und die REST APIs. Diese unterscheiden sich hinsichtlich Datenumfang, Zeitraum und Funktionalität. Eine Nutzung dieser APIs setzt, im Vergleich zu simplen Datenexporten aus Twitter, technisches Know-How und ein entsprechend einsetzbares IT-System voraus. Die in folgenden Kapitel vorgestellten Twitter APIs unterstützen eine Vielzahl gängiger Programmiersprachen zur Steuerung der Datenabfragen und -verarbeitung, wie *Java*, *ASP*, *Ruby* und *Python*.

Alle Schnittstellen benötigen für den Daten-Zugriff einen Twitter-Account und eine darin registrierte Anwendung, die die Datenabfragen verwaltet. Diese App autorisiert schließlich die Endbenutzer für die Verwendung der Twitter APIs, wobei der Zugriff entweder von einem Programm allein (*App Auth*) oder von mehreren Nutzern parallel (*User Auth*) über einen Dienst erfolgen kann. Die Autorisierung muss vor Beginn der Abfrage einmalig durchgeführt werden. Listing 2 zeigt einen typischen Vorgang zur Autorisierung eines Programms bei den Twitter APIs. Hierfür verlangt die API den *Consumer Key* und das *Consumer Secret* aus der App-Verwaltung[8], die in den Programmcode eingetragen werden. Wenn die Schnittstelle parallel von mehreren Programmen über dieselbe App abgerufen werden soll (also per User Auth), benötigt der Anmelde-Prozess zusätzlich *Access Token* und *Access Token Secret*.

[8] Der Aufruf der App-Verwaltung erfolgt über https://apps.twitter.com/app/

Listing 2: OAuth-Autorisierung bei der Twitter API.

```
import tweepy

# Authentifizierung mittels Zugangsdaten, die unter
https://apps.twitter.com/app abgerufen werden

consumer_key="<Consumer/API Key>"
consumer_secret="<Consumer/API Secret>"
access_key="<Access Token>"
access_secret="<Access Token Secret>"

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
# Wenn UserAuth-Methode:
auth.set_access_token(access_key, access_secret)

api = tweepy.API(auth)
```

Neben der Möglichkeit, Twitter-Daten mithilfe (selbst) geschriebener Programmroutinen über die Schnittstellen abzurufen, können Daten zusätzlich von Drittanbietern erworben werden. Hierfür werden weder einer Registrierung bei Twitter, noch Programmierkenntnisse vorausgesetzt. Die folgenden Unterkapitel stellen diese drei Varianten kurz vor und vergleichen sie abschließend.

4.1.1 Streaming API

Die Streaming API ist die wohl meistgenutzte Datenquelle, da sie aufgrund ihres Datenumfangs und ihrer Handhabung ideal für großvolumige, langfristige quantitative Analysen ist (Parmelee & Bichard, 2012, S. 56). Twitter-Daten werden hier als konstanter Datenstrom nahezu in Echtzeit gesendet. Man spricht diesbezüglich von einer *push*-basierten API. Es werden also nach einmaliger Anfrage kontinuierlich Daten an den Empfänger übermittelt, wogegen beim *pull*-Prinzip spezifische Daten immer erst nach Aufforderung geliefert werden würden (Kumar et al., 2014, S. 5).

Die Streaming API gliedert sich in drei *Streams*, die einen jeweils unterschiedlichen Endpunkt zur Datennutzung haben. *User Streams* übermitteln alle Daten, die zu einem spezifizierten Nutzer gehören (Tweets der gefolgten Accounts,

Replies auf die eigenen Tweets, und – wenn autorisiert[9] – Direct Messages). Die *Site Streams* bilden die Mehrnutzer-Variante der *User Streams* ab und sind vor allem für öffentliche Webanwendungen gedacht. Während *User* und *Site Streams* nur die Daten einzelner Nutzer enthalten, liefern die *Public Streams* alle öffentlich erhältlichen Daten auf Twitter und ermöglichen, spezifische Nutzer oder Themen zu verfolgen. Sie sind somit am besten zum Data Mining geeignet.

Alle Streams stehen in drei Bandbreiten zur Verfügung: *Spritzer*, *Gardenhose* und *Firehose*, die entsprechend maximal 1, 10 und 100 Prozent aller veröffentlichten Tweets zu einem Suchterm pro Sekunde übermitteln (Gaffney & Puschmann, 2014, S. 57). Die Funktionsweise dieser Limits soll das folgende Beispiel anhand der *Spritzer* genauer erklären: Die APIs benötigen in den meisten Fällen, beziehungsweise für die meisten Abfragemethoden[10] einen Suchterm. Dieser könnte ein Begriff („#merkel"), eine spezifische Nutzer-ID oder ein Koordinaten-Bereich sein. Übersteigt das Gesamtergebnis des Suchterms ein Prozent des momentanen Twitter-Volumens – macht also das potentielle Suchergebnis in einer Sekunde mehr als ein Prozent des gesamten Tweet-Volumens weltweit aus – werden nur ein Prozent der Ergebnisse ausgegeben. Mit welchen Kriterien (z.B. Zeitstempel, Relevanz, Popularität des Nutzers) diese von der API ausgegebenen Tweets gefiltert werden, ist nicht bekannt.

Während Spritzer für alle Nutzer kostenlos ist, wird der Zugriff zur Gardenhose nur nach Anfrage und in begründeten Fällen (Forschungszwecke, Online-Dienste) erteilt. Der Zugang zur Firehose ist stark reglementiert und wird meist nur im Zusammenhang mit wirtschaftlichen, kostenpflichtigen Kooperationen freigegeben: Aufgrund der umfassenden und detaillierten Datenmenge besitzen nur wenige Unternehmen, wie die Datenhändler *Gnip* und *DataSift*, diese Möglichkeit.

Der Datenstrom von Spritzer und Gardenhose steht außerdem in zwei unterschiedlichen Methoden zur Verfügung: `Sample` und `Filter`. Erstere Variante erzeugt ein Datensample von einem Prozent aller Tweets, wogegen letztere gefilterte übermittelt: Mittels `Track` Befehl können mehrere kommagetrennte Werte abgefragt werden. Die Schnittstelle gibt dann alle Tweets aus, die diese Begriffe beinhalten, sofern das Volumen nicht ein Prozent des momentanen Twitter-Volumens übersteigt. Analog wird `Follow` für Benutzer-IDs und `Locations` für Koordina-

[9] Die Autorisierung erfolgt immer auf Nutzerebene. Jeder Anwender kann über die API nur die Direct Messages verwalten, die mit dem jeweilig autorisierten Account verknüpft sind. Forschende haben also keine Möglichkeit, private Nachrichten anderer Nutzer zu lesen.

[10] Ausgenommen der Sample-Methoden.

ten verwendet. Ein Sprachfilter steht momentan nicht zur Verfügung. Für alle Methoden gibt es außerdem Einschränkungen bei der Anzahl an Filter-Parameter: So können zeitgleich höchstens 400 Wörter (*Keywords*), 5.000 User-IDs und 25 Orte abgefragt werden (Twitter, Inc., 2015e).

Morstatter, Pfeffer, Liu und Carley (2013) verglichen den Standard-Datenoutput der Streaming API (Spritzer) mit den vollständigen Daten der Firehose hinsichtlich Daten-Abdeckung und Stichprobenqualität. Hierfür wurden unter anderem Tweets nach Hashtag und Ort gefiltert und gegenübergestellt sowie Trend-Themen ermittelt. Die Untersuchung ergab, dass es sich beim 1-prozentigen Sample der *Spritzer* um keine vollständig verlässliche Stichprobe handelt, sondern die Qualität des Samples von den analysierten Begriffen/Themen/Nutzern abhängt. Bei geringen Fallzahlen von Hashtags oder Themen traten kleinere Abweichungen auf. Bei zu großen Fallzahlen, also beispielsweise zu vielen Tweets, die ein bestimmtes Hashtag enthalten, sank die Genauigkeit des Samples. Dies lässt sich damit begründen, dass nach Überschreiten des 1%-Limits der Spritzer nicht mehr alle betreffenden Tweets übermittelt werden. Werden also zu einem Zeitpunkt mehr als ein Prozent aller Nachrichten auf Twitter zu einem gefilterten Begriff /Thema/Nutzer verfasst, wird der Datenstrom auf diesen Anteil (respektive 10% bei der Gardenhose) gedeckelt und die Menge an „verloren gegangenen" Tweets als Zahl ausgegeben.

Ein etwaiges Erreichen des Limits kann jedoch durch eine vorherige Eingrenzung der Daten mittels mehrerer Filter vermieden werden, sodass letztlich in diesen Fällen alle relevanten Daten vollständig erhoben werden können. Dennoch kann es in einzelnen Fällen, bei sehr großem Tweet-Volumen zu einem einzelnen Thema (beispielsweise bei medialen Großereignissen wie Katastrophen), nicht möglich sein, durch Filter die Tweet-Anzahl so einzugrenzen, um die Bandbreitenbeschränkung nicht zu überschreiten. Auch ist ein sehr spezifisches Filtern der Daten nicht immer erwünscht und sinnvoll (bei globalen oder sehr unspezifischen Themen). In diesen Fällen müssen die Tweets, die durch die Deckelung des Datenstroms nicht übermittelt wurden, nachträglich und manuell mit Hilfe der REST API eingespeist werden (siehe Anwendungsbeispiel in Kapitel 4.1.2). Möglich wären auch aufgeteilte Abfragen, die synchron von unterschiedlichen Endpunkten gestartet werden: Je Hashtag wird eine eigene Abfrage über einen separaten Account mit App gestartet – die Zusammenführung der Daten erfolgt dann während des Sammelns (Schreiben in die gleiche Datenbank) oder nachträglich.

4.1.1.1 Anwendungsbeispiel: Sammeln von Echtzeitdaten auf Twitter

Das folgende Beispiel zeigt einen typischen Prozess zum Sammeln von Tweets: Das Programm in Listing 3 erfasst in Echtzeit Tweets mit Hashtag *#Obama* und gibt diese direkt im Programmfenster (*Shell*) aus. Zur Vereinfachung des Skriptes wird hier also auf ein Speichern und weiteres Verarbeiten der Tweets verzichtet – dieses Vorgehen wird in Kapitel 4.2 besprochen.

Nach der Autorisierung des Programms bei der Streaming API über die *OAuth*-Methode wird zuerst die Klasse `tweepylistener` erstellt, die das Python-Paket Tweepy nutzt und das Vorgehen bei Eintreffen eines neuen Tweets definiert. Hier werden zur Veranschaulichung des Daten-Outputs alle eingehenden Daten in das Programmfenster geschrieben. Möglich wäre aber auch ein Abspeichern in eine Datei oder Datenbank. Etwaige Fehler erscheinen ebenfalls im Shell-Fenster. Diese Klasse kann jederzeit und an jeder Stelle des Programms aufgerufen werden. Schließlich werden die wesentlichen Parameter der API-Abfrage definiert: Der Suchterm entspricht einer vorher definierten Liste von Begriffen (hier: `obama`). Das Vorgehen bei Tweets, die der Abfrage beziehungsweise dem Suchterm entsprechen, definiert bereits die Klasse `tweepylistener`, die hier schließlich abgerufen wird. Des Weiteren erfolgt eine explizite Einbindung der Streaming API und die Definition dafür notwendiger Parameter, wie Autorisierungs-Daten und eine Time-Out-Zeit für die Anfragen. Sollte eine Suche nach 600 Sekunden kein Ergebnis liefern, endet der Prozess automatisch. Zuletzt werden noch die Suchfilter über die Variable `list_terms` integriert.

Listing 3: Simpler Vorgang zum Sammeln von Tweets mit dem Begriff „obama“ und direkter Ausgabe im Programmfenster

```
import tweepy

# Authentifizierung
consumer_key="<Consumer/API Key>"
consumer_secret="<Consumer/API Secret>"
access_key="<Access Token>"
access_secret="<Access Token Secret>"
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_key, access_secret)
api = tweepy.API(auth)

# Definiere Vorgehen bei neuen Tweets und Fehlern
class tweepylistener(tweepy.StreamListener):
```

```
    def on_data(self, data):
        print(data)
        return True
    def on_error(self, status):
        print(status)

# Definiere Parameter des Streams
if __name__ == '__main__':
    list_terms = ["#obama"]
    listen = tweepylistener(api)
    stream = tweepy.Stream(auth, listen, timeout=600.0)
    stream.filter(track=list_terms)
```

Der Suchfilter `track` weist eine spezielle Systematik auf, die es erlaubt, Suchbegriffe mit den logischen Operatoren AND und OR zu verknüpfen. Diese Systematik verknüpft Wörter, die nur durch ein Leerzeichen getrennt sind, als Konjunktionen und behandelt kommaseparierte Begriffe als Disjunktionen. Dabei berücksichtigt der Algorithmus des Suchfilters auch explizit Punktationen und Sonderzeichen, wobei diese nicht in #hashtags und @mentions erlaubt sind und somit nur die Suche in normalem Fließtext davon betroffen ist. Tabelle 3 veranschaulicht den Such-Mechanismus.

Tabelle 3: Operatoren des Track Filters der Streaming API. In Anlehnung an Twitter, Inc. (2015f).

SUCHPARAMETER	BERÜCKSICHTIGTE BEGRIFFE/TWEETS	NICHT BERÜCKSICHTIGTE BEGRIFFE/TWEETS
OBAMA	Obama / #Obama / OBAMA / @obama / obama. / http://obama.com	BarackObama / #presidentobama / obamaUSA
OBAMA'S	Watching Obama's speech.	Watching @Obama's speech.
OBAMA SPEECH, OBAMA TV	Watching #obama speech on TV #Obama is on TV! Watching obama speech.	Nice speech on TV! Watching #obama…
OBAMA, CASTRO	#Obama handshake with #castro #obama is now on TV #castro is meeting POTUS	Obamas handshake with #raulcastro

Des Weiteren besteht die Möglichkeit, mittels `follow`, `locations` und `language` Filter nur Tweets bestimmter Nutzer, Orte oder Sprachen zu sammeln. Dieser Mechanismus ist nahezu analog zum `Track` Filter und wird durch kommagetrennte Listen definiert. Logische Operatoren stehen hier allerdings nicht zur Verfügung.

Der Suchfilter aus obiger Streaming API Anfrage ergibt einen beispielhaften Output wie in Listing 4. In Abhängigkeit der Häufigkeit der Suchergebnisse würden sequentiell alle dem Suchfilter entsprechenden Tweets angezeigt werden. Bei einem Fehler erschiene der entsprechende Fehlercode (siehe Code `420` am Ende von Listing 4).

Listing 4: Shell-Output der Abfrage aus Listing 3 für einen Tweet. Ausgabe-Code durch Autor gekürzt.

```
{"created_at":"Sat Apr 11 11:26:02 +00002015",
"id":586852702115663872, id_str":"58685270211566387",
"text":"ABC News:  Watch President Obama Geek Out While
Meeting Usain Bolt . More #Obama #news - http:\/\/t.co\/
OkWn16C9Mn","source":"\u003ca href=\"http:\/\/www.1sthead-
lines.com\" rel=\"nofollow\"ObamaInTheNews ","trun-
cated":false, "in_reply_to_status_id":null,
"in_reply_to_status_id_str":null,"in_reply_to_user_id":null,
"in_reply_to_user_id_str": null,"in_reply_to_screen_name":
null,"user":{"id":45326213,"id_str":"45326213","name":
"1stHeadlines","screen_name": "ObamaInTheNews","location":
"USA","url":"http:\/\/www.1stheadlines.com\/obama.htm",
"description":"Breaking news stories about Barack Obama from
top online news sources.","protected":false,"verified":
false,"followers_count":1616,"friends_count":0,
"listed_count":68,"favourites_count":0,"statuses_count":
113751,"created_at":"Sun Jun 07 11:48:45 +0000 2009",
[...]
"favorite_count":0,"entities":{"hashtags":[{"text":"Obama",
"indices":[74,80]},{"text":"news","indices":[81,86]}],
"trends":[],"urls":[{"url":"http:\/\/t.co\/OkWn16C9Mn","ex-
panded_url":"http:\/\/tinyurl.com\/ltvgvk","display_url":
"tinyurl.com\/ltvgvk","indices":[89,111]}],
"user_mentions":[],"symbols":[]},"favorited":false,
"retweeted":false,"possibly_sensitive":false,"filter_level":
"low","lang":"en","timestamp_ms":"1428751562037"}
420
```

Twitter stellt eine Liste mit Fehlercodes und Erläuterungen zur Verfügung[11]. Die Bedeutung reicht von einfachen Autorisierungs-Fehlern bis hin zu komplexen Problemen wie den *Rate Limits*. Letztere treten auf, wenn gleichzeitig zu viele Anfragen über eine autorisierte App innerhalb eines Zeitfensters gestartet werden. Sollte dieser Fehlercode missachtet werden, droht eine temporäre Sperrung (*Blacklisting*) der IP-Adresse[12]. Um das zu vermeiden, ist eine differenzierte Behandlung einzelner Fehler sinnvoll. Dementsprechend wird die in Listing 3 definierte Klasse `tweepylistener` um einige Fälle erweitert.

Listing 5: Erweiterte Suchklasse der Streaming API mit zusätzlichen Fällen

```
class tweepylistener(tweepy.StreamListener):
 def __init__(self, api = None):
  self.api = api or API()
  self.deleted= open('geloescht.txt', 'a')

 # Bestimme Vorgehen für unterschiedliche Tweet-Typen
 def on_data(self, data):
  if "in_reply_to_status" in data:
   self.on_status(data)
  elif "delete" in data:
   delete = json.loads(data)["delete"]["status"]
   if self.on_delete(delete["id"], delete["user_id"]) is
      False:
    return False
  elif "limit" in data:
   if self.on_limit(json.loads(data)["limit"]["track"]) is
      False:
    return False
  elif "warning" in data:
   warning = json.loads(data)["warnings"]
   print warning["message"]
   return False

 # Bestimme Vorgehen in unterschiedlichen Situationen
 # Fall 1: neuer Status-Tweet
 def on_status(self, status):
  print(status)
  return True
```

[11] Siehe https://dev.twitter.com/streaming/overview/connecting
[12] Weitere Informationen unter: https://dev.twitter.com/rest/public/rate-limiting

```
  # Fall 2: User löscht Tweet nach gewisser Zeit
  def on_delete(self, status_id, user_id):
    self.deleted.write( str(status_id) + "\n")
    return

  # Fall 3: Streaming API Rate Limit
  def on_limit(self, track):
    sys.stderr.write(time.strftime("%Y%m%d-%H%M%S") +
                     ">> Rate Limit: " + str(track))
    return

  # Fall 4: Fehlermeldung mit Fehlercode
  def on_error(self, status_code):
    sys.stderr.write(time.strftime("%Y%m%d-%H%M%S") +
                     ">> Fehler: " + str(status_code) + "\n")
    time.sleep(60)
    return False

  # Fall 5: Verbindungs-Timeout keine Reaktion
  def on_timeout(self):
    sys.stderr.write(time.strftime("%Y%m%d-%H%M%S") +
                     ">> Timeout, warte für 120 Sekunden\n")
    time.sleep(120)
    return

def main():
  list_terms = ["#obama"]
  listener = tweepylistener(api)
  stream = tweepy.Stream(auth, listener, timeout=600.0)

  while True:
    print time.strftime("%Y%m%d-%H%M%S") +
                        ">> Streaming gestartet... beobachte
                        und sammle"
    print time.strftime("%Y%m%d-%H%M%S") +
                        ">> Suche Twitter nach: " +
                        str(list_terms)[1:-1]
    try:
      stream.filter(track=list_terms, async=False)
      break
    except Exception, e:
      time.sleep(60)

if __name__ == "__main__":
  main()
```

Die erweiterte Klasse in Listing 5 definiert nun das Vorgehen des Programms für die unterschiedlichen Arten von Tweets und Fehlermeldungen. Zuerst werden die einzelnen Tweets nach bestimmten Signalwörtern durchsucht und in Typen eingeteilt. Enthält ein Datensatz beispielsweise eine `delete`-Anweisung in Verbindung mit einem Account- oder Status-ID-Datenfeld, handelt es sich um eine Lösch-Anweisung. Enthält er ein `limit`, so handelt es sich um eine Mitteilung, dass das Bandbreiten-Limit überschritten wurde.

Das Vorgehen in den einzelnen Situationen wird schließlich genauer vorgegeben: Bei einem normalen Tweet (inklusive Retweet) wird der gesamte Tweet-Datensatz ausgegeben. Wenn ein `delete`-Befehl enthalten ist, schreibt das Skript die entsprechende Tweet-ID in die vorher definierte Textdatei `geloescht.txt`. So erhält man im Nachhinein eine Auflistung aller gelöschten Tweets. Diese können bei Bedarf manuell entfernt werden. Unter Umständen ist es aber sinnvoll, diese Tweets zu behalten und die Löschanweisung als zusätzliche Information zu verarbeiten: Sei es als Anzeichen späterer Selbstselektion oder als Reaktion auf eine kritische Resonanz durch andere Nutzer.

Im Fall einer Limit-Überschreitung erfolgt die Ausgabe der Meldung im Programm-Fenster. Diese Mitteilung enthält auch die Zahl der nicht erfassten, sozusagen verlorenen Tweets. Bei Meldungen mit Fehlercode wird dieser angezeigt und alle weiteren Abfragen über die Streaming API um eine Minute pausiert. Wenn ein Timeout der Anfrage eintritt, also nach einer vorher definierten Zeit kein dem Filter entsprechender Tweet übermittelt wird, pausiert das Programm für zwei Minuten und setzt danach die Aktivität fort.

Diese Routine behandelt einen Großteil möglicher Probleme automatisch. Einschränkungen, die aufgrund technischer Rahmenbedingungen der Streaming API existieren, wie die Bandbreiten-Limitierung, können hiermit allerdings nicht umgangen werden. In diesem Fall ist es bestenfalls möglich, analog zum Verfahren mit Delete-Anweisungen, die Limit-Meldungen mit der Anzahl der nicht erfassten Tweets in einer separaten Log-Datei zu speichern, sodass zumindest Zahlen über die Missings vorliegen. Ein nachträgliches Erfassen der ausgelassenen Tweets wäre nur über die REST API möglich.

4.1.1.2 Bewertung der Streaming API

Zusammenfassend ist die Streaming API ein geeignetes Mittel zur kontinuierlichen Erfassung einer großen Anzahl von Tweets über einen längeren Zeitraum. Mit Hilfe intelligenter Programmroutinen läuft der Prozess der Datensammlung nahezu vollständig automatisiert. Die Echtzeit-Daten sind kostenlos und in nahezu

unbegrenztem Umfang verfügbar, wodurch nicht nur Ad-hoc-Analysen, sondern auch Langzeitstudien möglich sind. Zwar ist der Zugriff für die meisten Nutzer und somit auch für viele Forschende nur auf ein Prozent des gesamten Tweet-Aufkommens begrenzt. Eine gezielte Wahl geeigneter Filterparameter könnte dieses Problem in vielen Fällen umgehen. Jedoch besteht dann die Gefahr, relevante Tweets durch zu strikte Filter auszusondern. Hinsichtlich der Zielsetzung, möglichst viele Daten zu sammeln, ist die Verwendung der *Public Streams* die einzige sinnvolle Methode. *User Streams* und *Site Streams* bieten nur eine Plattform für Web-Anwendungen, in denen sich Nutzer mit ihrem Benutzerkonto einwählen können und dienen daher nicht zum Sammeln von Daten.

Zudem gibt es Verzerrungen zwischen dem über die Streaming API zur Verfügung gestellten Sample und der Grundgesamtheit, die momentan noch nicht kompensiert werden können (Morstatter et al., 2013, S. 9). Eine Repräsentativität dieses automatischen Samples ist demnach nicht gegeben. Außerdem gibt es seitens Twitter keine Angabe über die Generierung dieser Stichprobe.

Es gilt zu beachten, dass der Abfrageprozess von Tweets aufgrund der *Push*-Architektur während der kompletten Sammlung laufen muss. Die Streaming API übermittelt kontinuierlich in Echtzeit Daten, die durch ein Skript sofort erfasst werden. Ein nachträgliches Sammeln ist demnach nicht möglich. Somit resultiert aus jeder Unterbrechung der Internetverbindung oder des Prozesses ein Datenverlust.

Die Tatsache, dass kein Abrufen historischer Daten möglich ist, birgt weitere Probleme: Trends müssen frühzeitig (also eigentlich ad hoc) erkannt werden, um möglichst viele relevante Tweets erheben zu können. Dies ist allerdings nur bei langfristig terminierten Ereignissen (wie Wahlen oder Sportereignissen) zuverlässig möglich. Spontane Bewegungen, Themen oder Ereignisse wie politische Skandale oder Naturkatastrophen können mit dieser Methode erst post hoc oder zumindest zeitlich verzögert erfasst werden. Letztlich fehlt bei Echtzeitdaten auch die Möglichkeit, die Anzahl von Retweets oder Favorites direkt abzufragen. Da die übermittelten Twitter-Daten immer Momentaufnahmen zur Veröffentlichung eines Tweets sind, ist die Wahrscheinlichkeit hoch, dass jeder neue Tweet keine Favorites und Retweets hat[13].

[13] Da der Tweet sofort bei Veröffentlichung über die Streaming API übermittelt wird und die Wahrscheinlichkeit sehr gering ist, dass der Tweet Millisekunden nach Veröffentlichung bereits Retweets oder Favorites hat, steht er folglich immer im Ursprungszustand (ohne Retweets und Favorites) im Datensatz.

Eine Erfassung (späterer) Retweets und Favorites wäre innerhalb der *Public Streams* nur über die Betrachtung des jeweils letzten/aktuellsten Retweets möglich: Die Meta-Daten eines Retweets beinhalten immer die Anzahl an Retweets und Favorites des Original-Tweets zum Veröffentlichungszeitpunkt des Retweets (siehe Abbildung 7). Wie die schematische Darstellung zeigt, taucht folglich ein Tweet unter Umständen mehrmals in einer Datensammlung auf: Zumindest einmal als Tweet und eventuell mehrmals als eingebetteter Retweet. Diese Methode könnte letztlich auch Tweets erfassen, die außerhalb des Betrachtungszeitraums liegen, aber als aktuelle Retweets weiterhin im Datenset auftauchen.

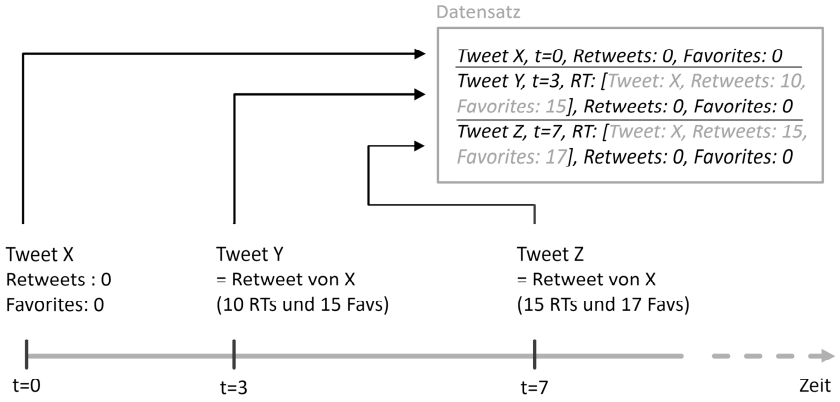

Abbildung 7: Ansatz zum nachträglichen Erfassen von Favorites und Retweets bei der Streaming API. Eigene Darstellung.

Die Datenerfassung könnte also über die Extraktion der eingebetteten Meta-Daten des neuesten Retweets erfolgen. Eine weitere Möglichkeit besteht in der post-hoc Erfassung aller Tweets, wie es die REST APIs erlauben. Da die Daten bei Abruf bereits ein gewisses Alter haben, beinhaltet das nachträglich erhobene Datenset bereits Informationen über Retweets und Favorites bis zum Zeitpunkt der Abfrage. Mit der Funktionsweise sowie den Vor- und Nachteilen der REST APIs befasst sich nun das folgende Kapitel.

4.1.2 REST APIs

Während die Streaming API Echtzeitdaten liefert, übermitteln die REST APIs nur vergangenheitsbezogene Daten. *REST* steht für *Representational State Transfer* und ist ein in der digitalen Welt sehr weit verbreitetes Schema der Datenarchitektur und -weitergabe. Es kategorisiert An- und Abfragen in die Operatoren GET, POST, PUT und DELETE. Die REST Schnittstellen bestehen aus einem Bündel an Methoden zur Daten-Interaktion, die sich erheblich von denen der Streaming API unterscheiden. So stehen momentan 35 Methoden für jeweils spezifische Abfragen zur Verfügung: Von den momentan trendigen Themen (*Trending Topics*) bis hin zu den Tweets, Retweets, Blocks, Followern und Favorites eines Nutzers. Über Kombination von Daten mehrerer Abfragen ließen sich zum Beispiel User-Netzwerke oder das Verhalten spezifischer Nutzer/-innen visualisieren. Zudem besteht die Möglichkeit, nicht nur Daten zu lesen (GET-Parameter), sondern auch direkt über die Schnittstelle Tweets zu posten oder Nutzern zu folgen (POST-Parameter) sowie Tweets und Nutzer zu löschen (DELETE).

Für das Sammeln von Tweets eignet sich vor allem die *Search API*, die Teil der REST API ist. Diese funktioniert wie eine Suchmaske und erlaubt Suchanfragen wie: `love OR hate from:mustermann until:2015-01-01` (Suche nach Tweets des Nutzers „mustermann“, die „love“ oder „hate“ beinhalten und vor dem 01.01.2015 verfasst wurden).

Twitter erlaubt in diesem Kontext eine Sortierung der Suchergebnisse: Der Parameter `result_type` ermöglicht eine absteigende Sortierung nach Datum (`recent`), eine Ausgabe absteigend nach Popularität des Tweets (`popular`) oder eine gemischte Ausgabe (`mixed`), welche standardmäßig eingestellt ist (Twitter, Inc., 2015c). Die Sortierung nach `recent` ist für die Datensammlung am praktikabelsten, um eine Vollständigkeit der Daten zu erzielen.

Die REST APIs stellen Daten nicht wie bei der Streaming API über die Push-Methode zur Verfügung, sondern übermitteln diese erst nach einzelnen Pull-Abfragen. Diese Anfragen unterliegen einer starken Reglementierung seitens Twitter. Je nach Abfrage-Methode gibt es unterschiedliche Einschränkungen: Jeder GET oder POST Befehl stellt einen *Request* dar. Sollen beispielsweise Tweets mit einem oder mehreren definierten Keywords abgerufen werden, liefert die API maximal 100 Tweets je Abfrage bei einem Limit von 180 Abfragen je 15 Minuten-Intervall, was ein stündliches Maximum von 72.000 gesammelten Tweets ergibt. Es besteht die Möglichkeit, dieses Limit auf 450 Requests je 15 Minuten zu er-

weitern, wenn die registrierte Twitter-Anwendung nur durch einen Nutzer verwendet wird[14]. Dies ergibt einen Höchstwert von 180.000 Tweets pro Stunde. Ähnlich restriktiv wird auch die Anzahl möglicher Abfrage-Parameter gehandhabt: Die API erlaubt pro Request 20 Keywords (empfohlen wird ein Wert von maximal 10) sowie 100 User-IDs (Twitter, Inc., 2015g). Für die nutzerspezifische Tweet-Suche (Suche über die Nutzer-ID) gilt die Obergrenze von 3200 Tweets. Im Anhang A befindet sich eine Auflistung aller Einschränkungen.

Dieses Beispiel verdeutlicht, dass die Abfrage von Tweets stark eingeschränkt ist. Wächst bei der Streaming API die Obergrenze an erfassbaren Daten je Sekunde noch mit dem gesamten Twitter-Volumen (maximal ein Prozent des Gesamtaufkommens), ist das Limit bei der REST API fixiert. Mehr als 180.000 relevante Tweets können in einer Stunde von einer App nicht gesammelt werden, auch wenn der Anteil am Gesamtvolumen womöglich deutlich geringer als ein Prozent war. Finden sich mehrere Millionen Tweets zu einem Suchterm und sollen diese gesammelt werden, dauert dies unter Umständen mehrere Tage. In diesem Kontext spielt der Verfügbarkeits-Zeitraum von Tweets eine besondere Rolle. Der Zeithorizont für die nachträgliche Suche über die REST APIs liegt momentan bei etwa 6-9 Tagen (Twitter, Inc., 2015g). Meistens besteht keine Möglichkeit, Tweets über diese Periode hinaus abzurufen[15]. Wäre das potentielle Datenset, das durch einen Suchterm angesprochen wird, sehr groß (z.B. bei einem medialen Großereignis wie dem Fußball WM-Finale der Männer 2014 mit über 32 Millionen Tweets), würde die Erfassung über die Search API mehrere Tage dauern. Dies könnte dazu führen, dass das relevante Datenset mit fortlaufender Zeit zum Teil aus dem verfügbaren Zeithorizont rückt und somit nicht mehr erfasst werden kann.

Abbildung 8 veranschaulicht die Problematik in einer schematischen Darstellung. Die Search API sucht in diesem Fall inkrementell nach historischen Tweets zum WM-Finale. Nach jedem Request (zur Erinnerung: ein Request gibt maximal 100 Tweets aus), sucht das Skript nach jeweils vorher veröffentlichten Tweets. Die Suche läuft folglich retrospektiv ab – jeder Request erfasst 100 Tweets, die vor dem vorigen Request datiert sind. Da pro Tag theoretisch maximal 4,32 Millionen Tweets über die Search API gesammelt werden können, würde es über eine Woche dauern, bis alle 32 Millionen relevanten Tweets zum WM-Finale gespei-

[14] Da in dieser Arbeit die wissenschaftliche Verwendung der Twitter API betrachtet wird und kein Online-Dienst für mehrere Nutzer, wird davon ausgegangen, dass nur ein Anwender je Twitter-Account/App die Schnittstelle nutzt (App Auth).

[15] Die Begrenzung gilt jedoch nicht für die Tweet-Abfrage auf Nutzer-Ebene über die Timeline (GET statuses/user-timeline). Twitter übermittelt hier maximal die letzten 3200 Tweets eines Nutzers – unabhängig des Veröffentlichungszeitpunktes (Twitter, Inc., 2015d).

chert wären. Da sich der erfassbare Zeithorizont konstant mit der Zeit bewegt (immer die letzten sechs bis neun Tage zum momentanen Abfragezeitpunkt; in der Grafik durch den Zeitraum von t=0 bis t=6 dargestellt), wandern die ersten/frühen Tweets des relevanten Datensets bereits nach einem Tag aus dem erfassbaren Zeitfenster. Einen Tag später (Zeitpunkt t=7 im unteren Bereich der Grafik) könnten zwar wieder bis zu 4,32 Millionen Tweets der letzten sechs bis neun Tage abgefragt werden, jedoch ist der Anfang des relevanten Datensets zu diesem Zeitpunkt bereits außerhalb des verfügbaren Zeitfensters.

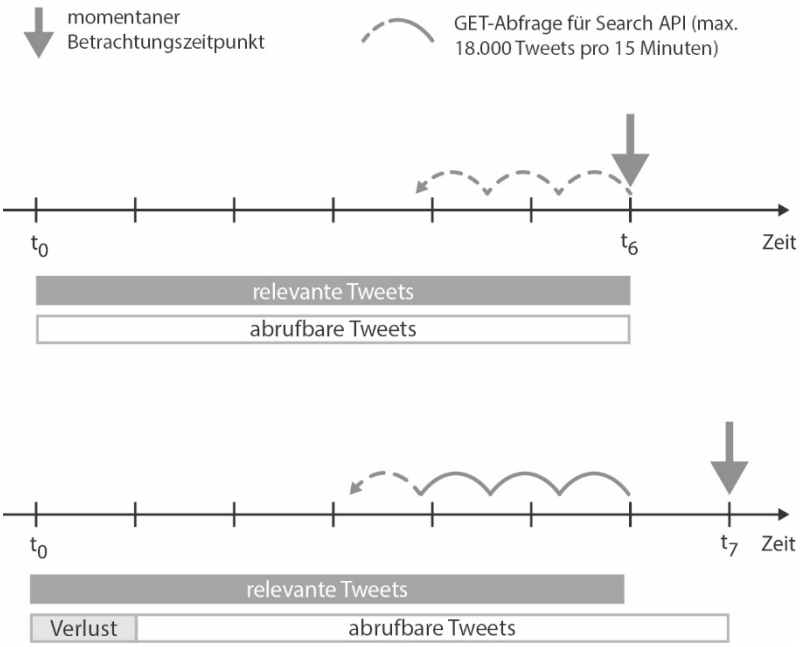

Abbildung 8: Problem der zeitlichen Verfügbarkeit historischer Tweets über die REST APIs. Eigene schematische Darstellung.

Um die negativen Folgen einiger Einschränkungen zu mindern, ermöglicht Twitter bei der REST API eine Eingrenzung der Suchergebnisse über zahlreiche Parameter. Das in Abbildung 8 dargestellte Problem des dynamischen Zeitfensters für

Abfragen ließe sich theoretisch über die zwei Zeitparameter `until` und `since` umgehen: Man startet die Suche am frühestmöglichen Zeitpunkt und sammelt zunächst die Tweets, die als erstes aus dem relevanten und verfügbaren Datenset fallen würden. Die Suche verläuft in diesem Fall entlang des Zeitverlaufs vom ältesten bis zum neuesten relevanten Tweet. Allerdings erlauben diese Parameter nur Datumsangaben und keine Verfeinerung der Zeitgrenzen auf Uhrzeiten. Eine erste Abfrage beliefe sich folglich auf einen ganzen Tag und würde schrittweise alle nachfolgenden Tage erfassen. Dies führt nicht nur zu Problemen mit der strengen Limitierung von Abfragen, sondern ist auch zu ungenau für die Eingrenzung von Suchabfragen.

Relevant für die gängigen Abfrage-Methoden sind deshalb vor allem: `since_id` und `max_id` (Tweet-ID, ab/bis zu der Daten ausgegeben werden sollen). Mithilfe dieser beiden Konstanten sowie `Count` (Höchstzahl an Ergebnissen je Abfrage)[16], `until` (Stichtag, bis zu diesem Ergebnisse angezeigt werden sollen) und `lang` (Nutzersprache) können beispielsweise Tweets zu einem Hashtag, gestaffelt nach Tweet-ID, abgerufen werden, wodurch sich doppelte Abfrageergebnisse vermeiden lassen und die begrenzten Abfragen ökonomisch sinnvoll genutzt werden können. Jedoch löst auch dieser Ansatz nicht die Problematik des dynamischen Verfügbarkeitshorizonts bei großen Datensets. Sinnvoll wäre also eine Kombination beider Ansätze: Mehrere Skripte mit unterschiedlicher Account- beziehungsweise App-Autorisierung sammeln simultan Tweets. Jeder Account mit eigener App greift dann über die REST APIs Daten eines bestimmten Zeitraums ab (definiert durch `since` und `until`). Durch die gestückelte Datenabfrage reduziert sich dann die Dauer der Datenerhebung.

4.1.2.1 Anwendungsbeispiel: Erheben historischer Tweets

Eine bereits genannte Einschränkung der Streaming API ist der Zeithorizont erfassbarer Tweets: Es können nur Daten ab dem gegenwärtigen Zeitpunkt gesammelt werden. Die Erhebung historischer Tweets kann nur über die REST APIs, beziehungsweise deren integrierte Search API erfolgen. Daneben stehen 35 weitere Methoden zur Verfügung, die jeweils spezifische Daten, wie Followers und Favorites von definierten Nutzern, zur Verfügung stellen. Die hinsichtlich der Datenbreite umfassendste und damit wohl auch meistgenutzte Methode ist jedoch die Nutzung der Search API.

[16] Der benutzerdefinierte Wert von Count muss unterhalb der von Twitter zugelassenen Zahl an Ergebnissen liegen. Dieses Limit unterscheidet sich nach Abfrage-Methode, liegt jedoch meist bei 100.

Listing 6: Einfache Suchabfrage nach Tweets mit „apple“ über die Search API

```
import tweepy

# Authentifizierung
consumer_key="<Consumer/API Key>"
consumer_secret="<Consumer/API Secret>"
access_key="<Access Token>"
access_secret="<Access Token Secret>"
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_key, access_secret)
api = tweepy.API(auth)

#Definiere Suchbegriff
term = "apple"

#Definiere relevante Parameter der Abfrage
for tweet in tweepy.Cursor(api.search,
                           q=term,
                           count=100,
                           lang="de",
                           result_type="recent",
                           include_entities=True).items():
    #Gebe nur Tweet-Zeitpunkt und Text aus
    print tweet.created_at, tweet.text
```

Das Beispiel Listing 6 zeigt eine typische Suche von Tweets, die bis zum gegenwärtigen Zeitpunkt erstellt wurden und ein vorher definiertes Schlagwort enthalten. Die Autorisierung entspricht dem Vorgehen bei der Streaming API. Nach der Definition des Suchterms werden weitere relevante Parameter der Abfrage bestimmt. Zum einen wird die Cursor-Methode verwendet, die die Ergebnisse jedes Requests in ein Cursor-Objekt bündelt. Ein Cursor ist ähnlich wie eine Seite und dient der Aufteilung der gesammelten Daten in Datenblöcke, um diese besser sichten und verarbeiten zu können. Als Ausgabe-Obergrenze für die Suchabfrage wird mit `count` ein Wert von 100 Tweets festgelegt. Dies entspricht dem allgemeinen Limit der REST API für Suchanfragen (siehe Anhang A). Zudem wird das Abfrageergebnis auf deutsche Tweets[17] eingeschränkt und absteigend nach Tweet-Zeitpunkt sortiert. Die Suche integriert auch die Tweet-Entities (also #hashtags, @mentions, URLS etc.), sodass die API auch Tweets ausgibt, die den Begriff nicht

[17] Twitter analysiert alle Tweets und erkennt automatisch die geschriebene Sprache. Jedoch ist dieser Spracherkennungs-Mechanismus nicht vollkommen zuverlässig. Siehe hierzu auch Kapitel 4.3.1.

im Tweet-Text, aber in einer URL[18] enthalten. Zur Vereinfachung der Darstellung wird der Output schließlich auf den Tweet-Zeitpunkt und die Nachricht beschränkt.

Listing 7: Shell-Output für Programmcode aus Listing 6

```
>>>
2015-04-13 13:12:27 RT @onlinekosten: Apple Watch: Fast eine
Million Vorbestellungen in den USA. Welche Modelle sind
besonders gefragt? http://t.co/Hw5eecA6x5 …
2015-04-13 13:12:11 New Free iPad app - Apotheke im Kaufland
Lörrach - http://t.co/GHc5bR52uh
2015-04-13 13:12:10 #itunes #iphone5 PhotoStitcher - Maxim
Gapchenko http://t.co/QEjDVJMYu3 #apps #apple
2015-04-13 13:11:31 @timohetzel das Video ist komplett
nervig, ansonsten ist der Stick aber überraschend gut.
Dagegen sieht der Apple TV (bisher) alt aus.
2015-04-13 13:11:23 RT @DJUNDERGROUND: #NowPlaying "Player
Party-IceBerg x @JTMoneyMIATL x @YoungTrizo" on @AL-
LOUTHUSTLE.COM Listen http://t.co/l0GmimxmzJ #Clu…
2015-04-13 13:10:51 @CHIP_online #Apple hats vorgemacht
```

Listing 7 stellt beispielhaft das Ergebnis einer solchen Abfrage dar. Dabei zeigt sich, dass der Suchterm nicht zwangsläufig im Tweet-Text vorkommen muss: Der zweite und fünfte Tweet enthalten nur in den Meta-Daten den Begriff „apple“ (hier im Link, der durch Twitter automatisch zu einem *t.co*-Shortlink umgewandelt wird) und werden deshalb von der Search API als relevantes Ergebnis angesehen. Bei der Formulierung des Suchterms ist außerdem zu beachten, dass sich nicht nur der Mechanismus, sondern auch die Systematik der Suchterm-Formulierung von der Streaming API unterscheidet. Eine durch Komma getrennte Liste mehrerer Begriffe ist nicht möglich, der logische Operator OR wird direkt zwischen zwei Suchbegriffe geschrieben. Tabelle 4 stellt die weitere Systematik dar.

[18] Twitter kürzt alle URLs und wandelt diese in t.co-Links um. So wird aus www.link.de/ziel etwas wie t.co/abcde123456. Da Twitter die ursprüngliche URL in den Meta-Daten einbettet, können diese Teile der URL als Suchergebnis dienen.

Tabelle 4: Such-Operatoren der Search API. In Anlehnung an Twitter, Inc. (2015h).

SUCHPARAMETER	BERÜCKSICHTIGTE BEGRIFFE/TWEETS
apple tv	Tweets mit beiden Begriffen (Position egal)
“apple tv”	Tweets mit der genauen Wortfolge
apple or tv	Tweets mit einem dieser Begriffe
apple -tv	Tweets mit *apple*, aber ohne *tv*
#apple	Tweets mit Hashtag *apple*
from:userx	Tweets des Users *userx*
to:userx	Tweets an User *userx* (Replies oder Mentions)
@userx	Tweets mit Mention von *userx*
apple or tv since:2015-07-01	Alle Tweets mit Begriffen *apple* oder *tv* seit dem 01.07.2015
apple until:2015-07-01	Alle Tweets mit Begriff *apple* bis zum 01.07.2015
apple :)	Alle Tweets mit Begriff *apple* und positiver Stimmung
tv?	Alle Tweets mit Begriff *tv* und gestellter Frage
apple filter:links	Alle Tweets mit Begriff *apple* und einer enthaltenen Verlinkung

Die gezeigte Abfrage über die Search API ist zum Beispiel als Vorbereitung für eine umfassende und langfristige Erhebung mittels der Streaming API, bei der zunächst der aktuelle Datenbestand gesichtet werden soll, oder für eine rein historische Datensammlung denkbar. Möglich wäre aber auch eine ergänzende Abfrage von Tweets, die aufgrund eines Überschreitens der Bandbreiten-Höchstgrenze nicht mehr erfasst werden konnten. Dies ist wiederum mit Hilfe ergänzender Such-operatoren möglich. Hierfür stehen die Parameter `since_id` und `max_id` zur Verfügung, die die Such-Ergebnisse mit einer unteren und oberen Tweet-ID-Grenze einschränken können. Da Tweet-IDs fortlaufend nummeriert werden, können – bei Kenntnis einer bestimmten Tweet-ID – alle Tweets bis oder ab diesem Wert angefordert werden.

Es gilt jedoch zu beachten, dass die REST APIs die Anzahl von Abfragen in einem bestimmten Zeitintervall restriktiv behandeln (siehe oben). Diese Problematik sollte die Programmierung automatisierter Abfragen berücksichtigen, indem zum einen die Suchanfragen in Sequenzen unterteilt werden und zum anderen die Zahl der bereits getätigten Requests und respektive die Zahl der noch möglichen Abfragen kontrolliert wird. Tweepy ermöglicht sowohl eine Eingrenzung der Abfragen mithilfe der Parameter `since_id` und `max_id`, als auch eine Überwachung der Abfrage-Limitierung.

In Listing 8 wird der Code um die entsprechenden Funktionen erweitert. Das Programm soll in diesem Fall nachträglich bis zu eine Million Tweets mit dem Begriff „tatort“ erfassen. Zudem speichert die Suchschleife alle ermittelten Tweets in einer Textdatei. Das Programm sucht in sequentiellen Abfragen retrospektiv nach Tweets, die dem Suchterm entsprechen. Dabei findet auch ein Abgleich nach Duplikaten statt. Der Suchbereich wird anhand automatisch gesetzter Werte für `sinceID` und `maxID` dynamisch begrenzt: die Schranken verschieben sich je Anfrage, wobei in diesem Fall die obere Grenze der ID des jeweils zuletzt gespeicherten Tweets entspricht und die untere Grenze nicht definiert ist. Möglich ist aber auch eine manuelle Festlegung auf bestimmte IDs. Pro 15-Minuten-Intervall tätigt das Skript 450 Requests, die jeweils maximal 100 Tweets sammeln. Diese Werte entsprechen den maximal zugelassenen Interaktionen per AppAuth über die REST API, welche während der gesamten Programmaktivität durch das Programm überwacht werden. Dies gewährleistet, dass die Limits nicht überschritten werden, sondern das Programm bis zum nächsten Zeitfenster pausiert. Die Zahl der gesammelten Tweets je Sequenz wird addiert und mit einer zuvor definierten Obergrenze abgeglichen. Bei Erreichen dieses Limits endet der Prozess.

Somit ist diese Programmschleife zum nachträglichen Sammeln von Tweets geeignet: Nach Initiierung erfolgt der Erfassungs- und Speicherprozess selbstständig, in Abhängigkeit der verfügbaren Requests. Es gilt jedoch zu beachten, dass eine Änderung der Paramater nach Start des Prozesses nicht mehr möglich ist. Bei einer Anpassung des Suchterms oder der oberen/unteren Tweet-ID muss der Prozess neu gestartet werden.

Listing 8: Erweiterte Suchschleife zur Abfrage von Tweets über die REST API

```
import tweepy, time, sys, os, jsonpickle
[...]
api = tweepy.API(auth, wait_on_rate_limit=True,
                 wait_on_rate_limit_notify=False)
if (not api):
 print (time.strftime("%Y%m%d-%H%M%S") +
                      " Autorisierungsfehler")
 sys.exit(-1)

term = "tatort"
exportfile = "tatort.txt"
maxTweets = 1000000
qLimit = 100
sinceID = None # oder sinceID = 012345678
maxID = -1L       # oder maxID = 123456789
tweetCount = 0 # Setze Zähler zu Beginn auf 0
print("Beginne mit dem Sammeln von maximal {0} Tweets mit
      dem Begriff '{1}'".format(maxTweets, term))

# Definiere Schleife
with open(exportfile, "w") as file:
 while tweetCount < maxTweets:
  try:
   if (maxID <= 0):
    if (not sinceID):
      newTweets = api.search(q=term, count=qLimit)
    else:
      newTweets = api.search(q=term, count=qLimit,
                             since_id=sinceID)
   else:
    if (not sinceID):
       newTweets = api.search(q=term, count=qLimit,
                             maxID=str(maxID - 1))
    else:
      newTweets = api.search(q=term, count=qLimit,
                             maxID=str(maxID - 1),
                             since_id=sinceID)
   if not newTweets:
    print("keine weiteren Tweets gefunden")
    break
   for tweet in newTweets:
    file.write(jsonpickle.encode(tweet._json,
               unpicklable=False) +  "\n")
```

```
      tweetCount += len(newTweets)
      print(time.strftime("%Y%m%d-%H%M%S") + " | {0} Tweets
            gedownloaded".format(tweetCount))
      maxID = newTweets[-1].id
   except tweepy.TweepError as error:
    print("Fehler : " + str(error))
    break
print("------------------")
print("{0} Tweets in {1} gespeichert".format(tweetCount,
      exportfile))
```

4.1.2.2 Bewertung der REST APIs

Grundsätzlich sind die REST APIs ein umfassendes Toolkit mit einer Vielzahl an Einsatzmöglichkeiten und damit ähnlich geeignet für das Sammeln von Daten, wie die Streaming API. Allerdings hat die Schnittstelle eine grundlegend andere Funktionsweise als die zuvor betrachtete Streaming API. Es werden nur historische Daten übermittelt und diese nur nach spezifischer Anfrage. In ihrer Funktionsweise ist die Schnittstelle sowohl zur Einarbeitung beziehungsweise Sichtung der Datengrundlage zu Beginn eines Forschungsprojektes, als auch zur Erhebung vollständiger Datensätze ideal. Denkbar ist auch eine Verwendung als zusätzliche Datenquelle, falls das Bandbreiten-Limit der Streaming API überschritten wurde und diese nicht erfassten Tweets nachträglich in den Datensatz aufgenommen werden sollen. Des Weiteren stehen umfassende Abfragemöglichkeiten zur Verfügung: Neben trendigen Themen (*Trending Topics*) oder an einem bestimmten Ort veröffentlichte Tweets auch sehr spezifische Informationen, wie die Friends eines Twitter-Users oder dessen Favorites und Retweets. Dies erlaubt nicht nur das Abbilden des historischen Nutzerverhaltens, sondern auch ganzer Nutzer-Netzwerke.

Andererseits bewirkt die Fülle an Abfragen und Kombinationsmöglichkeiten auch eine höhere Komplexität der Datenverarbeitung. Es stehen insgesamt 36 unterschiedliche Methoden zur Verfügung, die sich sowohl hinsichtlich ihrer Funktion, als auch ihrer Parameter unterscheiden. Um verwertbare Daten zu erhalten, sind oftmals mehrere Abfragen mit unterschiedlichen Methoden notwendig, deren Ergebnisse dann verknüpft werden müssen. Um beispielsweise das Netzwerk eines Nutzers aufzulisten, sind zwei separate Requests notwendig: `GET friends/ids` und `GET followers/ids`, die dann wiederum mit `GET mutes/users/ids` abgeglichen werden müssen. Des Weiteren ist die Anzahl an Requests streng reglementiert. Je nach Methode dürfen innerhalb eines 15-Minuten-Intervalls 15 bis 450 Abfragen gestartet werden, die wiederum einen limitierten

Datenumfang haben. Dadurch wird zwar eine Überlastung der Twitter-Server vermieden, allerdings erhöht die unübersichtliche Struktur an Einschränkungen ebenfalls die Komplexität von Abfragen. Nutzer der REST API müssen immer darauf achten, die jeweiligen Höchstgrenzen einzuhalten, um nicht eine Sperrung der IP-Adresse zu riskieren.

Wie bereits erwähnt, stellt die REST API nur historische Daten zur Verfügung. Dies hat den Vorteil, wichtige Daten ex post abrufen zu können. Da sich Trends beziehungsweise populäre Themen auf Twitter meist sehr schnell formieren und ebenso schnell wieder abebben, ist eine nachträgliche Daten-Erhebung für wissenschaftliche Zwecke sehr praktikabel. Somit muss man nicht innerhalb kürzester Zeit auf Trends reagiert oder einen etwaigen Verlust von Tweets vor Betrachtung eines Themas befürchten. Allerdings ist der Zeitraum für die vergangenheitsbezogene Tweet-Erfassung ebenfalls eingeschränkt: Laut Twitter (2015g) stehen über die Search API nur Tweets zur Verfügung, die nicht älter als 6 bis 9 Tage sind. Somit ist beispielsweise eine nachträgliche Betrachtung von älteren Themen/Ereignissen über diese Schnittstelle nicht möglich. Zudem gilt es zu beachten, dass bei sehr bedeutenden Ereignissen mit mehreren Millionen potentiell relevanten Tweets die Gefahr besteht, aufgrund der sehr eingeschränkten Sammel-Kapazität von 180.000 Tweets pro Stunde nicht rechtzeitig alle Tweets zu sammeln, bevor diese aus dem verfügbaren Zeitfenster verschwinden (siehe weiter oben).

Letztlich sollte auch die Qualität des Datensets kritisch betrachtet werden: Twitter (2015g) gibt an, dass die REST API Daten nicht mit dem Anspruch auf Vollständigkeit übermittelt, sondern die Ergebnisse nach Relevanz auswählt. Die Zahl der verfügbaren historischen Tweets hängt von der Relevanz des Themas im gesamten Tweet-Volumen ab. Bei unbedeutenden Themenkomplexen kann der Zeithorizont auch geringer als die mögliche Zeitspanne von 6 bis 9 Tagen ausfallen. Beziehungsweise kann es durchaus sein, dass nicht alle Daten, die den Filterkriterien entsprechen, zur Verfügung stehen.

Das Konzept der Vollständigkeit der Twitter-Daten verfolgt vor allem der kostenpflichtige Anbieter *Gnip*. Die Möglichkeit, Tweets und deren Metadaten über Drittanbieter zu sammeln oder über Datenhändler zu erwerben, soll im Folgenden skizziert werden.

4.1.3 Drittanbieter

Zusätzlich zu der manuellen Datenerhebung über die Twitter APIs stehen spezialisierte Online-Dienstleister zur Verfügung. Die bekanntesten sind *Gnip*[19] und *DataSift*[20]. Im April 2015 gab jedoch Datasift bekannt, dass das Unternehmen ab August 2015 den Zugriff auf Twitter-Daten verliert (Halstead, 2015). In Zukunft soll Gnip der einzige Anbieter für Twitter-Daten sein, womit sich Twitter die alleinige Kontrolle über die kommerzielle Verwertung seiner Daten schafft (Hofer-Shall, 2015). Auf Basis einer monatlichen Gebühr (mehrere tausend USD) können unter anderem repräsentative Stichproben (z.B. das 10%-Sample *Decahose* von Gnip) oder historische Tweets bis zu einem Stichtag angefordert werden. Das verfügbare Datenset ist somit hinsichtlich der Informationstiefe sehr ausführlich[21], weist jedoch im Normalfall in der Breite nicht den Datenumfang auf, der theoretisch mit gebündelten Abfragen über beide APIs abgerufen werden könnte. Gnip beschränkt sich hier, aufgrund der großen Datenmenge, auf die gängigsten Metadaten (Gaffney & Puschmann, 2014, S. 58). Statt einem Request über einer der APIs müssen hier spezifische Datenanfragen an den Dienstleister gestellt werden, der dann – je nach Datentyp – die relevanten Daten einmalig oder regelmäßig übermittelt. Die Daten sind bereits strukturiert und können beispielweise als Datenbank-Import oder mit vorher extrahierten, relevanten Werten versendet werden.

Neben Gnip steht eine größere Zahl kostenloser oder gebührenpflichtiger Programme oder Dienste zur Verfügung, die das Sammeln von Tweets vereinfachen, jedoch teilweise nur eingeschränkte Funktionen beinhalten. Als beispielhafte Auswahl dieser Dienste stellt diese Arbeit die zwei gängigen Online-Dienste *Twitonomy* und *Tweet Archivist* vor.

Twitonomy (Diginomy Pty Ltd., 2015) erlaubt in seiner kostenlosen Version unter anderem das Sammeln von Tweets, ein Aggregieren von Tweets eines Nutzers sowie grundlegende Analysen über dessen Twitter-Verhalten, Interaktion und Vernetzung. Die kostenpflichtige Variante beinhaltet detailliertere Informationen und eine Funktion zum Speichern beziehungsweise Exportieren von Tweets und Analysen als PDF oder Excel-Sheet. Twitonomy ist sehr einstiegsfreundlich und gibt mit geringem Aufwand viele aufschlussreiche und übersichtlich dargestellte Informationen. Ebenso ist der Export der Daten intuitiv. Dennoch eignet sich der Dienst nur für die Einarbeitung in die Twitter-Forschung, oder – bei geringem Informationsbedarf – für sehr grundlegende Analysen. Die Datenverfügbarkeit ist

[19] *Gnip* wurde 2014 von Twitter übernommen. www.gnip.com
[20] www.datasift.com
[21] Bei *Gnip* sind alle Tweets seit dem Tag der Freischaltung des Online-Dienstes (21.05.2006) gespeichert.

sehr eingeschränkt: Es stehen nur die etwa 3000 neuesten Tweets zu einem Hashtag oder Nutzer zur Verfügung. Zudem übermittelt Twitonomy nur einen sehr rudimentären Datenumfang: Neben Datum, Nutzername und Tweet-Text umfasst ein Datensatz nur den Link zu diesem Tweet, die Client, über den der Tweet geschrieben wurde sowie Angaben über Followers, Retweets und Favorites. Das Datenmaterial ist dabei immer historisch. Twitonomy verwendet also die REST APIs.

Demgegenüber ermöglicht Tweet Archivist sowohl retrospektive Abfragen (bis zu maximal 2000 Tweets) also auch das Einrichten von zukünftigen Abfragen über die Search API (Tweet Archivist, Inc., 2015a). Die Kernfunktion ist das Definieren von Suchtermen, für die der Dienst (stündlich aktualisiert) alle zugehörigen Tweets ausgibt. Die Ausführung der REST API-Abfragen wird folglich auf die Server von Tweet Archivist ausgelagert, welcher die Suchabfragen koordiniert. Zusätzlich besteht auch hier die Möglichkeit, grundlegende Statistiken über Nutzer, Inhalte oder die zeitliche Verteilung abzurufen. Momentan können bei einer monatliche Pauschale bis zu drei Archive, also drei Suchabfragen, eingerichtet werden.

Dennoch eignet sich auch Tweet Archivist nur für die Einarbeitung in die Twitter-Forschung oder für sehr einfache und limitierte Abfragen. Das Unternehmen gibt eine maximale Archivgröße von 50.000 Tweets an, danach erfolgt eine neue Befüllung mit Daten (Tweet Archivist, Inc., 2015b).

Zusammenfassend sind beide vorgestellten Online-Dienste nur eingeschränkt für die wissenschaftliche Nutzung zu empfehlen. Aufgrund des begrenzten Datenumfangs (hinsichtlich Breite und Tiefe) besteht kaum eine Möglichkeit, ausreichend große Datensets zu erzeugen. Zudem fehlt ein Einblick in die Erzeugung des übermittelten Datensets: Die Daten werden aus einer *Black Box* geliefert, ohne dass Einblicke in die Auswahlmethoden oder die Zuverlässigkeit der Datenerfassung (Missings, Latenzen) gewährt werden (Gaffney & Puschmann, 2014). Gerade im Hinblick auf Reliabilität sind Online-Dienste somit für die Erhebung von Daten für wissenschaftliche Analysen ungeeignet, sondern eher für das Einarbeiten in die Thematik. Demgegenüber ermöglicht Gnip einen umfassenden Zugriff auf theoretisch alle verfügbaren Twitter-Daten. Jedoch sind alle Datenanfragen mit sehr hohen Kosten verbunden.

4.1.4 Vergleich der Möglichkeiten zur Datensammlung

Ein Vergleich der vorgestellten Möglichkeiten zur Datenabfrage ist schwierig, da sich alle Varianten in Zeitraum, Umfang und Methode der Erhebung grundsätzlich unterscheiden (siehe Abbildung 9). Die Streaming API übermittelt nach einmaliger Initiierung des Abfrageprozesses zeitlich unbegrenzt Echtzeitdaten, die anhand von Filtern eingegrenzt werden. Dagegen erlauben die REST APIs nur die Erfassung historischer Daten eines begrenzten Zeitraums, wobei diese explizit (und bei höherem Datenumfang mehrfach und sequentiell) angefragt werden müssen. Der Datenanbieter Gnip bietet wiederum Daten von sowohl vergangener als auch zukünftiger Tweets, jedoch in einem eingeschränkten Informationsgrad und vor allem bei einer hohen monatlichen Gebühr. Gerade hinsichtlich des finanziellen Aspektes ist die Variante, Daten über Drittanbieter zu erhalten, wohl für viele Forschungszwecke unattraktiv.

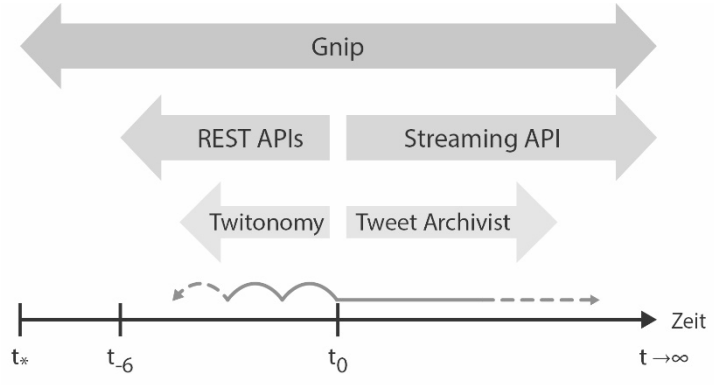

Abbildung 9: Zeithorizont verschiedener Methoden zur Datensammlung. Eigene Darstellung.

Die Streaming API bietet ideale Möglichkeiten zum Sammeln eines möglichst kompletten und zeitlich unbegrenzten Datensatzes relevanter Tweets (oder eines Samples aller Tweets) ab einem definierten Zeitpunkt, wogegen hier die Möglichkeit fehlt, historische Tweets zu erfassen. Sofern die Menge relevanter Tweets nicht ein Prozent des Gesamtvolumens überschreitet, werden alle Daten ohne Einschränkung übermittelt. Sinnvoll gesetzte Abfrage-Filter können das Risiko einer Bandbreitenüberschreitung auf ein akzeptables Niveau senken. Jedoch reichen

auch diese Filter bei sehr populären Themen (wie Katastrophen) unter Umständen nicht aus, um das Suchergebnis ausreichend einzugrenzen. Wie bereits aufgezeigt, besteht aufgrund der Funktionalität der Schnittstelle keine Möglichkeit, Tweets vor dem gegenwärtigen Zeitpunkt zu sammeln – diese Funktion ersetzen die REST APIs.

Die REST APIs beziehungsweise deren Search API eignen sich hervorragend für frühe Phasen eines Forschungsprojektes: Die Abfrage historischer Twitter-Daten ermöglicht einen Überblick der Datenstruktur, Kommunikationsweisen und Häufigkeiten. Die Begrenzung der Datenabfrage hinsichtlich Umfang und Zeitintervall kann mit geringem Programmieraufwand kontrolliert und durch sequentielle Abfragen mit entsprechenden Parametern nahezu umgangen werden (siehe Listing 8). Denkbar und sinnvoll ist auch die nachträgliche Suche von Tweets, die bei Überschreiten des Bandbreiten Limits der Streaming API nicht erfasst wurden. Dies bedarf allerdings einer Kenntnis der Zeitpunkte, zu denen eine Überschreitung stattfand, um über die erste und letzte Tweet-ID dieses Zeitfensters die Suche nach Tweets einzugrenzen. Deshalb empfiehlt sich eine Speicherung der Limits in Logfiles. Ein anschließender Dubletten-Abgleich filtert schließlich bereits vorhandene Tweets. Jedoch ist der Zeitraum für die Datenerfassung über die REST API nur auf etwa eine Woche begrenzt und eine Vollständigkeit des Datensatzes für diese Zeitspanne nicht garantiert. Dementsprechend sollte die Nutzung der REST APIs immer mit Vorsicht erfolgen.

Daher eignen sich beide Twitter-Schnittstellen nicht zur Generierung valider Daten aus der größeren Vergangenheit für Forschungszwecke. Hierfür müssen Informationen vom Drittanbieter Gnip erworben werden. Während bei der Streaming API das Datenvolumen unter Umständen zu groß ist und durch Filter beschränkt werden muss, besteht bei der REST API wiederum die Gefahr, dass nicht (mehr) alle Daten übermittelt werden können. Dieses Problem gibt es bei gekauften Daten nicht: Gnip besitzt einen exklusiven und uneingeschränkten Zugriff auf alle Tweet- und Nutzer-Daten und speichert diese für mehrere Jahre. Dies ermöglicht auch Langzeitstudien oder komparative Analysen der Daten verschiedener Jahre. Die Daten sind jedoch, in Abhängigkeit vom gewünschten Umfang, kostspielig.

Sowohl über die Streaming API als auch über die REST API lassen sich weder valide Vollerhebungen, noch verlässliche Zufallsstichproben generieren. Die Ausgabe der REST APIs basiert, wie bereits angemerkt, auf Relevanz und nicht Vollständigkeit. Selbst wenn über die Streaming API das Limit von einem Prozent nicht überschritten wird, gewährleistet Twitter keine vollständige Übermittlung

aller relevanten Tweets zum Suchterm. Die Betrachtung birgt bei diesen beiden Datenquellen immer ein Risiko.

Bei Daten, die durch die Twitter APIs gesammelt wurden, besteht im Hinblick auf rein quantitative Analysen zudem die Gefahr der Fehlinterpretation von Tweet-Volumina. Aufgrund des unbekannten Gesamtvolumens auf Twitter könnten eigentlich natürliche Veränderungen im Tweet-Volumen überbewertet werden. Gerade bei geringen Betrachtungszeiträumen sind Daten anfällig für tages-, zeit- und nutzerbedingte Schwankungen, beispielsweise durch Wetter, Tag-Nacht-Wechsel, Zeitzonen oder dem individuellen Nutzerverhalten. Eine plötzliche Spitze in der Hashtag-Häufigkeit bedeutet demnach nicht zwangsläufig eine hohe Popularität, sondern kann auch aus einem allgemeinen Anstieg der momentan aktiven Nutzerzahl (aufgrund der Tageszeit) resultieren. Demnach sollte eine rein quantitative Betrachtung von Tweet-Häufigkeiten immer im Kontext der Gesamtzahl aller Tweets zu diesem Zeitpunkt betrachtet werden. Es besteht allerdings nur die Möglichkeit, diese Werte über die eingeschränkt verfügbare, beziehungsweise kostenpflichtige *Firehose* abzurufen (bei Gnip). Eine Einordnung der erfassten Werte ist somit in den meisten Fällen mit kostenlosen Mitteln kaum oder gar nicht möglich. Tabelle 5 (nächste Seite) fasst die jeweiligen Vor- und Nachteile der vorgestellten Quellen nochmals zusammen.

Sollten Tweets ohne zusätzliche Kosten erhoben werden, ist letztlich die Art der zu erhebenden Daten für die Wahl der Datenquelle entscheidend. Hinsichtlich der Datenstruktur sind beide Schnittstellen nahezu identisch. In manchen Fällen ist eine Kombination beider APIs die praktikabelste Methode zur Datenerfassung auf Twitter, um die jeweiligen Beschränkungen zu mindern, beziehungsweise Vorteile bestmöglich zu nutzen. Die REST APIs dienen dabei zur Sichtung, Einarbeitung sowie zur Erhebung vergangener Tweets oder nutzerspezifischer Informationen. Außerdem werden sie, bei Überschreiten des Bandbreitenlimits der Streaming API, zur nachträglichen Erfassung „verloren gegangener“ Tweets eingesetzt. Die Nutzung der Streaming API erfolgt dagegen ab einem definierten Zeitpunkt zur Sammlung aller zukünftigen Tweets. Somit werden die Stärken beider APIs kombiniert: Die mächtigen Suchwerkzeuge der REST API und deren Fähigkeit, zeitgleich eine große Anzahl an Parametern abzufragen, und die umfangreichen Datenströme der Streaming API, die in Echtzeit eine zeitlich unbegrenzte Zahl von Tweets inklusive aller Meta-Daten übermitteln.

Nachdem nun einzelne, gängige Ansätze zur Datensammlung präsentiert sowie die Vor- und Nachteile für die Forschung aufgezeigt wurden, folgt im nächsten Kapitel eine Betrachtung von Möglichkeiten zur Speicherung und Verwaltung der gesammelten Daten.

Tabelle 5: Vergleich der Quellen für Twitter-Daten

	STREAMING API	REST APIs	DRITTANBIETER
EIGENSCHAFTEN	• Echtzeit-Daten • 3 Streams • Daten in JSON-Format	• Historische Daten • 36 Abfrage-Methoden • Daten in JSON-Format	• Echtzeit- und historische Daten (unterschiedlich) • Mehrere Datenformate möglich
VORTEILE	• Kostenlos* • Filtermöglichkeiten • Keine Begrenzung des Zeitintervalls • Verfolgung/Protokollierung von gelöschten Tweets	• Kostenlos • Vielzahl an Abfrage- und Filtermethoden • Nachträgliche Datenabfrage	• Umfassende Daten (hinsichtlich Zeitraum) • Usability • Minimaler Aufwand • Keine Programmierkenntnisse notwendig • Bereits strukturierte und angereicherte Daten
NACHTEILE	• Bandbreiten-Limitierung, die vor allem bei Großereignissen schnell erreicht wird • Keine historischen Daten • Favorites, Retweets schwer zu erheben • Kein Anspruch auf Vollständigkeit • Keine Informationen über Gesamtvolumen für Einordnung • Prozess muss kontinuierlich laufen, jede Unterbrechung bedeutet Datenverlust	• Restriktive Datenausgabe • Kurzer Zeithorizont • Komplexität der Abfragestruktur • Kein Anspruch auf Vollständigkeit • Keine Informationen über Gesamtvolumen auf Twitter für Einordnung	• Kostenpflichtig und kostspielig • Vollständigkeit des Tweet-Sets nicht gewährleistet • Keine Angaben über Reliabilität • *Black Box* – Erhebungsverfahren unbekannt

*) bei Verwendung der standardmäßigen Streaming API (Spritzer)

4.2 Systeme der Datenverwaltung

Dieses Kapitel behandelt die Speicherung und Verwaltung der Daten, die durch die im vorigen Kapitel gezeigten verschiedenen Ansätze gewonnen wurden. Hierfür stehen sehr unterschiedliche Möglichkeiten zur Verfügung. Zuerst soll das Speichern in einfache Text-Dateien betrachtet werden, das bereits im vorangegangenen Kapitel kurz angesprochen wurde. Schließlich befasst sich Kapitel 4.2.2 mit Datenbanksystemen, die ein systematischeres und verlässlicheres Sichern der Daten ermöglichen. Hierbei werden zunächst zwei unterschiedliche Datenbank-Architekturen skizziert, um schließlich beispielhaft den Speicherprozess zum Datenbanksystem *MongoDB* zu zeigen. Abschließend folgt ein Vergleich aller betrachteten Alternativen.

4.2.1 Speicherung in Textdateien

Die wohl einfachste Methode zum Speichern von Tweets ist die Verwendung von einzelnen Dateien, die die Tweets deserialisiert im JSON-Format oder bereits formatiert, gefiltert und umstrukturiert beinhalten. Für den Export von Twitter-Daten eignen sich TXT- oder JSON-Dateien, die Tweets im ursprünglichen JSON-Format umfassen, oder CSV-Dateien, die bereits selektierte Werte beinhalten. Python bietet die Möglichkeit, Tweets sowohl im JSON-Format abzuspeichern, als auch umstrukturiert im CSV-Format. Während die benötigten Programmteile zum Lesen der Daten bereits in Python integriert sind, muss jedoch ein zusätzlicher Konverter zur Ausgabe strukturierter JSON-Daten installiert werden, um Datenströme nahezu direkt in Textdateien zu schreiben. Es müssen lediglich die im Binär-Code vorliegenden JSON-Daten (BSON) aus der Twitter-API in ein strukturiertes JSON-Format deserialisiert werden. Dies erfolgt in der Regel mit einem `json-pickle.encode`-Befehl. Die Codierung erfolgt schrittweise je Datensatz.

Der Auszug in Listing 9 zeigt ein einfaches Verfahren zum Abspeichern von Tweets im JSON-Format. In diesem Beispiel sollten Tweets mit dem Begriff „Tatort“ nachträglich über die REST APIs gesammelt werden. Nach der Definition des Datei-Namens und Formates wurde im Skript das Vorgehen bei neuen Tweets vorgegeben. Jeder neue Tweet wird im JSON-Format als neue Zeile dem Dokument angefügt, sodass jede Zeile einem Tweet entspricht. Jeder Schreibvorgang erhöht den Zähler um 1. Bei Erreichen des definierten Maximalwerts stoppt die Schleife den Speicherprozess.

Listing 9: Speicherung von Tweets in eine Textdatei (Auszug aus Listing 8)

```
[...]
exportfile = "tatort.txt"
maxTweets = 1000000
[...]
with open(exportfile, "w") as file:
  while tweetCount < maxTweets:
    try:
    [...]
     for tweet in newTweets:
       file.write(jsonpickle.encode(tweet._json,
                  unpickable=False) + "\n")
       tweetCount += len(newTweets)
[...]
```

Die Begrenzung der Tweetsammlung ist in mehreren Hinsichten sinnvoll: Sie dient indirekt zur Kontrolle von Umfang und zeitlicher Relevanz, sodass beispielsweise nur die letzten/neuesten 1.000 Tweets gesammelt werden. Letztlich steuert eine Begrenzung auch die Dateigröße, was sich wiederum auf die erforderliche Rechenleistung auswirkt. Größere Dateien mit mehreren Gigabyte erfordern für die Analyse und Verarbeitung der gesammelten Daten eine bessere Systemausstattung. Zudem gibt es je nach Dateisystem unterschiedliche Anforderungen hinsichtlich der maximal erlaubten Dateigröße: *FAT32*, ein gängiger Standard von USB-Sticks, erlaubt beispielsweise nur Dateigrößen bis etwa 4 Gigabyte. Sollte der Datenumfang sehr groß und nur schwer kalkulierbar sein, weil Tweets über einen längeren Zeitraum mit Hilfe der Streaming API gesammelt werden, besteht die Möglichkeit, den Datensatz auf mehrere Dateien zu verteilen. Dies soll im folgenden Beispiel näher erläutert werden.

4.2.1.1 Anwendungsbeispiel: Speichern von Tweets in JSON- und CSV-Dateien

Fällt bei der Wahl der Speichermethode von Twitter-Daten die Entscheidung auf Textdateien, ist es in vielen Fällen (wie bereits oben erwähnt) sinnvoll, die Daten auf mehrere Dateien zu verteilen. Auch hierfür kann ein Zähler eingesetzt werden, um die Aufteilung der Daten zu steuern. Listing 10 (nächste Seite) zeigt den Programmauszug eines Skripts zum sequentiellen Speichern in einzelnen JSON-Dateien. Dazu wurde die Klasse `tweepylistener` aus Listing 5, die der Erhebung von Tweets mit dem Inhalt „obama“ diente, um einige Parameter erweitert. Die

Datei-Benennung erfolgt in diesem Anwendungsfall mit dem Zeitstempel der jeweiligen Dateierstellung, was ein Suchen beziehungsweise Eingrenzen von Tweets auf bestimmte Dateien anhand der Zeitangabe ermöglicht. Zusätzlich wird ein Datei-Präfix eingefügt, das in diesem Fall dem Suchterm entspricht. Jeder neue Tweet erhöht den Zählerstand `counter`. Nach einem vorher bestimmten, maximalen Zählerstand (hier 1000) legt das Skript alle weiteren Tweets in einer neuen Datei ab, bis wiederum das definierte Limit erreicht ist.

Neben dem direkten JSON-Export erlaubt Python auch eine Konvertierung in das CSV-Format. Hier besteht die Möglichkeit, bereits vorher relevante Datenfelder auszuwählen und diese dann je Tweet zeilenweise zu speichern. Dies ist von Vorteil, wenn zum einen keine Möglichkeiten zur nachträglichen Verarbeitung von JSON-Daten bestehen oder wenn zum anderen bereits vor Erhebung sicher ist, dass nur bestimmte Datenfelder benötigt werden. Eine Beschränkung auf wesentliche Daten verringert den Datenumfang deutlich, da einige Entities Informationen beinhalten, die für Auswertungen unerheblich sind, wie beispielsweise grafische Angaben zum Profil.

Listing 10: Erweiterter Prozess zum Speichern in mehreren Textdateien

```
import tweepy, time, sys, json
[...]
class tweepylistener(tweepy.StreamListener):
 def __init__(self, api = None, prefix = "obama"):
  self.api = api or API()
  self.counter = 0
  self.prefix = prefix
  # Definiere Zieldatei für Tweet-Daten
 self.output  = open("<Datei-Pfad>" + prefix + "." +
                     time.strftime("%Y%m%d-%H%M%S") +
                     ".json", "w")
  # Definiere Zieldatei für Daten gelöschter Tweets
  self.deleted  = open("geloescht.txt", "a")
  [...]

  # Bestimme Vorgehen in unterschiedlichen Situationen
  # Fall 1: neuer Status-Tweet
  def on_status(self, status):
   try:
       self.output.write(status + "\n")
       self.counter += 1
```

```
    if self.counter >= 1000:
      self.output.close()
      self.output = open("<Datei-Pfad>" + self.prefix +
                         "." + time.strftime("%Y%m%d- \
                         %H%M%S") + ".json", "w")
      self.counter = 0
    sys.stdout.write(time.strftime("%Y%m%d-%H%M%S") +
                     ">> Neue Datei erstellt \n")
      return
[...]
```

Ein Export *aller* Felder im CSV-Format ist allerdings aufgrund des leistungs- und zeitintensiven Prozesses zur Extrahierung, Decodierung und Restrukturierung der Daten nicht sinnvoll. Zwar ist das CSV-Format mit seiner Tabellen-Struktur deutlich kompakter als das JSON-Format, bei dem jedes Daten-Feld einzeln definiert werden muss. Jedoch ist ein Zusammenfügen von Datensätzen im JSON-Format wesentlich komfortabler und sicherer. Bei CSV muss die Reihenfolge und Vollständigkeit der Felder unbedingt eingehalten werden, da sonst die Zuordnung der Werte nicht mehr stimmt. Fehlende Werte erhalten einen Leereintrag in einer Zelle, um die feste Struktur der Daten zu erhalten. Dagegen sind JSON-Daten unsortiert und können in ihrem Umfang unvollständig sein: Nur vorhandene Werte werden gespeichert, die Reihenfolge ist unerheblich. Dadurch können JSON-codierte Daten einfach aneinandergefügt werden, während bei CSVs zuerst die Reihenfolge der Daten und Konsistenz der Struktur überprüft werden muss.

Das Skript in Listing 11 extrahiert zur Veranschaulichung einzelne Tweet-Entities aus Streaming-Daten schreibt diese in eine CSV-Datei. Hierfür wurde der Code aus Listing 10 abgeändert und erweitert. Zuerst wird eine CSV-Datei erstellt und die Spaltennamen definiert. Anschließend wird das Vorgehen bei einem neuen Tweet vorgegeben: Ein *Writer* soll die Daten im CSV-Format in die vorher angegebene Output-Datei schreiben und einzelne Werte mit einem Semikolon trennen. Zudem definiert der Parameter `dialect="excel"` eine Formatierung, die ein späteres Öffnen und Interpretieren durch Excel erlaubt. Mit dieser Option legt das Skript alle für die Dateistruktur relevanten Parameter automatisch fest. Schließlich erfolgt die Auswahl einzelner Entities, die aus dem JSON-Code extrahiert werden. Der Tweet-Inhalt wird zudem in das weit verbreitete Format UTF-8 kodiert, welches sich zur Darstellung sprachspezifischer Zeichen (z.B. chinesischer Zeichen oder Umlaute) eignet. Schließlich hängt das Skript die Daten zeilenweise an das CSV-Dokument an. Ein Zähler erzeugt nach jeweils 1.000 Dokument-Zeilen eine neue Datei.

Listing 11: Export gesammelter Tweets in Semikolon-getrennte CSV-Dateien

```
[...]
class tweepylistener(tweepy.StreamListener):

  def __init__(self, api = None, prefix = "apple"):
    [...]
    # Definiere Zieldatei für Tweet-Daten
    self.output  = open("<Datei-Pfad>" + prefix + "." +
                        time.strftime("%Y%m%d-%H%M%S") +
                        ".csv", "w")
    # Schreibe Spaltennamen für CSV-Datei
    self.output.write("Tweet;Zeit;Txt;User;UID;Name"+"\n")

    def on_data(self, data):
    [...]
     def on_status(self, status):
     writer = csv.writer(self.output, delimiter=";",
                         dialect="excel")
     # Definiere einzelne Tweet-Entities
     t_id = json.loads(status)["id"]
     t_tme = json.loads(status)["created_at"]
     t_txt = json.loads(status)["text"].encode("utf-8")
     t_uid = json.loads(status)["user"]["id"]
     t_usr = json.loads(status)["user"]["screen_name"]
     # Fülle Zeile mit den Werten
     writer.writerow([t_id, t_tme, t_txt, t_uid, t_usr])
     self.counter += 1
     if self.counter >= 1000:
       self.output.close()
       self.output = open("<Datei-Pfad>" + self.prefix + "." +
                          time.strftime("%Y%m%d-%H%M%S") +
                          ".csv", "w")
       self.counter = 0
       sys.stdout.write(time.strftime("%Y%m%d-%H%M%S") +
                        ">> Neue Datei erstellt" + "\n")
       return
[...]
```

4.2.1.2 Bewertung der Speicherung in Text-Dateien

Zusammenfassend ist der Speicheransatz über Textdateien ein geeignetes Mittel, um schnell und ohne größeren Aufwand eine kleine bis mittlere Datenmenge zu speichern. Da Python bereits eine JSON- und CSV-Funktionalität integriert hat, sind keine zusätzlichen Installationen notwendig. Der eigentliche Speicher-Prozess ist kurz und erlaubt eine direkte Einflussnahme auf die Verarbeitung der Daten bis zum Schreibbefehl: So können einzelne Datenfragmente extrahiert oder umstrukturiert werden. Die Methode ermöglicht zudem die Wahl eines gewünschten Endformats: Der Export in das gängige CSV- oder TXT-Format erlaubt hier nicht nur ein direktes Bearbeiten, Suchen, Extrahieren oder Einfügen von Datensätzen in der Datei, sondern auch eine schnelle Weitergabe von Daten an andere Personen. Zusätzlich kann die Dateigröße mit der Festlegung der Größe von Datensätzen kontrolliert werden. So können auch Dateien in kleinerem Umfang erzeugt werden, um sie zum Beispiel per E-Mail oder auf einem USB-Stick weiterzugeben. Die einfache und offene Dateistruktur mit einer oder mehreren Dateien im CSV-, JSON- oder TXT-Format erlaubt das direkte Lesen und Bearbeiten einzelner Datensätze. Aufgrund der klaren Struktur (jede Textzeile enthält alle Informationen zu einem Tweet) erhält man schnell einen guten Überblick über den Datenbestand. Letztlich ermöglichen fertige Textdateien auch einen schnellen Import in Statistik-Programme, wie *SPSS* oder *Stata*, die Informationen direkt aus CSV- oder JSON-Dateien auslesen können.

Jedoch eignet sich diese Methode nicht für Erhebungen mit sehr großem Datenumfang und/oder langer Dauer. Hier würden, je nach Wahl des `counter`-Parameters, wenige sehr große oder viele kleine Dateien entstehen. Zu große Dateien können unter Umständen nicht von allen Programmen eingelesen werden: So verarbeitet *Microsoft Excel* nur CSVs mit maximal etwa 65.000 Zeilen, *Microsoft Access* nur Daten bis zu 2 Gigabyte. Sehr große Text-Dateien mit mehreren Gigabyte an Informationen benötigen zudem leistungsfähige Rechner mit ausreichend großem Arbeitsspeicher (optimal sind 16 GB und mehr). Zu kleine Dateien führen mit zunehmender Dateizahl wiederum zu einer unübersichtlichen Dateilandschaft. Zudem stellt sich die Frage, wie die in den Dateien enthaltenen Informationen verknüpft und verarbeitet werden sollen: Einzelne Auswertungen je Datei oder aggregierte Auswertungen über alle Dateien? Für umfassende Analysen müssten alle Einzeldateien zunächst nachträglich zusammengefügt werden.

Ein großer Nachteil von Textdateien ist die mangelnde Zugriffsmöglichkeit auf die Daten. Während der Erhebung ist die momentan beschriebene Datei für Änderungen gesperrt, sodass beispielsweise nicht die ID des zuletzt hinzugefügten Tweets ausgelesen oder die Zuverlässigkeit des Programms überprüft werden

kann. Nach Abschluss des Schreibprozesses besteht keine Möglichkeit für einen simultanen Zugriff auf die Daten durch mehrere Personen. Zudem kann die zugreifende Person nur einen Prozess (Suchen, Kopieren, Überschreiben etc.) gleichzeitig durchführen. Funktionen einer komplexeren Datenverwaltung, wie ein Umstrukturieren, Aggregieren oder Filtern von Daten ist, wenn überhaupt, nur eingeschränkt möglich. Hierfür eignen sich besonders Datenbanksysteme, die das nächste Kapitel betrachtet.

4.2.2 *Datenbank-Systeme*

Für die Speicherung von Tweets stehen grundsätzlich alle Arten von Datenbanken zur Verfügung. Diese untergliedern sich in mehrere Datenbanksysteme, wie unter anderem in hierarchische, relationale und NoSQL-Datenbanken, und unterscheiden sich teils grundlegend in ihrer Datenstruktur (siehe Abbildung 10).

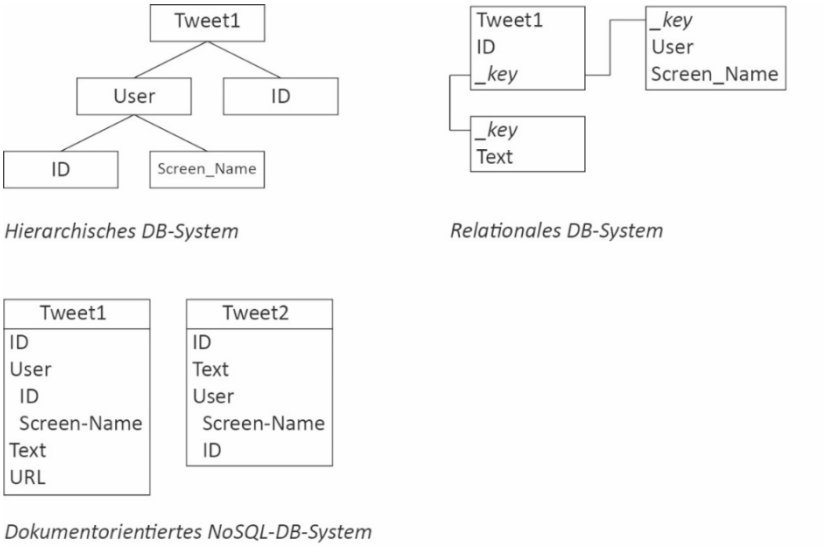

Abbildung 10: Vergleich mehrerer Datenbank-Systeme. Eigene Darstellung.

Bei hierarchischen DB-Systemen, der ältesten Datenbank-Architektur, sind alle Datensätze (*Records*) über eine Baumstruktur mit über- und untergeordneten Werten verbunden. Die Verknüpfung von Daten basiert auf Eltern-Kind-Beziehungen. Bei den sehr weit verbreiteten relationalen Datenbanksystemen (RDBMS), die häufig mit dem Begriff *SQL* zusammengefasst werden, sind Daten in der Regel über mehrere Tabellen verteilt und anhand von Schlüsseln miteinander verknüpft. So besteht ein *Record* unter Umständen aus mehreren Einträgen in verschiedenen Tabellen, wobei ein Eintrag üblicherweise aus einer Tabellenzeile (*Tupel*) mit mehreren Spalten (*Attributen*) besteht. Ein für jeden *Record* einmaliger Schlüssel stellt schließlich die Verbindung zwischen den einzelnen *Tupeln* dar. Vor der Dateneinspeisung in eine SQL-Datenbank muss zunächst jeder Record geparsed werden, um Daten in eine verwertbare Struktur und Formatierung zu überführen (Tugores & Colet, 2013, S. 21). Dieser Schritt wird bei einer größeren Datenmenge zeit- und rechenintensiv, weshalb andere Datenbank-Architekturen, wie *NoSQL*, unter Umständen eine bessere Eignung für das Sammeln großer Datenmengen aus dem Social Web haben.

NoSQL-Systeme („not only SQL“) erweitern die Funktionsweise von RDBMSs um mehrere Datenstruktur-Konzepte, auf denen die Daten verteilt und verknüpft sein können. Die wichtigsten Vertreter davon sind: Key-Value- und Graphen-Datenbanken sowie spaltenorientierte und dokumentorientierte Datenbanken. Key-Value-Datenbanken verwalten Tupel, die aus einem Schlüssel und dem dazu gehörenden Wert bestehen. Graphen-Datenbanken verknüpfen Datensätze über Knoten und Kanten. Spaltenorientierte Datenbanken (auch Wide Column Stores) haben, im Gegensatz zu RDBMS, eine dynamische Spaltenzahl. Es müssen also für einzelne Datenätze nicht alle Spalten definiert sein. Dokumentorientierte Datenbanken bündeln und speichern einzelne Datensätze in Dokumenten. Dokumente bestehen aus einer geordneten Liste von Key-Value-Paaren, wobei der Daten-Umfang dynamisch ist. (Trelle, 2014, S. 3)

Relationale Datenbanksysteme sind aufgrund ihrer Verbreitung und Funktionsfülle ideal für eine Vielzahl an Problemen. Bei der Wahl des Systems für die Sammlung großer Datensätze aus dem Internet sollten jedoch vor allem die jeweiligen Eigenschaften hinsichtlich Datenstruktur und Geschwindigkeit berücksichtigt werden. Durch das Web 2.0 gab es drastische Änderungen bei den Anforderungen an Datenbanksystemen: Sehr große Datenmengen sollen mit einem einfachen Schema bei geringer Datenkonsistenz und kurzen Laufzeiten abgespeichert werden (Trelle, 2014, S. 2). Zudem sollte die Möglichkeit bestehen, Speichersysteme problemlos zu erweitern (Skalierung).

Vergleicht man die oben genannten Architekturen, wird schnell die hervorragende Eignung von dokumentorientierten NoSQL-Systemen für die Sammlung von Twitter-Daten ersichtlich. NoSQL-Systeme erlauben eine horizontale Skalierung – im Lauf des Sammelprozesses können problemlos weitere Server zur Erweiterung des Speicherplatzes hinzugefügt werden (Chodorow, 2013, S. 4). Dies ist hilfreich, da eine Schätzung des erwarteten Datenvolumens (im Vergleich zu Befragungen) sehr schwer ist.

Im Vergleich zu relationalen Datenbanken mit einer Verteilung von Datenbündeln auf mehrere Tabellen bildet in dokumentorientierten Datenbanken ein Dokument genau einen Tweet-Record ab, der die von der Twitter-API übermittelten Key-Value-Paare und Arrays enthält. Somit entfallen Querverweise zu anderen Teilen eines Datensatzes und damit zusätzliche Datenbankabfragen. Dies beschleunigt den Abfrageprozess enorm und minimiert die Systemauslastung. Tugores und Colet (2013) verglichen die Schreibleistung des NoSQL-Systems MongoDB mit MySQL und erkannten bei ersterem eine deutlich höhere Verarbeitungsgeschwindigkeit bei geringerem Leistungsbedarf.

Die Datenstruktur von NoSQL-Systemen ist zudem eine abgewandelte Version des JSON-Formates, welches Twitter bei der Datenübermittlung nutzt. Dokumentorientierte Datenbanksysteme haben eine geringe Konsistenz-Anforderung an Daten: Datensätze können in ihrem Umfang und Aufbau variieren. Demgegenüber müssen bei SQL-Systemen Daten-Records einem vorher definierten Schema entsprechen. Verschachtelte Daten aus Arrays müssen in separate Tabellen geschrieben werden, die wiederum mit einem Schlüssel verknüpft sind. Während bei Document Stores einzelne Records schnell und einfach identifiziert und exportiert werden können, müssen diese bei SQL-Datenbanken zuerst anhand der Verknüpfungen zusammengefügt werden.

Aufgrund der guten Eignung von No-SQL-Systemen für die Speicherung großer Mengen von Internet-Daten verwendet diese Arbeit das dokumentorientierte Datenbank-System *MongoDB*, welches das folgende Kapitel kurz vorstellt.

4.2.2.1 MongoDB

MongoDB wurde 2007 entwickelt und wird seitdem als Open Source System angeboten. Das Datenbank-System besteht in seiner einfachsten Form aus einem Server (Host), der die Daten speichert und verwaltet, sowie einem oder mehreren Clients, die Daten in die Datenbank einspeisen oder abrufen. Das System ist skalierbar, sodass zu einem späteren Zeitpunkt weitere Hosts und Clients hinzugefügt

werden können. Eine Benutzerverwaltung steuert den Zugriff und die Berechtigungen einzelner Nutzer.

MongoDB strukturiert Daten anhand von Objekten, die in Collections gesammelt werden. Eine Collection ist eine Sammlung von Objekten, bei der, im Gegensatz zu SQL-Tabellen, eine Schemafreiheit besteht. Es wird also kein festes Schema mit Namen, Typen und Reihenfolge für einzelne Felder vorgegeben, sondern Daten relativ unstrukturiert gespeichert. Mehrere Collections werden wiederum in einer Datenbank gebündelt. Der Schreibzugriff auf Collections beziehungsweise Datenbanken ist dabei nicht auf einen Nutzer beschränkt, wie das bei Text-Dateien der Fall ist. Dadurch ist es möglich, dass mehrere Prozesse parallel Twitter-Daten sammeln und in verschiedene Collections speichern. Technisch besteht zudem die Möglichkeit, in einem Abfrage-Prozess über eine Weiche spezifische Tweets, z.B. anhand eines Schlagwortes, auf bestimmte Collections zu verteilen. MongoDB beinhaltet außerdem mehrere Funktionen zum Filtern, Sortieren, Umstrukturieren, Importieren und Exportieren sowie grundlegende Möglichkeiten zum Analysieren von Daten.

Vor dem Speichern in MongoDB muss die Datenbank zuerst installiert und eingerichtet werden. Eine Trennung von Datenbank-Server und Client ist dabei sinnvoll[22]. Des Weiteren empfiehlt sich das Anlegen mehrerer Nutzer, die entweder Zugriff auf die Verwaltungsoptionen oder Lese-/Schreibzugriff auf bestimmte Datenbanken haben. Das gewährleistet nicht nur eine allgemeine Sicherheit der Daten vor Fremdzugriff, sondern verhindert zusätzlich, dass Prozesse versehentlich (bei falscher Angaben des Collection-Parameters) Daten in eine falsche Collection schreiben. Zudem ist es ratsam, das Datenbankprogramm als Dienst einzurichten, der bei Bedarf automatisch den Server-Prozess startet.

Sobald der Datenbank-Server eingerichtet ist, kann der Zugriff auf MongoDB von jedem PC erfolgen, der einen MongoDB-Client installiert hat. Grundsätzlich besteht die Möglichkeit, per Kommandozeile die Datenbank zu verwalten, Daten zu lesen, zu sortieren und filtern. Zusätzlich gibt es eine Vielzahl leicht bedienbarer Management-Systeme mit grafischen Benutzeroberflächen. Einige sind dabei bei eingeschränktem Funktionsumfang kostenlos verfügbar, wie beispielsweise

[22] Für eine Gewährleistung von Datensicherheit, Datenschutz und Performance empfiehlt sich immer eine Trennung von Datenbank und Nutzer.

Mongo Management Studio[23] (siehe Abbildung 11), *MongoVUE*[24] oder *3T MongoChef*[25]. Über diese Systeme können gängige Prozesse, wie der Import oder Export, sowie die Suche oder das Filtern von Daten einfach ausgeführt werden.

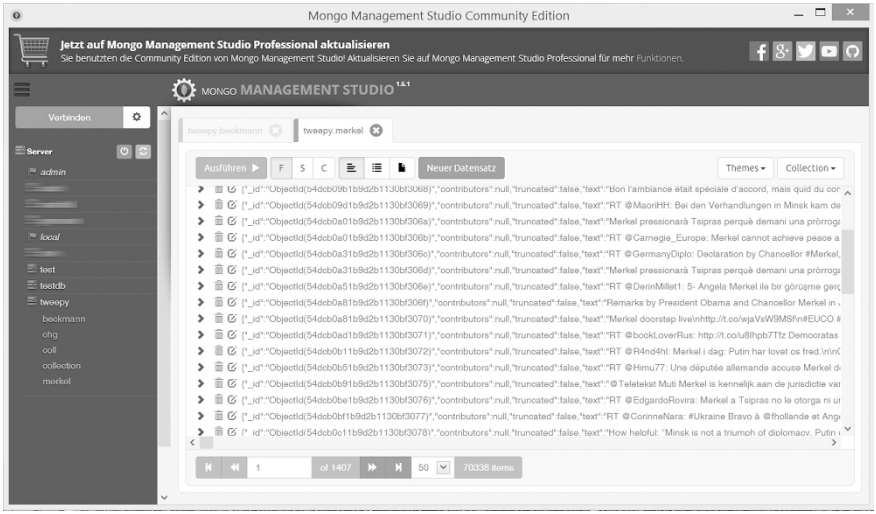

Abbildung 11: Benutzeroberfläche des Mongo Management Studio.

Für eine Verbindung per Kommandozeile muss zuerst im Shell-Fenster der Installationsordner von MongoDB aufgerufen werden, beziehungsweise dessen Unterverzeichnis `/bin`, in dem sich die Startdateien befinden. Zum Start werden weitere Parameter, wie die IP-Adresse des Datenbank-Servers, der Port und Zugangsdaten angegeben. Mit Hilfe der Befehle `show dbs` und `show collections` können Datenbanken und – nach Selektion per `use`-Befehl – die jeweiligen Collections angezeigt werden.

Der Code aus Listing 12 zeigt das übliche Vorgehen zum Aufrufen einer Collection. Bei Verbindung mit der Datenbank findet eine Autorisierung (hier mit einem Administrator-Konto) statt. Nach einer Auflistung der Datenbanken und der Auswahl der Datenbank `tweepy`, werden deren Collections abgerufen. In diesem Fall soll zudem die Twitter-ID des zuletzt gespeicherten Tweets angezeigt werden.

[23] http://www.litixsoft.de/mms/
[24] http://www.mongovue.com/
[25] http://3t.io/mongochef/

Dabei werden mehrere Abfrage-Operatoren miteinander kombiniert: `find`, `limit` und `sort`. Die Tweet-ID ist im Schlüssel `id_str` hinterlegt. Da MongoDB jedem Datenbank-Objekt nochmals eine eigene ID zuweist (Feld „_ID“), soll diese bei der Ausgabe ausgeblendet werden, um Verwechslungen zu vermeiden.

Die zweite geschweifte Klammer des Operators `find` definiert die Ausgabe der Felder. Um nur einen Datensatz zu erhalten, wird ein Limit von 1 vorgegeben. Zudem erfolgt eine absteigende Sortierung anhand der Objekt-ID, um den neuesten Tweet (mit der höchsten ID) zu erhalten. Listing 12 stellt nochmals alle vorgestellten Befehle und deren Ergebnisse dar.

Listing 12: Einfache Datenabfrage bei MongoDB

```
C:\<Pfad>\bin>mongo -host <IP-Adresse>
MongoDB shell version: 2.6.7
connecting to: <IP-Adresse>:<Port>/mongo
> use admin
switched to db admin
> db.auth("<DB-Benutzer>","<DB-Passwort>")
1
> show dbs
admin       0.078GB
local       0.078GB
tweepy      0.953GB
> use tweepy
switched to db tweepy
> show collections
chg
merkel
system.indexes
> db.merkel.find({},{id_str: 1, _id:
0}).limit(1).sort({$_id:-1})
{ "id_str" : "565872318040514560" }
>
```

Diese Methode ist die schnellste Möglichkeit zur Abfrage spezifischer Daten bei MongoDB und sollte, wenn möglich, immer auf diese Weise angewendet werden. Ein Sortieren der Daten über Verwaltungssoftware wie Mongo Management Studio benötigt bei großem Datenumfang sehr viel Arbeitsspeicher. Ist nicht genügend Arbeitsspeicher vorhanden, bricht die Operation mit einer Fehlermeldung ab. Ein Grund für den erhöhten Speicher-Bedarf ist die Art des Such-Prozesses: Die

grafische Benutzeroberfläche muss die sortierten Tweets auch gleich am Bildschirm anzeigen, weshalb immer die kompletten Objektdaten (also alle Meta-Daten) verarbeitet werden müssen. Der Befehl über das Shell-Fenster beschränkt die Suche bereits vorher auf das Feld `id_str`. Jedoch besteht auch bei vielen Verwaltungssystemen die Möglichkeit, Such-, Filter- und Sortierbefehle in ein Abfragefeld einzugeben, sodass nicht zwingend ein Zugriff per Shell erforderlich ist.

MongoDB erlaubt mit ähnlich strukturierten Befehlen auch die Selektion, Aggregation beziehungsweise Restrukturierung von Daten. Diese Funktionen betrachtet Kapitel 4.3.2 genauer. Das folgende Beispiel skizziert den Ablauf eines Speicherprozesses.

4.2.2.2 Anwendungsbeispiel: Speichern von Tweets in MongoDB

Nachdem bereits eine Daten-Abfrage auf dem MongoDB-Server getätigt wurde, zeigt dieses Kapitel die Abwandlung des bestehenden Skriptes aus Listing 10 zum Speichern von Tweets. Für Python gibt es ein Programmpaket namens *pymongo*, das den Datenverkehr über die MongoDB Schnittstelle verwaltet. Listing 13 stellt die übliche Konfiguration der Datenbankverbindung dar. Zuerst werden die IP-Adresse des Datenbank-Servers sowie, wenn eingerichtet, Benutzername und Passwort angegeben. Zusätzlich folgt der Name der gewünschten Datenbank. Bei Verbindungsproblemen oder fehlerhaften Anmeldedaten erscheint eine entsprechende Fehlermeldung und das Skript beendet die Verbindung zu MongoDB.

Schließlich wird das Vorgehen bei einem neuen Status definiert. In diesem Fall schreibt das Skript jeden Tweet, der über die Streaming API übermittelt wird, direkt in die Datenbank. Möglich wäre aber auch eine vorherige Definition des Datenumfangs, sodass der Insert-Befehl nur bestimmte Werte (wie Name, Tweet-ID und Datum) umfasst. Da außerdem die Möglichkeit besteht, verschiedene Werte in unterschiedliche Collections zu schreiben, ist bei jedem Insert die Angabe einer Ziel-Collection erforderlich. Der Parameter `safe=True` stellt sicher, dass die Datenbank bei jedem Schreibprozess auf eine Reaktion, genauer gesagt auf die Bestätigung einer korrekten Ausführung des Speicherprozesses, wartet. Dies verhindert einen Datenverlust durch unvollständige Schreibbefehle (bei Verbindungsproblemen oder Überlastung). Bei Fehlern gibt das Skript eine Fehlermeldung aus und unterbricht die Verbindung.

Listing 13: Auszug eines Prozesses zum Speichern von Tweets in MongoDB

```
import pymongo
[...]
class tweepylistener(tweepy.StreamListener):
 def __init__(self, api = None):
  self.api = api or API()
  super(tweepy.StreamListener, self).__init__()

  try:
   connection = pymongo.MongoClient("<IP-Adresse>")
   connection.tweepy.authenticate("<DB-Benutzer>",
                                  "<Passwort>")
   print "Verbindung zu MongoDB erfolgreich"
   self.db = connection.<DB-Name>

  except ConnectionFailure, e:
   sys.sterr.write("keine Verbindung möglich: %s" % e)
   sys.exit(1)connection = pymongo.MongoClient("<IP- \
                                               Adresse>")
[...]
 def on_status(self, status):
  try:
   self.db.<Collection-Name>.insert(json.loads(status),
                                    safe=True)
```

4.2.3 Vergleich der Systeme zur Datenverwaltung

Beide Arten der Datensicherung haben Vor- und Nachteile. Die Speicherung in Einzeldateien ist einfach auszuführen und bedarf keiner speziellen System-Konfiguration. Die erzeugten Dateien sind direkt einlesbar und können ohne Umwandlung weiterverarbeitet oder weitergegeben werden. Es bedarf keiner Installation weiterer Software. Jedoch wird die Dateistruktur gerade beim Sammeln großer Datenmengen schnell unübersichtlich, da große Datenmengen systembedingt auf mehrere Dateien verteilt werden müssen. Für eine vollständige Analyse müssten diese Datenfragmente vorher wieder zusammengefasst werden. Des Weiteren besteht ein hohes Risiko des Datenverlustes: Es sind regelmäßige Backups der Dateien nötig, welche aber nur bei bereits vollständig beschriebenen Dateien möglich sind. Sollte ein Systemfehler beim Schreibprozess auftreten, ist die momentan beschriebene Datei unter Umständen verloren. Somit eignet sich das Speichern in

Einzeldateien eher für Ad-hoc-Analysen oder zum Testen der gewählten Abfrageparameter der APIs.

Datenbanksysteme haben dagegen spezielle Anforderungen an Hard- und Software. Das System muss zuerst installiert und konfiguriert werden, indem beispielsweise Hosts und Clients, Benutzerkonten und Rechte eingerichtet werden. Dafür weist das System einen hohen Grad an Sicherheit hinsichtlich Zugriffsschutz und Datenbeständigkeit auf. Der Zugriff kann mithilfe von Benutzerrollen auf spezifische Datenbanken eingeschränkt und der Datenbestand durch Replikation der Daten auf mehrere Server gewährleistet werden. Zudem gestatten professionelle Datenbanksysteme simultane Lese- und Schreibzugriffe, sodass gleichzeitig sowohl mehrere Sammelprozesse, als auch Ad-hoc-Analysen des bestehenden Datenmaterials möglich sind. Das Speichern in Datenbanken eignet sich aufgrund des erhöhten Einrichtungsaufwands eher für eine größere, langfristig durchgeführte Datensammlung. Tabelle 6 fasst die Vor- und Nachteile der beiden Alternativen nochmals zusammen.

Tabelle 6: Vergleich der Möglichkeiten zur Datenspeicherung

	TEXT-DATEIEN	**DOKUMENTORIENTIERTE DATENBANKEN**
EIGENSCHAFTEN	• JSON-, TXT-, CSV-Format • Record entspricht Textzeile	• BSON/JSON-codiert • Record entspricht Dokument
VORTEILE	• Einfach & schnell anwendbar • Mit „Bordmitteln“ des Betriebssystems realisierbar • Daten direkt einsehbar • Direkte Datenweitergabe möglich	• Datensicherheit und Zugriffsschutz • Gute Performance • Multi-User, Multi-Thread • Grundlegende Analyse- und Aggregations-Funktionen bereits integriert
NACHTEILE	• Mangelnder Schutz vor Datenverlust • Nur ein Lese- oder Schreibprozess gleichzeitig • Bei großem Datenumfang unübersichtliche Einzeldateistruktur	• Installation und Konfiguration zwingend notwendig • Unter Umständen große Hardware-Anforderungen • Programmierkenntnisse für Shell-Kommandos erforderlich

Die Wahl des Systems wird, wie bereits bei der Wahl der Sammelmethode, vom Zweck beziehungsweise der Absicht der Datenverwendung beeinflusst. Generell sind immer Datenbanken zu empfehlen, da diese hinsichtlich Sicherheit, Beständigkeit, Leistung und Verwaltung einfachen Textdateien überlegen sind. Für simple Zwecke genügen jedoch auch Einzeldateien. Zudem besteht bei allen gängigen Datenbanksystemen die Möglichkeit, manuell gesammelte Daten (beispielsweise im CSV-, XML- oder XLS-Format) nachträglich zu importieren.

Ein Vorteil von MongoDB liegt in der Möglichkeit, eine Strukturierung und grundlegende Analyse der Daten bereits mit systemeigenen Funktionen durchführen zu können. Auf diese Analysemöglichkeiten und andere Python-basierte Auswertungen geht das folgende Kapitel genauer ein.

4.3 Methoden der Datenanalyse

Nachdem bereits Ansätze zum Sammeln und Speichern von Twitter-Daten besprochen wurden, folgt nun eine Betrachtung der Möglichkeiten zur Analyse dieser Daten. Dieses Kapitel stellt dabei keinen Anspruch an Vollständigkeit, sondern dient zur Darstellung und Bewertung mehrerer, gängiger Analyseansätze und Methoden. Zuerst wird auf etwaige Probleme bei der Datenanalyse und deren Ursachen eingegangen sowie Ansätze zur Vermeidung und Behebung thematisiert. Da in dieser Arbeit auf dem Datenbanksystem MongoDB basiert, zeigt Kapitel 4.3.2 zunächst die systemintegrierten Funktionen zur Datenverarbeitung und -analyse. Kapitel 4.3.3 befasst sich schließlich mit ausgewählten Methoden des *Natural Language Processing* (NLP) – der automatisierten Inhaltsanalyse der Computerlinguistik. Hierbei gilt zu beachten, dass dies nur eine kleine Auswahl gängiger Analyseverfahren ist, die spezifisch für das folgende Beispiel getroffen wurde.

Der Vergleich der Methoden erfolgt anhand eines Beispiels – der Analyse von Tweets über den Franken-Tatort[26]. Das Datenset umfasst dabei 37.556 vollständige Tweet-Records (inklusive aller Meta-Daten, siehe Abbildung 6 weiter vorne), die ex post mit der Search API (Suchterm: „tatort“) ermittelt wurden. Die Collection `tatort_tweets` in der Datenbank `tatort` beinhaltet in ihrer ursprünglichen Form Tweets unterschiedlichster Sprachen und dadurch mit großer Wahrscheinlichkeit auch irrelevante Tweets. Dies entspricht der üblichen Ausgangssituation bei Twitter-Analysen. Folglich empfiehlt sich in vielen Fällen ein vorgelagertes Bereinigen der Daten.

[26] „Der Himmel ist ein Platz auf Erden“, ARD, 12.04.2015, 20:15-21:45. Erfassung aller Tweets zwischen 05.04., 14:52, und 13.04., 15:34 Uhr.

4.3.1 Vorverarbeitung der Daten

Nach der Datensammlung und vor der eigentlichen Analyse der gesammelten Daten müssen diese zuerst aufbereitet werden, indem sie beispielsweise normalisiert, sortiert, aggregiert, gefiltert und korrigiert werden (Abbildung 12 zeigt den typischen Ablauf der Datenanalyse). Besonders bei großen Datensets sind mehrere vorgelagerte Filter zur Verringerung des Datenumfangs sinnvoll. So können beispielsweise irrelevante Werte der Tweet-Records gelöscht werden. Besonders Angaben über die grafische Gestaltung des Urheber-Accounts (Hintergrundbild, Farbe, etc.) sind für die meisten Auswertungen unbedeutend. Ebenso empfiehlt sich vor der Analyse das Aussondern anderer Sprachen oder Tweets außerhalb eines bestimmten Zeitraums. Die Filterung nach Sprache benötigt allerdings eine verlässliche Spracherkennung. Der Twitter-eigene Spracherkennungs-Dienst ist die momentan zuverlässigste Identifikations-Methode (Carter, Weerkamp & Tsagkias, 2013). Sofern jedoch Daten vor Einführung des Spracherkennungs-Algorithmus durch Twitter im März 2013 (Roomann-Kurrik, 2013) analysiert werden, bedarf es anderer Programme.

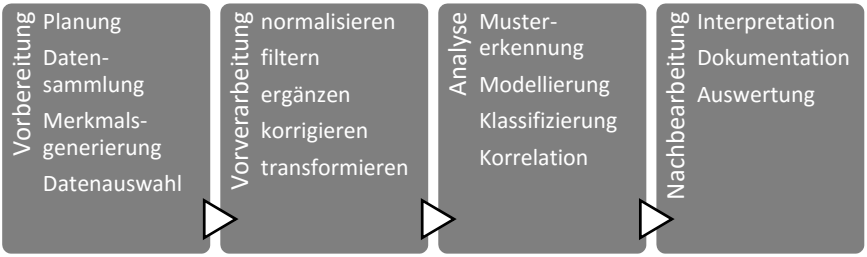

Abbildung 12: Prozess der Datenanalyse. In Anlehnung an Runkler (2000, S. 2).

Neben dem Twitter-eigenen Algorithmus gibt es auch andere Dienste, wie Googles kostenpflichtige *Translate API* (Google Inc., 2015), oder Skripte, wie *Langid* für Python (Lui & Baldwin, 2012). Alle verfügbaren Werkzeuge können keine hundertprozentig (oder annähernd) verlässliche Spracherkennung gewährleisten (Lui & Baldwin, 2014). Die Merkmale der Twitter-Sprache (Dialekt, Neologismen usw.), besonders die Vermischung mehrerer Sprachen in einem Tweet, aber auch die sehr limitierte Textlänge und Häufung von Abkürzungen stellen hier ein großes Hindernis dar (Carter et al., 2013). Da diese Eigenheiten nicht beein-

flussbar sind, besteht in der Zukunft nur die Möglichkeit, intelligentere und zuverlässigere (lernfähige) Algorithmen zu entwickeln. Momentane Lösungen bieten allerdings schon relativ verlässliche Ergebnisse.

Ein großes Problem, nicht nur bei der Inhaltsanalyse, könnten Spam-Tweets darstellen (McCord & Chuah, 2011). Das Veröffentlichen vieler (identischer) Tweets durch mehrere Accounts in kurzer Zeit kann beispielsweise dazu genutzt werden, eigene Hashtags (etwa von Aktionen) in die Trending Topics zu bekommen, um dadurch einfach eine größere Zielgruppe zu erreichen (Benevenuto, Magno, Rodrigues, & Almeida, 2010). Oftmals enthalten Spam-Tweets auch momentan populäre – und themenfremde – Hashtags und einen Link zu einer Webseite. Die Erkennung von Spam kann unter Umständen die Qualität des Datensatzes entscheidend erhöhen. Laut einem Bericht von Twitter, Inc. an die New Yorker Börsenaufsicht SEC schätzt das Unternehmen den Anteil an Spam-Bots, also Accounts, die automatisiert Spam verschicken, auf etwa 5 Prozent der monatlich aktiven Accounts (United States Securities and Exchange Commission [SEC], 2014).

Wenn neben reinen Häufigkeitsauszählungen auch quantitative Inhaltsanalysen durchgeführt werden sollen, muss das Tweet-Datenset noch detaillierter vorverarbeitet werden. Für eine Betrachtung des reinen Twitter-Textes, beispielsweise zur Ermittlung der häufigsten Begriffe eines Datensets, erscheint das Filtern von Mentions, Hyperlinks und Retweets als sinnvoll. Zudem sollten hier Satzzeichen und andere Sonderzeichen entfernt werden. Sofern eine Analyse nicht auf Satz-, sondern auf Wortebene erfolgt, empfiehlt sich der Ausschluss häufiger Begriffe, wie Artikel, Konjunktionen oder Präpositionen aus dem Datensatz. Man spricht hier von *Stoppwörtern*.

Eine Konvertierung in Kleinbuchstaben vereinfacht nicht nur eine spätere Lemmatisierung oder ein Stemming, sondern auch das Erkennen/Filtern von Begriffen. So müssen nicht alle Kombinationen möglicher Groß- und Kleinschreibungen in Begriffskatalogen gespeichert werden. Des Weiteren ist eine Umwandlung von Abkürzungen sinnvoll, da diese in der Internetsprache häufig vorkommen – besonders aufgrund der Beschränkung auf 140 Zeichen pro Tweet erscheint dies bei Twitter als wahrscheinlich. Für diese speziellen Bedarfssituationen muss die Umwandlung allerdings anhand manuell erstellter Lexika erfolgen. Im Gegensatz zu gängigeren Abkürzungen, wie „lol“ oder „ftw“, stellt die Konvertierung unkonventioneller Abkürzungen eine größere Herausforderung dar. Nutzerspezifische Sprache kann selbst durch ein Einlesen in themenspezifische Tweets nicht zwangsläufig erkannt werden. Sehr spezifische Abkürzungen oder falsch geschriebene Begriffe müssten manuell ersetzt und korrigiert werden. Zudem empfiehlt

sich eine Filterung und Korrektur nicht rein alphabetischer Begriffe (z.B. „w00t“, „n8“).

Neben Abkürzungen enthalten Tweets auch häufig Symbole, wie Emoticons, die bestimmte Emotionen ausdrücken. Zwar gibt es einen durch das Unicode Konsortium standardisierten Katalog von *Emojis* (Unicode, Inc., 2015), jedoch verweisen Park, Barash, Fink und Cha (2013) auf kulturelle und persönliche Unterschiede in der Interpretation. Da Emoticons per Definition eine bedeutende Informationsquelle für vermittelte Emotionen sind, sollten diese bei der semantischen Analyse berücksichtigt werden. Um Verzerrungen durch anders interpretierte Smileys zu reduzieren, ist womöglich eine Beschränkung der Analyse auf die gängigsten Symbole – wie beispielsweise *:-) :-(;-) :-P :D* – am sinnvollsten. Dennoch kann es auch hier zu unterschiedlichen Interpretationen kommen, da beispielweise jeder Hersteller von Betriebssystemen für Smartphones eigene Grafiken verwendet (siehe Tabelle 7).

Hinzu kommt die Tatsache, dass Twitter mittlerweile ein eigenes Set an *Twemojis* zur Verfügung stellt, das die von den jeweils unterschiedlichen Betriebssystemen verwendeten Icons in der Darstellung auf der Twitter-Webseite vereinheitlicht (Davidson, 2014). Das Beispiel unten zeigt zudem, dass die Gefahr besteht, die ursprüngliche Aussage des Emojis (hier: Schläfrigkeit) falsch zu deuten. Eine mögliche Interpretation wäre in diesem Fall auch ein trotziges Spucken oder Weinen.

Vorgelagerte Filter und Verarbeitungsschritte ermöglichen eine Reduktion des Datenumfangs und eine Verbesserung der Lesbarkeit von Texten. Zwar gehen dadurch möglicherweise Informationen verloren, dennoch kann die Aussagekraft des Inhalts durch Ausschluss irrelevanter Zeichen oder ganzer Text-Fragmente steigen. Zudem sinkt das Datenvolumen, was die Analyse beschleunigen kann. Vor der eigentlichen Analyse sollte deshalb immer eine Bereinigung erfolgen.

Tabelle 7: Unterschiede in der Darstellung von Emojis. Bildquelle: Unicode, Inc. (2015).

HERSTELLER	APPLE	GOOGLE	MICROSOFT	TWITTER
SYSTEM	iOS und OS X	Android	Windows	Twemoji
SCHLÄFRIGES EMOJI				
WEINENDES EMOJI				

4.3.2 Verarbeitung und Analyse mit MongoDB

MongoDB stellt für das sogenannte *Preprocessing* eine Vielzahl an Möglichkeiten zur Manipulation und Analyse von Daten zur Verfügung. Diese eignen sich jedoch eher zur allgemeinen Strukturierung der Daten, als für die Textmanipulation oder detailliertere Analysen. So können Datensätze gefiltert oder gruppiert werden, während eine inhaltliche Verarbeitung von Texten (z.B. Stoppwort-Filter, Tokenisierung) nur auf Umwegen über andere Programme oder Skripte möglich ist.

Ein wichtiges Toolkit für die Verarbeitung auf Datensatz-Ebene sind die Methoden zur Datenaggregation, welche drei wesentliche Funktionen beinhalten: Abfragemethoden zur Aggregation, das *Aggregation Framework* sowie das *MapReduce*-Verfahren. Die drei Methoden unterscheiden sich im Allgemeinen vor allem in ihrer Performance und ihrem Funktionsumfang. Während die Abfragemethoden einen geringen Funktionsumfang, aber eine hohe Leistung hinsichtlich Verarbeitungsgeschwindigkeit vorweisen, bietet MapReduce eine sehr hohe Funktionalität (auch für Analysen) mit jedoch eingeschränkter Performance. Einen Mittelweg zwischen Leistung und Funktionalität geht das Aggregation Framework.

4.3.2.1 Abfragemethoden zur Aggregation

MongoDB stellt mit mehreren Abfragemethoden eine Grundfunktionalität zur Aggregation von Daten zur Verfügung. So können beispielsweise Häufigkeitsauszählungen von (gruppierten) Werten bereits mit den internen Funktionen der Datenbank durchgeführt werden. Da die Möglichkeiten zur Einschränkung der Suche über die REST APIs begrenzt sind, wurden im eingangs erwähnten Erhebungsfall auch Tweets mit dem Begriff Tatort erfasst, die deutlich vor dem Sendetermin lagen. Das Datenset besteht folglich aus Tweets zwischen dem 05.04. und 13.04. Für die Datenanalyse sollen zunächst alle Einträge vor dem 11.04. gefiltert werden. Um den Datenumfang anhand des Datums auf relevante Tweets zu begrenzen, muss das Feld `created_at` zunächst umformatiert werden. Twitter verwendet das Unicode-Format, welches aber nicht direkt kompatibel mit den Filteroperatoren von MongoDB ist. Listing 14 beinhaltet ein Skript zur Konvertierung der Zeitstempel in das *Datetime*-Format nach dem Standard ISO 8601 (International Organization for Standardization [ISO], 2015).

Listing 14: Konvertierung des Datumsformats

```
import pymongo, datetime
from pymongo import MongoClient

connection = pymongo.MongoClient("<DB-IP>")
connection.admin.authenticate("<DB-Benutzer>","<Passwort>")
db = connection.<DB-Name>
collection = db.<Collection-Name>
tweets = collection.find({})
for tweet in tweets:
 tweetdate = tweet[u'created_at']
 if(type( tweetdate ) == unicode):
  newdate = datetime.datetime.strptime(tweetdate,
            '%a %b %d %H:%M:%S +0000 %Y')
   pointer = tweet[u'_id']
   collection.update({'_id': pointer},
                     {'$set': {'created_at': newdate}})
 else:
  skip=1
  pass
print("Alle Daten konvertiert")
if skip==1:
 print("Mindestens 1 Datum wurde übersprungen")
```

Nach der Umwandlung des Datums besteht nun eine Filter-Möglichkeit nach Datum. Dadurch ergibt sich ein kleinerer Datensatz mit 20.310 Einträgen. Da der Tatort vornehmlich ein deutsches beziehungsweise deutschsprachiges Phänomen ist, interessieren vor allem Tweets auf Deutsch. Eine Sichtung des Materials ergab, dass manche Twitter-Nutzer momentan populäre Hashtags ohne thematischen Bezug in ihre Nachrichten einbauen, um vermutlich eine größere Beachtung zu finden (Spam). Dies konnte man vor allem bei fremdsprachigen Tweets erkennen. Auch deshalb werden für diese Analyse nur deutschsprachige Tweets analysiert. Twitter stellt für die Textsprache das Entity `iso_language_code` zur Verfügung. Twitters eigener Spracherkennungs-Algorithmus erkennt anhand mehrerer Parameter automatisiert die Textsprache und trägt diese als Code in das Feld ein. Aktuelle Studien bescheinigen eine relativ hohe Zuverlässigkeit, wenngleich unbereinigte Tweets eine nicht zu vernachlässigende Fehlerquote bei der Erkennung aufweisen (vgl. Kapitel 4.3.1).

Listing 15: Abfragemethoden zur Aggregation in der MongoDB-Shell

```
> db.tatort_tweets.count()
37556
> db.tatort_tweets.count({
...     created_at: {
...         $gte: ISODate("2015-04-11T12:00:00.000Z"),
...         $lt: ISODate("2015-04-13T12:00:00.000Z")}
... })
20310
> db.tatort_tweets.find({
... ...     created_at: {
... ...         $gte: ISODate("2015-04-11T12:00:00.000Z"),
... ...         $lt: ISODate("2015-04-13T12:00:00.000Z")},
... ...     "metadata.iso_language_code" : "de"
... ...    }).count()
18517
> db.tatort_tweets.group({
...     key: {"metadata.iso_language_code" : 1},
...     cond: {
... ...     created_at: {
... ... ...    $gte: ISODate("2015-04-11T12:00:00.000Z"),
... ... ...    $lt: ISODate("2015-04-13T12:00:00.000Z")}},
...     initial: {count:0},
...     reduce: function(curr,result){
...     result.count++;}
})
```

```
[
        {
                "metadata.iso_language_code" : "de",
                "count" : 18517
        },
        {
                "metadata.iso_language_code" : "en",
                "count" : 739
        },
        [...]
```

Listing 15 beginnt mit einer einfachen Auszählung aller Dokumente in einer Collection. Diese allgemeine Abfrage liefert einen erstmaligen Überblick der Tweet-Anzahl. Hierfür wurde der Count-Befehl verwendet und schließlich um einen Datumsfilter erweitert. Um Tweets des relevanten Zeitraums in deutscher Sprache zu erhalten, wird Count mit dem Filter Find ergänzt, der die Abfrage des ursprünglichen Befehls ersetzt. Der Datensatz beinhaltet demnach 18.517 Tweets in deutscher Sprache. Zudem besteht die Möglichkeit, Daten anhand eines Schlüssels zu gruppieren und zu zählen (hier die Textsprache). Der Group-Befehl kombiniert dabei Count mit einem Zähler und Datumsgrenzen als Bedingung.

Die Abfragen mittels Count und Group stellen zwei sehr einfache und schnelle Möglichkeiten zur Häufigkeitsanalyse. Diese Methoden geben jedoch nur (gezählte) Werte in der Kommandozeile aus und erlauben weder ein direktes Speichern der Ergebnisse in Collections, noch eine Restrukturierung der Daten. Zudem sind diese Methoden nur auf 20.000 Datensätze beschränkt und bei Datensets, die mittels Sharding auf mehrere Server verteilt sind, nicht anwendbar (Trelle, 2014, S. 200). Für diese Zwecke und für komplexere Abfragen eignet sich das Aggregation Framework.

4.3.2.2 Aggregation Framework

Das Aggregation Framework wurde eingeführt, um die technische Lücke zwischen den schnellen aber einfach konzipierten Abfrage-Methoden und dem komplexen aber mächtigen MapReduce zu schließen. Das Framework ist in zwei Komplexe gegliedert: einer stufenweisen *Pipeline*, die Dokumente sequentiell verarbeitet, und *Expressions*, die die Dokumente innerhalb der Pipeline manipulieren (Chodorow, 2013, S. 127-130). Eine Aggregationspipeline besteht aus einer Kette mehrerer vordefinierter Prozeduren, die sequentiell auf die Dokumente angewendet werden. So kann eine Vielzahl an Operationen mithilfe einer mehrstufigen

Pipeline verbunden werden. Die Pipeline leitet dabei die Daten von einer Prozessstufe mit jeweils spezifischen Operationen zur nächsten weiter. Abbildung 13 stellt den Ablauf der Aggregation schematisch dar.

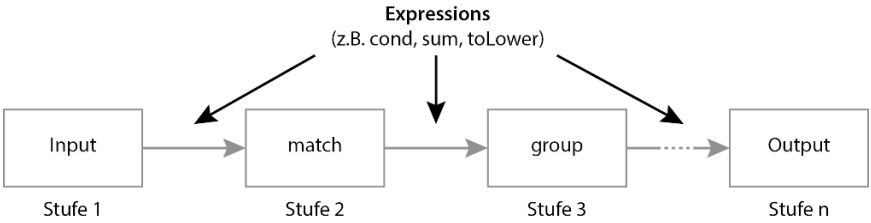

Abbildung 13: Schematische Darstellung einer Aggregation Pipeline. Eigene Darstellung.

Die Manipulation an Dokumenten erfolgt innerhalb einer Stufe der Pipeline über Expressions. Diese Expressions reagieren dabei immer nur auf den momentanen Zustand der aktuell verarbeiteten Dokumente. Es können folglich keine zusätzlichen Daten anderer Dokumente außerhalb der Pipeline hinzugefügt werden. Die möglichen Funktionen von Expressions gehen von einfachen Bedingungs-Verknüpfungen (`$and`, `$or`, …) und Vergleichen (`$eq`, `$gt`, …) über einfache mathematische Funktionen (`$multiply`, `$add`, …) bis hin zur Manipulation von Strings (`$toLower`, `$toUpper`, …).

In Listing 16 wird zunächst ein einfacher `aggregate`-Befehl ausgeführt: Alle Tweets der Collection `tatort_tweets` sollen anhand der durch Twitter erkannten Textsprache `iso_language_code` gruppiert werden. Schließlich wird die Anzahl der Tweets je Sprache gezählt und in einer absteigend sortierten Liste ausgegeben. Bereits hier zeigen sich die Vorteile gegenüber den einfachen Abfrage-Methoden: Es ist gibt keine Limitierung der Datenmenge[27] und es besteht die Möglichkeit, Ergebnisse anhand von Werten zu sortieren.

Der zweite `aggregate`-Befehl verdeutlicht das Konzept von Pipeline und Expressions. Zuerst werden Tweets ausgewählt, die im gewünschten Zeitraum veröffentlicht wurden, und anhand der Sprache gruppiert. Daneben erfasst ein Zähler die Anzahl der Tweets je Sprache und errechnet die durchschnittliche Retweet-Zahl je Sprache. Die nächste Stufe sortiert die ermittelte Liste mit den Ergebnissen absteigend anhand der Gruppengröße (nach Sprache) und begrenzt das Ergebnis

[27] Anmerkung: Mit zunehmender Größe der Collection steigt folglich auch die Dauer der Ausführung der Datenbank-Befehle.

auf 10 Einträge. Diese *Top 10*-Liste von Tweets und durchschnittlichen Retweets nach Sprache schreibt die letzte Stufe in die neue Collection `top`. Die nun dauerhaft gespeicherten Datensätze stehen so als Ausgangsmaterial für weitere Analysen zur Verfügung. Der `find`-Befehl gibt die Liste schließlich aus.

Listing 16: Aggregation-Framework in MongoDB

```
> db.tatort_tweets.aggregate(
... {$group: {_id: "$metadata.iso_language_code",
... ... count: {$sum:1}}},
... {$sort: {count: -1}});
{ "_id" : "de", "count" : 18517 }
{ "_id" : "en", "count" : 739 }
[...]

> db.tatort_tweets.aggregate(
... { $match : {"created_at":
... ...{$gte: ISODate("2015-04-11T12:00:00.000Z"),
... ...$lt: ISODate("2015-04-13T12:00:00.000Z")}}},
... { $group : {
... ... _id: "$metadata.iso_language_code",
... ...count: {$sum:1},
... ...avgRetweets: {$avg: "$retweet_count"}}},
... {$sort: {count:-1}},
... {$limit: 10},
... {$out: "top"});

> db.top.find()
{ "_id" : "de", "count" : 18517, "avgRetweets" :
6.2377814980828425 }
{ "_id" : "en", "count" : 739, "avgRetweets" :
0.5940460081190798 }
{ "_id" : "und", "count" : 476, "avgRetweets" :
2.926470588235294 }
{ "_id" : "fr", "count" : 134, "avgRetweets" :
0.1417910447761194 }
[...]
```

Dieses Beispiel zeigt, dass bereits *in* der Datenbank eine sinnvolle Selektion und Umstrukturierung von Twitter-Daten erfolgen kann. So besteht die Möglichkeit, Daten anhand von (errechneten) Werten zu vergleichen, zu gruppieren und sortieren sowie die Ergebnisse in eine separate Collection zu schreiben. Jedoch gibt es

auch hier, wie bei den Abfragemethoden technische Einschränkungen: Wenn `group` und `sort`-Befehle bereits zu Beginn der Pipeline eingesetzt werden, hängt die Zahl der maximal analysierbaren Tweets von der Verfügbarkeit des Arbeitsspeichers ab. Für kumulative Operationen wie die Gruppierung oder Sortierung von Daten muss zuerst das komplette Datenset eingelesen werden. Werden dabei mehr als 10% des verfügbaren Arbeitsspeichers belegt, bricht der `aggregate`-Prozess ab.

Neben einer guten Rechnerausstattung ist es daher sinnvoll, Daten in frühen Stufen der Pipeline zu filtern. Beispielsweise können einzelne Werte extrahiert und in eine neue Collection geschrieben werden. Diese Collection kann dann wiederum als Ausgangspunkt für weitere Analysen im Aggregation Framework dienen, oder – bei Export der Daten – für andere Analyse-Programme. Zudem muss beachtet werden, dass der Algorithmus zur Spracherkennung von Twitter nicht immer zuverlässig arbeitet und auch nicht alle Tweets anhand ihrer Sprache klassifizieren kann. In Listing 16 befanden sich beispielsweise 467 undefinierbare Tweets („`_id`“: „`und`“).

Neben dem Aggregation-Framework, das wohl für die meisten Anwendungsfälle genügt, unterstützt MongoDB noch eine weitere Technik zur Datenaggregation, die plattformübergreifend bei sehr großen und stetig wachsenden Datensets angewendet wird: *MapReduce*.

4.3.2.3 MapReduce

MapReduce ist eine von Google, Inc. (Dean & Ghemawat, 2008) konzipierte Technik zum kontinuierlichen Verarbeiten sehr großer Datensets. Da sie eine sehr hohe Leistungsfähigkeit hat, aber eher kompliziert und programmierintensiv in der Einrichtung ist, eignet sie sich vor allem für *Big Data*, also sehr großen und stetig wachsenden Datenmengen. Insgesamt ist die Verarbeitungsgeschwindigkeit im Vergleich zum Aggregation Framework jedoch langsamer, was auch darauf zurückzuführen ist, dass die Verarbeitung des Aggregation Frameworks vor allem im (schnellen) Arbeitsspeicher stattfindet, während MapReduce die Daten direkt auf der DB verarbeitet und die einzelnen Dokumente während der Bearbeitung (für Millisekunden) sperrt (Trelle, 2014, S. 231). Dafür können bei MapReduce deutlich mehr Daten verarbeitet werden. Der Algorithmus gliedert sich in drei Phasen: *Map*, *Group/Sort* und *Reduce*. Diese müssen zuvor einzeln mittels Javascript programmiert werden.

Jeder MapReduce-Prozess beginnt mit einer Map-Phase. Hier werden aus allen verfügbaren Dokumenten, die auf der Datenbank verteilt sind, vorher definierte Teilinformationen extrahiert. Das könnten alle Wörter eines Textes sein oder ein Zahlenwert eines bestimmten Schlüssels (wie die Anzahl der Follower eines Twitter-Nutzers). Danach folgt eine Phase des Gruppierens und Sortierens. Die extrahierten Informationen werden hier sortiert und in Bündel identischer Daten aggregiert. In der Reduce-Phase erfolgt schließlich die Reduktion der Listen-Einträge zu einem Element. (Chodorow, 2013, S. 140)

Zum besseren Verständnis zeigt das folgende, einfache Beispiel, den Ablauf des Prozesses, wie ihn Abbildung 14 vereinfacht darstellt. Ziel ist die Ermittlung der 10 häufigsten Wörter aller Tweets zum Franken-Tatort anhand der *Word Count* Methode. Das Zählen der Worthäufigkeit in Texten ist eine elementare Methode der Computerlinguistik.

Der Input für MapReduce umfasst auch in diesem Fall vollständige Tweet-Datensätze. Folglich muss der für die Analyse relevante Tweet-Text erst aus den Datensätzen extrahiert werden. Diese Erfassung aller Inhalte des Feldes `text` findet im ersten Schritt, der Map-Phase, statt. Dabei iteriert das Javascript in Listing 17 jeden Text und unterteilt diesen anhand von Wortgrenzen (hier: Leerzeichen) in einzelne Wörter. Danach folgt eine Bereinigung der Ergebnisse um Leerzeichen, Sonderzeichen und Füllwörter sowie ein Konvertieren in Kleinbuchstaben. Mit der Umwandlung in Kleinbuchstaben wird sichergestellt, dass Begriffe mit unterschiedlichen Schreibweisen (wie TatOrt, tatort, Tatort) als identische Werte erkannt werden können. Es entstehen nun Key-Value-Paare, in denen jedes Wort einzeln aufgelistet wird und die Häufigkeit 1 erhält, also beispielsweise `Tatort, 1`. Dabei gilt zu beachten, dass auch Wörter, die im selben Dokument mehrfach vorkommen, als einzelne KeyValue-Paare aufgelistet werden. Diese Paare tauchen in der Liste dementsprechend mehrfach als Dubletten auf und werden erst in der Reduce-Phase zusammengeführt.

In Phase 2 findet ein Sortieren und Gruppieren der Key-Value-Paare statt. Hier werden gleiche Worte dem gleichen Reducer zugewiesen um eine globale Aggregation der Wort-Häufigkeit zu ermöglichen. In diesem Beispiel erfolgen eine alphabetische Sortierung und schließlich eine Zuordnung jeweils gleicher Worte zu einem Reducer. In der anschließenden Reduce-Phase iteriert jeder einzelne Reducer über die ihm zugewiesene Wortmenge und ermittelt so die Summe der Begriffe. Aus `(Tatort, 1)` und `(Tatort, 1)` wird schließlich `(Tatort, 2)`. Am Ende des Prozesses schreiben alle Reducer ihre Ergebnisse als aggregierte Key-Value-Paare in die Ziel-Collection.

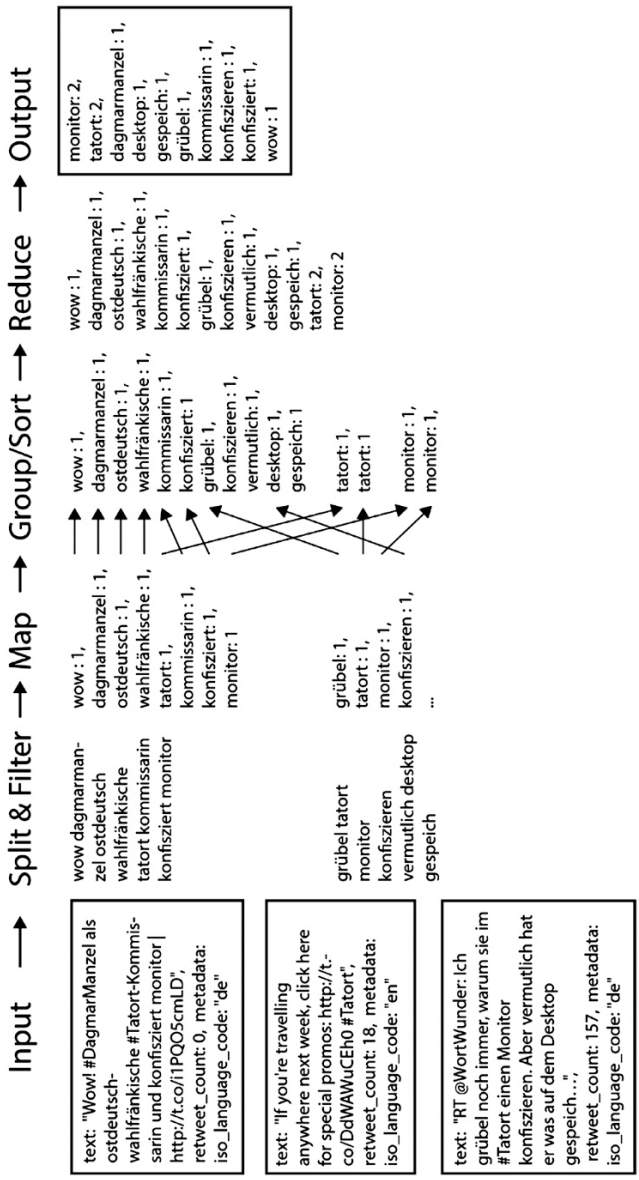

Abbildung 14: MapReduce-Prozess anhand der Word Count Methode. Eigene Darstellung.

Listing 17: JavaScript zur Definition der Map- und Reduce-Funktion bei Word-Count

```
// Map-Funktion
map = function() {
  if (!this.text) {
    return;
  }
  this.text.split(' ').forEach(function(word) {
    word = word.replace(/\s/g, "");
    word = word.replace(/[\.,- \/#!$%\^&\*;:{}@=\-_´`~()]/g,
                        "");
    word = word.replace(/der|die|das|ich|du|er|sie|
                        es|wir|ihr|ihnen|dir|du|dem|den|
                        ist|ein|eine|einer|eines|sein|mein|
                        dein|im|um|auch|ja|nein|kein|
                        immer|noch|und|warum|rt/gi, "")

    if(word != "") {
      emit(word.toLowerCase(), 1)
      }
  });
};

// Reduce-Funktion
reduce = function(key, values) {
  return values.length;
};
```

Der eigentliche MapReduce-Prozess wird über die MongoDB-Shell gestartet und bedient sich einer übersichtlichen Syntax. Sie beinhaltet die Namen der Map- und Reduce-Funktion sowie weitere Angaben über die Ausgabe der Ergebnisse. Wenn der jeweilige Map- und Reduce-Code bereits vorher in das Kommandofenster kopiert wurde, reicht ein einfacher Verweis auf den jeweiligen Funktionsnamen, um diesen einzubinden. Möglich wäre aber auch eine absolute oder relative Pfadangabe zum lokal gespeicherten JavaScript.

In diesem Fall speichern die Reducer ihre Ergebnisse in der Collection `wordcount`. Zuvor sortiert ein Filter alle Tweets aus, die nicht anhand der automatischen Spracherkennung von Twitter als deutschsprachig identifiziert wurden. Da der `query`-Filter vor dem Mapping einsetzt, reduziert dies unter Umständen auch die Bearbeitungszeit des Prozesses. In der Syntax können zudem weitere Verar-

beitungsschritte, wie das Errechnen von Kennzahlen (z.B. die Häufigkeit des Wortes innerhalb eines Satzes), initiiert werden. Auch besteht die Möglichkeit zum Vorsortieren und Begrenzen des Inputs für die Mapper.

Listing 18: Shell-Befehl zur Initiierung des MapReduce-Prozesses, nachdem die Map- und Reduce-Funktion geladen wurde

```
db.tatort_tweets.mapReduce(
    map,
    reduce,
    {
        out: "wordcount",
        query: {"metadata.iso_language_code" : "de"}
        }
)
```

Jede Phase des MapReduce-Algorithmus erlaubt das Ausführen einfacher bis sehr komplexer Prozesse: Dies reicht von simplen Filtern bis zu umfangreichen Berechnungen von Kennzahlen und der Identifizierung von Mustern. Ziel des Prozesses ist dabei immer die Reduzierung des Datenumfangs auf wenige Daten oder Kennziffern. MapReduce erlaubt zudem eine hohe Flexibilität bei der Ausgabe und Speicherung der Ergebnisse: So können die erzeugten Daten in einer separaten Collection innerhalb oder außerhalb der verwendeten Datenbank abgelegt werden oder bereits vorhandene Collections überschreiben und so den Speicherbedarf verringern. Dennoch ist MapReduce für die meisten Analysen von Twitter-Daten überdimensioniert. Bereits elementare Abfragen erfordern Programmierkenntnisse in JavaScript und einen größeren Zeitaufwand als das Aggregation-Framework (Chodorow, 2013, S. 140). Folglich ist diese Analysemethode vor allem für sehr großvolumige Berechnungen an kontinuierlich wachsenden Datensets geeignet.

4.3.2.4 Vergleich der Ansätze

Mit den in MongoDB integrierten grundlegenden Abfragemethoden stehen bereits einfache Möglichkeiten zum Zählen, Filtern und Restrukturieren zur Verfügung, deren Nutzung bereits während der Datensammlung möglich ist. Für größere und schnell ansteigende Datenmengen eignen sich jedoch nur das Aggregation Framework und MapReduce. Diese Werkzeuge ermöglichen nicht nur komplexere Ab-

fragen und Befehle, sondern funktionieren auch problemlos bei sehr großen Datensets. Allerdings ist hier die Befehls- und Prozessstruktur deutlich komplizierter und unübersichtlicher. Für MapReduce benötigt man zudem Kenntnisse in der Programmiersprache JavaScript. Im Vergleich zum Aggregation Framework ist MapReduce für sehr große Daten deutlich besser geeignet, da es keine Begrenzung hinsichtlich der zu verarbeitenden Datenmenge gibt. Allerdings ist bei dieser Methode auch der Leistungsbedarf (CPU-Auslastung) höher bei vergleichsweise geringerer Geschwindigkeit (Marturana, 2015). Tabelle 8 fasst die Vor- und Nachteile noch einmal zusammen.

Tabelle 8: Vergleich der Aggregations-Methoden von MongoDB

	ABFRAGEN	AGGREGEATION FRAMEWORK	MAPREDUCE
VORTEILE	• Schnell • Einfach und übersichtlich	• Vielzahl umfassender Funktionen • Stufenweiser Aufbau • Export der Ergebnisse • Sharding-Unterstützung	• Sehr mächtig • Kein Datenlimit • Export der Ergebnisse • Sharding-Unterstützung
NACHTEILE	• Kein Speichern/Export der Ergebnisse • Verarbeitet maximal 20.000 Datensätze • Keine Sharding-Unterstützung	• Datenaufnahme bis maximal 10 % des Arbeitsspeichers • Programmierung unübersichtlich bei vielen Stufen	• JavaScript-Kenntnisse erforderlich • Komplex • Langsamer als das Aggregation Framework • Erhöhter Leistungsbedarf

Die in diesem Kapitel vorgestellten Methoden sind alle mit den Bordmitteln[28] von Python und MongoDB realisierbar. Jedoch eignen sie sich vor allem nur für allgemeine quantitative Analysen, wie der Generierung von Häufigkeitsverteilungen oder der Berechnung von Kennzahlen. Für detaillierte Inhaltsanalysen von Tweets

[28] Mit Ausnahme kleinerer Zusatzpakete, wie beispielsweise dem Schnittstellen-Paket *pymongo*.

bedarf es spezialisierter Software aus dem Bereich des Natural Language Processing. Das folgende Kapitel stellt deshalb ausgewählte Programme und Methoden der Computerlinguistik vor, die über das grundlegende Zählen der Worthäufigkeit aus dem letzten Beispiel hinausgehen.

4.3.3 Natural Language Processing (NLP)

Für die computergestützte Textanalyse über Python eignet sich das umfangreiche Softwarepaket *Natural Language Toolkit* (NLTK) (Loper & Bird, 2002). NLTK ist eine Sammlung von Bibliotheken und Programmen, die eine plattformunabhängige Programmierumgebung für NLP-Programme in Python bereitstellt. Es beinhaltet unter anderem standardisierte Klassen für Part-of-Speech-Tagging, Text-Klassifizierung und syntaktische Analyse (Bird et al., 2009, S. 4). Zudem besteht die Möglichkeit, Texte direkt aus einer Datenbank auf MongoDB abzurufen. NLTK wird wie andere Module installiert, benötigt aber noch zusätzliche Bibliotheken zur Darstellung von Plots oder mathematischen Berechnungen.

Die im vorigen Kapitel begonnene Analyse von Tweets zum Frankentatort wird in diesem Kapitel fortgesetzt und vertieft. Vorbereitend (siehe Listing 19) filtert ein `aggregate`-Befehl alle Tweets nichtdeutscher Sprachen außerhalb des relevanten Zeitraums und schreibt diese in eine separate Collection, die nun als Datengrundlage für den Textkorpus dient.

Listing 19: Vorbereitung der NLP-Analyse in MongoDB

```
> db.tatort_tweets.aggregate(
... { $match : {"created_at":
... ... {$gte: ISODate("2015-04-11T12:00:00.000Z"),
... ... $lt: ISODate("2015-04-13T12:00:00.000Z")},
... ... "metadata.iso_language_code" : "de"}},
... {$out: "dt_tweets"});
```

4.3.3.1 Anwendungsbeispiel: Computerlinguistische Analyse des Franken-Tatorts

Zunächst wird über die MongoDB-Schnittstelle eine Verbindung zur Collection hergestellt und die Einträge ausgelesenen. Als Index für diesen Textkorpus dient in diesem Fall der Veröffentlichungszeitpunkt (`created_at`). Eine Ausgabe der

Häufigkeitsverteilung aller enthaltenen Tweets über den definierten Zeitraum dient der ersten Übersicht (siehe Abbildung 15). Tweets mit dem Begriff Tatort erscheinen demnach vor allem während der Ausstrahlung und in den Stunden danach. Daraus könnte man ableiten, dass hier das typische Phänomen des *second screen* vorliegt: Zuschauer nutzen während dem Fernsehschauen weitere Geräte (wie Tablets oder Computer) und kommentieren das Gesehene in Sozialen Medien (Courtois & D'heer, 2012).

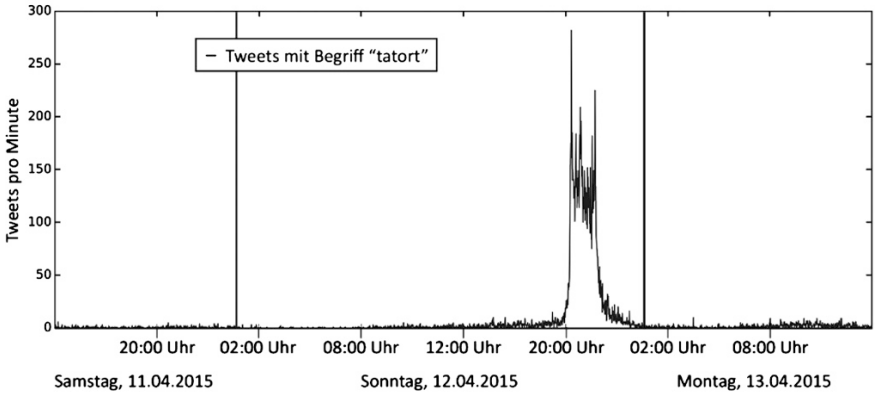

Abbildung 15: Verteilung der Tweets zum Franken-Tatort nach Uhrzeit.

Wie und über was die Twitter-Nutzer/-innen schreiben, soll nun eine detaillierte Textanalyse ermitteln. Wie bereits erwähnt wurde, bedarf die Analyse von Twitter-Text eine gründliche Vorbereitung. Im Gegensatz zu (relativ standardisierten) Buchtexten oder Nachrichtenartikeln, die eine korrekte Satzstruktur und Rechtschreibung aufweisen, bestehen Tweets häufig aus Abkürzungen, Neologismen, mehreren Sprachen, Sonderzeichen, Links und Umgangssprache (siehe dazu auch Kapitel 4.3.1). Deshalb muss der Textinput zunächst bereinigt werden, um ihn in der späteren Analyse zuverlässig zu verarbeiten.

Dafür extrahiert das Programm aus Anhang B[29] zunächst aus allen Tweet-Texten die einzelnen Wörter – man spricht hier von *Tokenisierung*. Um häufige Füllwörter aus der Häufigkeitsanalyse auszuschließen, sollen diese Stopwords in einem zweiten Schritt erkannt und gefiltert werden. NLTK stellt bereits Kataloge

[29] Aufgrund der Länge des Programmcodes erfolgt die Darstellung dieses Skript nur im Anhang.

solcher Stopwords (u.a. auf Englisch und Deutsch) zur Verfügung. Zusätzlich werden unter `customstopwords` eigene Begriffe definiert, die für den Franken-Tatort in einer hohen Häufigkeit vermutet wurden (wie zum Beispiel *Tatort*, *heute*, *gleich*). Die erkannten Wörter werden in Kleinbuchstaben umgewandelt. Das Skript liest alle Text Entities ein und filtert alle definierten Stopwords.

Zudem definiert das Skript alle Begriffe, die mit einem @-Zeichen beginnen, als Mentions und alle Wörter mit vorangestellter # als Hashtags. Alle Satzteile, die mit *http* oder *www* beginnen, werden als Links klassifiziert. Die Häufigkeitsanalyse schließt auch diese drei Wort-Klassen aus, wie auch Begriffe mit weniger als drei Buchstaben. Ebenso überprüft das Programm, ob die ermittelten Wörter rein aus Buchstaben bestehen. So fallen beispielsweise Begriffe wie *2rad* oder *kla4* aus der Untersuchung heraus. Das Histogramm in Abbildung 16 zeigt das Ergebnis der Analyse.

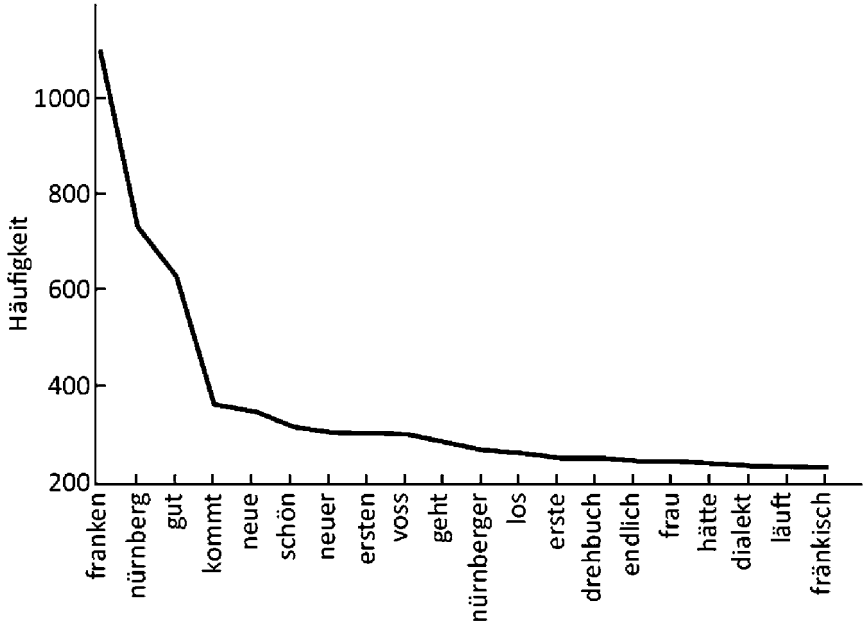

Abbildung 16: Häufigkeit der Top 20 Begriffe zum Franken-Tatort. Eigene Darstellung.

Bei Betrachtung des Resultates ragt die Prominenz des Wortes „gut“ hervor. Auf den ersten Blick scheint der Tatort eine positive Wirkung erzielt zu haben. Jedoch besteht auch die Möglichkeit, dass diese Begriffe in einem anderen Kontext verwendet wurden, wie „nicht schön“ oder „gar nicht gut“. Deswegen erscheint hier eine Betrachtung der Konkordanz, also die kontextuelle Einbettung eines Wortes in einem Satz, als sinnvoll. Für diese Auswertung wurde ein angepasstes Skript verwendet, das zwar analog zum ersten Filter alle Wörter separiert, jedoch keine Stopwords filtert (siehe Anhang B). Der Datensatz enthielte sonst beispielsweise keine wertenden Wörter wie „nicht“ oder „voll“. Das unter `sel_tokens2` gefilterte Datenset wurde nur um Links und Mentions bereinigt. Zudem erfolgt die Analyse nicht auf Wort-, sondern auf Satzebene. Auch für diese Tokenisierung stellt das NLTK eine Methode zur Verfügung. Die Separation erfolgt anhand gängiger Satzzeichen, die im Üblichen das Ende eines Satzes markieren: *:,.!?*.

Listing 20: Konkordanz des Wortes "gut" in Tweets zum Franken-Tatort

```
Displaying 25 of 677 matches:
- premiere ganz gut an . # tatort http :// t . co / ormg6
@ daserste sehr gut ! gute besetzung , von der handlung h
rt ich fand ihn gut mit potenzial nach oben . charaktere
rt ich fand ihn gut . solider fall und das neue duo - int
 amp ; # tatort gut \ xfcberstanden ? - # feelixgmbh star
nderdiddle ganz gut # tatort ', u ' rt @ media_n_mngmnt :
e4mlich richtig gut ! # dadord https :// t . co / z2hvg6k
der wirklich so gut ?', u '@ tatort @ brfrankentatort gib
 ', u ' war der gut der # tatort ? kann ihn fr \ xfcheste
rt war noch nie gut , da machen die amis bessere filme ,
ng nach richtig gut !\ nbester spruch : do is die katz gf
rt sensationell gut ! mehr davon ', u ' ich habe kein ein
ranken - tatort gut fand :', u ' rt @ mpjhaug : hab dich
in handwerklich gut gemachter krimi viel \ u2026 ', u ' h
in handwerklich gut gemachter krimi viel besser .', u ' r
ranken - tatort gut fand :', u ' rt @ tweetbarth : viel s
er ist wirklich gut gewesen ! # megusta ', u ' top 5 der
vielleicht doch gut , dass der n \ xe4chste # bpt der # p
n n \ xfcrnberg gut ! wobei das fr \ xe4nkisch da schon e
 ja ned so arch gut .', u ' rt @ loausro : @ condra1902 a
# tatort ', u ' gut , dass jetzt niemand mehr \ xfcber de
 tatort auch so gut gefallen ;) weiterhin w \ xfcnschen w
haut , ist echt gut geworden :) # nuernberg # tatort ', u
 ganz besonders gut hat mir am # tatort gefallen , dass n
und war richtig gut . weiter so !', u '# tatort \ n # rot
```

Der Index für das *Key Word in Context* „gut“, dargestellt in Listing 20, zeigt für die ersten Tweets eine mehrheitlich positive Einbettung des Begriffs. Um eine allgemein gültige Aussage treffen zu können, muss die Konkordanz für jeden Fall betrachtet werden. Diese Aufgabe übernimmt die Analyse von Kollokationen, die die häufigsten Wortpaare ausgibt. Das Skript im unteren Teil von Anhang B sucht nach den 20 häufigsten Bigrammen, die den Begriff „gut“ enthalten und öfter als 10 Mal im gesamten Textkorpus auftauchen. Der Output bestätigt die These der guten Bewertung des Franken-Tatorts.

Listing 21: Häufigste Bigramme zum Franken-Tatort

```
[(u'aufn', u'gut'), (u'gut', u'isser'), (u'richtig',
u'gut'), (u'sehr', u'gut'), (u'wirklich', u'gut'), (u'beson-
ders', u'gut'), (u'gut', u'gefallen'), (u'so', u'gut'),
(u'gut', u'tut'), (u'ganz', u'gut'), (u'ziemlich', u'gut'),
(u'gut', u'hat'), (u'gut', u'rt'), (u'echt', u'gut'),
(u'gut', u'dass'), (u'war', u'gut'), (u'gut', u'mehr'),
(u'gut', u'aber'), (u'ja', u'gut'), (u'zu', u'gut')]
```

Die bisherigen Auswertungen des Tatorts waren sehr grundlegend. Besonders bei der Ermittlung des allgemeinen Tenors auf Twitter zum Tatort genügen einfache Auszählungen oder die reine Betrachtung der Wort-Einbettung nicht. Für verlässliche Bewertungen bedarf es Algorithmen der Sentiment-Analyse, die die Stimmung beziehungsweise die Wertung von Tweets erkennen.

4.3.3.2 Anwendungsbeispiel: Sentiment-Analyse von Tweets zum Franken-Tatort

Laut Pang und Lee (2008) entwickelte sich die Sentiment-Analyse zu einem mittlerweile stark repräsentierten Forschungsgebiet innerhalb der Computerlinguistik mit zahlreichen etablierten Techniken und einer Vielzahl an freien und kostenpflichtig verfügbaren Programmen. Zur automatisierten semantischen Bewertung von Tweets bedarf es immer konnotierter Korpusse, die wertende Begriffe beinhalten und diese (zum Teil) hinsichtlich ihrer Aussagekraft gewichten. Die zur Verfügung stehenden Algorithmen, wie zum Beispiel der *Naive Bayes Klassifikator*, lesen den (vorverarbeiteten) Twitter-Text ein und vergleichen die enthaltenen Begriffe mit Katalogen konnotierter Begriffe. Da die abschließende Bewertung vor allem von der Quantität und Qualität dieser Korpusse abhängt, ist die Wahl des geeigneten Korpus sehr wichtig.

Auf Twitter spezialisierte Sentiment-Korpora existieren jedoch momentan nur für die englische Sprache. Somit besteht nur die Möglichkeit, allgemeine deutsche Korpora zu nutzen oder selbst einen Korpus zu erstellen. In dieser Arbeit wird *SentiWS* (Remus, Quasthoff & Heyer, 2010) der Universität Leipzig verwendet, das kostenlos erhältlich ist. Das Programm besteht vor allem aus zwei Lexika, die 1650 positive und 1818 negativ annotierte deutsche Begriffe zuzüglich ihrer Beugungen beinhalten. Jedes Wort ist mit einem Index versehen, der von -1 (sehr negativ) bis +1 (sehr positiv) reicht.

„Ich fand den # tatort ja ned so arch gut“ (J., 2015) – bereits mit diesem beispielhaft ausgewählten Tweet wird klar, dass aufgrund der Eigenheiten der Twitter-Kommunikation nicht alle verwendeten Begriffe in diesen Lexika enthalten sein werden. Deswegen müssten falsch geschriebene Begriffe und Dialektsprache in reguläres Deutsch umgeschrieben werden. Dies könnte zum größten Teil automatisiert geschehen, sofern ein entsprechender Korpus für deutsche Dialekte und Internetjargon vorläge. Da eine manuelle Erstellung dieses Korpus den Rahmen der Arbeit überschreiten würde, greift diese nur auf den regulären SentiWS-Korpus zurück.

Basierend auf den deutschen Korpus wird im Folgenden mithilfe des überwachten maschinellen Lernens ein Identifikator trainiert, der anhand von Begriffen die emotionale Wertung eines Tweets erkennt (siehe Anhang B). Der hierfür verwendete Naive Bayes Klassifikator (NBK) nutzt die bereits vorhandenen Listen positiver und negativer konnotierter Ausdrücke des Korpus, die um zwei Listen von fiktiven Test-Tweets mit positiven und negativen Ausdrücken erweitert werden. Das Training erfolgt anhand des Korpus und wird schließlich an 3.000 zufällig ausgewählten Begriffen angewendet.

Daraus ermittelt der Naive Bayes Klassifikator 20 der aussagekräftigsten Features. Die Merkmalserkennung (*Feature Extraction*) über ein Testset dient der Verschlankung des Ausgangswortschatzes des Klassifikators für die Gesamtanalyse. Die Reduktion auf die besonders aussagekräftigen Begriffe in Bezug auf die Stimmung eines Tweets minimiert Aufwand und Dauer der Klassifizierung. Durch die Einschränkung auf Features werden zudem auch Störfeatures, die die Qualität der Analyse minimieren, eliminiert. Das Feature Set dient nun als Wörterbuch für die Sentiment-Klassifikation des gesamten Tweet-Korpus. Nach der Erkennung der Stimmung jedes Tweets erfolgt die Ausgabe in ein Diagramm, das Abbildung 17 dargestellt.

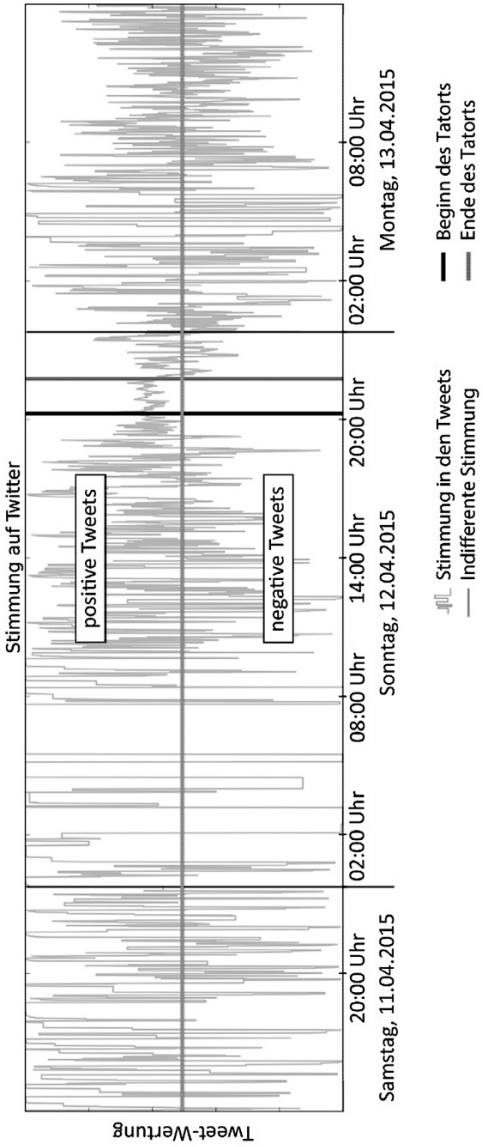

Abbildung 17: Stimmungsverlauf auf Twitter, basierend auf Tweets mit Begriff „tatort“. Eigene Darstellung.

Das Schaubild zeigt den Verlauf der aggregierten Stimmung in Tweets mit dem Begriff „tatort“ über das in Listing 15 gefilterte Datenset (11. bis 13.04.). Hier wird deutlich, dass es im allgemeinen Zeitverlauf keine eindeutige Stimmung in den Tweets gibt. Dagegen erkennt man in der Zeit während der Ausstrahlung der Sendung eine überwiegend positive Stimmung. Möglich ist jedoch auch, dass der NBK schlecht trainiert wurde. Der Trainingsprozess sollte deshalb sorgfältig und idealerweise mehrstufig erfolgen: Nach Erstellung eines Trainingssets folgt die Anwendung auf ein bereits manuell konnotiertes Testset. Das Trainingsset wird dabei solange editiert, bis eine hohe Übereinstimmung zwischen manuellem und automatisch ermitteltem Sentiment vorliegt. Eine ausführliche Sentiment-Analyse über die allgemeine Darstellung der Funktionsweise hinaus übersteigt jedoch den Rahmen dieser Arbeit.

Die beiden Anwendungsbeispiele machen deutlich, dass eine verlässliche automatisierte und computergestützte Inhaltsanalyse von Tweets nicht ohne weitere Schritte der Vorverarbeitung möglich ist. Da die Qualität der Daten entscheidend für die Aussagekraft der verwendeten Algorithmen ist, sollten folglich immer vorgelagerte Filter, Strukturierungen und Korrekturen erfolgen. Dennoch besteht auch nach diesen Schritten immer noch das Problem der fehlenden Berücksichtigung des Tweet-Kontextes. Das Erkennen von Ironie und Sarkasmus ist so nicht möglich. Die Aussagekraft, besonders bei Sentiment-Analysen, ist somit immer eingeschränkt.

Nachdem nun einige Ansätze zum Verarbeiten und Analysieren von Tweets für Forschung präsentiert wurden, folgt nun eine abschließende Betrachtung von Twitter als Quelle wissenschaftlicher Arbeiten. Dabei fließen die Erkenntnisse aus den vorigen Kapiteln mit ein.

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung - Nicht kommerziell 4.0 International Lizenz (http://creativecommons.org/licenses/by-nc/4.0/deed.de) veröffentlicht, welche für nicht kommerzielle Zwecke die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.
Etwaige Abbildungen oder sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende oder der Quellreferenz nichts anderes ergibt. Sofern solches Drittmaterial nicht unter der genannten Creative Commons Lizenz steht, ist eine Vervielfältigung, Bearbeitung oder öffentliche Wiedergabe nur mit vorheriger Zustimmung des betreffenden Rechteinhabers oder auf der Grundlage einschlägiger gesetzlicher Erlaubnisvorschriften zulässig.

5 Twitter als Quelle wissenschaftlicher Analysen

Der Mikroblogging-Dienst Twitter ist seit einiger Zeit eine viel genutzte Quelle für sozial- und wirtschaftswissenschaftliche, aber auch technische Studien. „In addition to being a versatile communications platform to users around the globe, Twitter is also an excellent source of current information“ (Gaffney & Puschmann, 2014, S. 55). Der Dienst vereint also vielseitige Kommunikationsmöglichkeiten mit einer guten Verfügbarkeit von Informationen, wobei letzteres nicht nur für die eigentlichen Nutzer gilt, sondern auch für die Forschung. Dennoch hat Twitter auch zahlreiche Einschränkungen, zum Beispiel hinsichtlich der Qualität und Verlässlichkeit der Daten. Die folgenden Kapitel skizzieren die Vor- und Nachteile von Twitter-Daten für die Wissenschaft und bewerten die Eignung von Twitter als Quelle für wissenschaftliche Analysen anhand mehrerer Aspekte.

5.1 Informationsgehalt

Ein großer Vorteil von Twitter im Vergleich zu anderen sozialen Netzwerken wie Facebook oder MySpace ist die offene Kommunikation. Einzelne Meldungen/Tweets oder ganze Konversationen können durch jeden gesucht und gelesen werden – unabhängig, ob man als Nutzer registriert und mit einer Person befreundet ist. Diese offene Gestaltung schafft einen hohen Grad an Publizität (Jürgens & Jungherr, 2011, S. 205). Dies kann hinsichtlich der häufig geführten Datenschutz-Debatte ein Problem sein, ist jedoch für Forschende von großem Nutzen: Zum einen erhält man Zugriff auf umfassende und ungefilterte Daten, zum anderen müssen (zunächst) keine strengen Datenschutz- und Anonymisierungsauflagen eingehalten werden. Zudem können die Daten detailliert sein: Neben personenbezogenen Informationen, wie Name, Wohnort und Sprache, Followers und Followings, kann über die einzelnen Tweets, Retweets, Mentions und Favorites die Einstellung zu Themen und Nutzern oder die Stimmung abgefragt werden. Ferner stehen diese Informationen – bei Verwendung der frei zugänglichen Twitter APIs – kostenlos zur Verfügung. Beim Sammeln und Verknüpfen dieser Daten über einen

© Der/die Autor(en) 2016
F. Pfaffenberger, *Twitter als Basis wissenschaftlicher Studien*,
DOI 10.1007/978-3-658-14414-2_5

längeren Zeitraum erhält man eventuell umfassende Informationen über das Verhalten, die Stimmung oder Einstellung zu einem Thema.

Aufgrund der Begrenzung der Tweets auf 140 Zeichen und einer einfachen Gestaltung des Dienstes können Informationen schnell an einzelne Personen (One-to-One), eine Gruppe (One-to-Few) oder die Öffentlichkeit (One-to-Many) kommuniziert werden (Williams et al., 2013, S. 385). In Verbindung mit einer relativ hohen Anonymität wird hierdurch auch eine ungezwungene Konversation gefördert, beziehungsweise die Hemmschwelle für eine kritische oder sozial unerwünschte Äußerung zu einem Thema gesenkt. Dies begünstigt nicht nur die Tweet-Wahrscheinlichkeit und -häufigkeit eines Nutzers und somit den Datenumfang für Studien, sondern ermöglicht der Forschung auch Einblicke in die Psyche der Nutzer (Java et al., 2007; Zhao & Rosson, 2009). Zudem beschränkt sich der Inhalt aufgrund der strengen Zeichenbegrenzung auf das Wesentliche: Statt langen Texten sind Nutzer gezwungen, in wenigen Sätzen Meinungen zu äußern oder Informationen zu vermittelt (boyd et al., 2010a).

5.2 Datenstruktur

Die Komprimierung des Inhalts auf konstante Textlängen und die hohe Standardisierung von Tweets durch feste Konventionen der Kommunikation ermöglichen eine einfache, automatisierte Auswertung der Inhalte auf Twitter (Marres & Weltevrede, 2013). Über Hashtags können die wichtigsten Themen erfasst werden, Mentions, Replies und Followings geben Auskunft über die Vernetzung eines Nutzers, Retweets und Favorites zeigen die Zustimmung eines Nutzers zu einer Äußerung oder das Gefallen einer Meldung an (Bollen, Pepe et al., 2011; Pak & Paroubek, 2010). Diese Informationen können schnell und ohne großen Aufwand aus Tweets extrahiert werden, da Twitter sie – wie auch Links zu Webseiten, Videos und Bilder – gesondert als Metadaten übermittelt.

Jedoch ist eine inhaltliche Analyse unter Umständen auch problematisch: Auf Twitter wird, wie in vielen anderen internetbasierten Diensten, die Sprache abgewandelt. Abkürzungen, Neologismen, vermischte Sprachen und ein unvollständiger Satzbau prägen die Konversation im Internet (Bifet & Frank, 2010; Carter et al., 2013; Gottron & Lipka, 2010). Dies kann, wie bereits in Kapitel 4.3.1 erwähnt, eine automatische Inhaltsanalyse durch Programme behindern oder zu Fehlinterpretationen führen. Ebenso ist die Richtigkeit der Nutzerangaben nicht verifiziert: Twitter-Nutzer können falsche Standortangaben machen, eine andere Sprache wählen oder fiktive Namen verwenden (Cheng et al., 2010; Orita & Hada, 2009).

Gerade die Tatsache, dass Tweets vor allem auf Englisch geschrieben werden (Semiocast, 2013), erschwert eine Zuordnung des Nutzers zu einer Nationalität.

Zudem fehlen wichtige, nutzerspezifische Merkmale wie das Alter oder das Geschlecht. Twitter verlangt diese Informationen nicht bei der Registrierung. Dies wiederum behindert eine mögliche Eingrenzung des zu beobachtenden Nutzerkreises für eine Erhebung. Es ist beispielsweise nicht möglich, nur Tweets von Bundesbürgerinnen zur Bundestagswahl zu erfassen, indem nur Tweets aus deutschen Städten in deutscher Sprache mit einem gewissen Hashtag gespeichert werden. Twitter ist eine globale Plattform und ist jederzeit von jedem Ort abrufbar. Somit eignen sich Twitter-Daten auch nur für global bezogene Analysen oder zumindest für solche, bei denen sozidemographische Merkmale, wie Alter, Geschlecht und Nationalität keine Bedeutung haben. Diese Einschränkung kann natürlich minimiert werden, indem Hashtags verwendet werden, die von sich aus nur von lokalem Interesse sind: *#tatort* wird mit hoher Wahrscheinlichkeit vor allem von deutschsprachigen Personen verwendet und entsprechend selten in Asien oder Amerika. Dennoch besteht die Möglichkeit, dass auch Tweets erfasst werden, die für die Analyse keine Bedeutung haben.

In diesem Kontext ist eine Eingrenzung des Datensets auch deshalb schwierig, weil nicht jeder Nutzer zwangsläufig dasselbe Hashtag für ein Thema nutzt, geschweige denn überhaupt ein Hashtag verwendet. Sollen eher diffusere Themen, wie die Meinung zu einer Partei erfasst werden, fällt die Wahl des geeigneten Suchterms schwer. In diesem Fall empfehlen sich mehrere Abfragen mit unterschiedlichen Begriffen (Beispiel CDU: „merkel“, „cdu“, „konservativen“, „christdemokraten“ usw.).

Ein ähnliches Problem stellen Tweets und Replies dar, die zu einem Thema oder einer Aussage geschrieben werden, ohne dabei Begriffe zu enthalten, die eine Zuordnung zu einem Thema ermöglicht. Zum Beispiel: Nutzerin1 schreibt „der tatort ist sowas von langweilig“ und Nutzer2 antwortet: „@Nutzerin1: ich stimme dir zu!“. Ebenso können Aussagen, die Ironie oder Sarkasmus enthalten, aufgrund des nicht ersichtlichen Kontextes falsch interpretiert werden. Diese Zusammenhänge können nur bei einer qualitativen Analyse erfasst werden, die den Kontext von Tweets berücksichtigt.

5.3 Repräsentativität

Twitter macht keine genauen Angaben zu Umfang und Vollständigkeit der durch die APIs übermittelten Daten, wie beispielsweise zu der Repräsentativität des Streaming API Samples. Das Unternehmen weist jedoch zumindest bei den REST APIs darauf hin, dass der Fokus dieser Schnittstellen auf der Übermittlung *relevanter* und *nicht vollständiger* Daten liegt (Twitter, Inc., 2015g). Das mag nicht nur daran liegen, dass Twitter den Zugriff auf vollständige Daten (über Firehose und den eigenen Datenhändler Gnip) als Einnahmequelle nutzt, sondern auch an den technologischen Herausforderungen bei der Bereitstellung von Echtzeit-Daten in einer ständig wachsenden Infrastruktur.

Bei der Betrachtung von Twitter-Daten sollte ferner immer berücksichtigt werden, dass die Nutzerzahl mit etwa 800 Millionen[30] weltweit zwar sehr hoch, die Zahl aktiver Nutzer mit 284 Millionen jedoch deutlich geringer ist, wovon wiederum etwa 30 Prozent US-Amerikaner sind (Semiocast, 2012). Dabei sind auch diese Werte nur eingeschränkt zuverlässig, da auch Semiocast die Herkunft der Accounts mittels der zur Verfügung stehenden Daten schätzen muss. Dennoch bleibt Twitter in vielen Ländern ein Nischen-Medium mit einem elitären Nutzerkreis: 2013 nutzten in Deutschland etwa 7 Prozent Twitter zumindest unregelmäßig, davon wiederum 7 Prozent täglich (Busemann, 2013, S. 398). 53 Prozent der Nutzer war zwischen 14 und 29 Jahren alt.

Dementsprechend ist die Repräsentativität von Twitter-Daten stark eingeschränkt. Norris (2001) spricht im Kontext der verzerrten Nutzung von Online-Massenmedien auch von einem *Digital Divide*. Im Internet bildeten sich demnach vermehrt Netzeliten mit höherem sozioökonomischen Status beziehungsweise formaler Bildung, die neue Medien (wie Twitter) früher und intensiver nutzen sowie besser informiert seien. Die wenigen Studien, die sich mit der Demografie von Twitter-Nutzern beschäftigen, beschreiben den typischen Twitter-Nutzer als relativ jung (19-29 Jahre) mit höherem Bildungsgrad und etwas erhöhtem politischen Interesse im Vergleich zur Gesamtbevölkerung (Duggan, Ellison, Lampe, Lenhart, & Madden, 2015; Gainous & Wagner, 2014; Vaccari et al., 2013).

Somit sollte man immer beachten, dass die Zuverlässigkeit und Repräsentativität von Twitter-Daten eingeschränkt ist und beispielsweise mit Interviews oder Umfragen nicht verglichen werden kann. Nach Jungherr (2015, S. 25) erlaubt

[30] Twitter, Inc. nennt keine offiziellen Zahlen zu registrierten Benutzerkonten, sondern veröffentlicht nur die Zahl aktiver Nutzer. Ältere Schätzungen gehen von etwa 800 Millionen registrierten Konten aus (Edwards, 2013).

Twitter zwar einen guten Einblick über die Veränderung von Absichten und Einstellungen von Nutzern, aber keine verlässlichen Rückschlüsse auf die allgemeine öffentliche Meinung. Eine Verlässlichkeit von Twitter-Inhalten ist nie vollständig gewährleistet: Weder die Echtheit eines (nicht-prominenten) Twitter-Accounts, noch die tatsächliche Meinung eines Nutzers oder die Echtheit einer verbreiteten Information können bestätigt werden (Gayo-Avello, 2012; Metaxas, Mustafaraj, & Gayo-Avello, 2011).

5.4 Datenverfügbarkeit

Neben der mangelnden Repräsentativität von Twitter-Daten ist auch deren eingeschränkte Verfügbarkeit problematisch: Je nach Datenquelle können entweder zukünftige (Streaming API) oder vergangene (REST APIs) Tweets kostenlos abgefragt werden. Zudem gibt es noch eine Vielzahl von Drittanbietern, die ebenfalls unterschiedliche Einschränkungen hinsichtlich Zeitraum und Umfang haben. Letztlich wäre eine Vollständigkeit der Daten nur bei Zugriff auf die Firehose beziehungsweise bei Erwerb von Datenpaketen bei Gnip gewährleistet.

Zudem sind historische Tweets nach 6 bis 9 Tagen nur noch über einzelne Abfragen (Anhand der User- oder Tweet-ID) abrufbar. Bei sehr aufsehenerregenden Ereignissen, wie beispielsweise dem Finale der FIFA Fußball-WM der Männer, ist eine umfassende Erhebung von Tweets mit Hilfe der Streaming API nicht möglich, da sehr schnell das Bandbreiten-Limit von einem Prozent überschritten wird. Da die Suchergebnisse Search API nicht vollständig und zeitlich begrenzt sind, ist es für derartige Ereignisse sinnvoll, Daten nachträglich zu kaufen. Diese Einschränkungen behindern nicht nur die Forschung an sich, sondern erschweren auch eine Reproduzierbarkeit der Ergebnisse, da nicht gewährleistet ist, dass jede Abfrage mit gleichen Parametern identische Daten erzeugt (Gaffney & Puschmann, 2014, S. 65).

Die Unvollständigkeit der Daten erzeugt noch ein weiteres Problem: Ähnlich wie bei der Analyse von wertenden Aussagen ist auch bei rein quantitativer Betrachtung der Kontext für eine Einordnung der Daten notwendig. Ohne das Wissen über das absolute Twitter-Volumen können, wie bereits in Kapitel 4.1.4 erwähnt, nur eingeschränkt Aussagen über die Bedeutung absoluter Zahlen getätigt werden.

5.5 Metriken und Methoden

Obwohl sich die Wissenschaft bereits seit geraumer Zeit mit Twitter beschäftigt, fehlen noch immer allgemein gültige Standards hinsichtlich Metriken aber auch Methoden. Bruns und Stieglitz (2014, S. 69-70) thematisieren den Bedarf an akzeptierten Standards für die quantitative Analyse von Nutzer-Aktivitäten auf Twitter. Diese sollten flexibel ausgelegt sein, um sie bei unterschiedlichste Analyseeinheiten anwenden zu können: Von einzelnen Nutzern bis zu Gruppen, von spezifischen Hashtag-Analysen bis zu Ad-hoc-Analysen des gesamten Datenstroms. Diese Metriken würden die Vergleichbarkeit von Studien und deren Ergebnissen vereinfachen (Bruns & Stieglitz, 2013, S. 3).

Mögliche Bereiche für Metriken wären nach Bruns und Burgess (2012) Hashtags, Nutzer und das Gesamtvolumen. Ein Beispiel wäre demnach das Verhältnis von aktiven zu wenig aktiven Nutzern, die ein Hashtag verwenden, um den Einfluss von sehr aktiven und vernetzten Nutzern bei der Verbreitung von Hashtags zu bestimmen. Hierfür fehlt dann allerdings wiederum die Definition von aktiven Nutzern. Nach der Sichtweise von Twitter, Inc. (2015k) sind Nutzer aktiv, wenn sie mindestens einmal im Monat den Twitter-Account verwenden (zum Lesen oder Schreiben). Huberman et al. (2008) setzten wiederum mindestens zwei Tweets voraus, wobei hier keine Angaben über den Zeitraum gemacht wurden. Cha et al. (2010) sahen eine aktive Nutzung erst ab einer Mindestzahl von 10 Tweets seit Erstellung eines Accounts. Ähnlich diffus ist die Definition von Meinungsführern (*Lead Users*).

Des Weiteren bemängeln Bruns und Stieglitz (2014) das Fehlen allgemeiner Informationen seitens Twitter über das absolute Twitter-Volumen zu einem Zeitpunkt. Zur richtigen Interpretation von Spitzen im Twitter-Volumen und zur Entwicklung sinnvoller Metriken (wie dem Anteil am Gesamtvolumen) bedarf es detaillierter Informationen über die (gesamte) Twitter-Aktivität. Damit könnten Ergebnisse gewichtet und verglichen werden. Problematisch ist hierbei auch, dass Tweet-Datensets, die über einen Suchbegriff erstellt werden, nur Tweets enthalten, in denen der Suchterm vorkommt. Dies wurde bereits im vorigen Kapitel angesprochen. Hier wäre es sinnvoll, wenn Twitter zumindest durch Replies verknüpfte Tweets in die Suchergebnisse mit einbeziehen würde, auch wenn diese den Suchterm im Text nicht enthalten.

Hierfür bedarf es aber einer Öffnung von Twitter für die Wissenschaft. Zwar gibt es bereits mit den *Data Grants* ein Programm für wissenschaftliche Nutzungszwecke (Krikorian, 2014) mit Vollzugriff auf alle erhobenen Twitter-Daten. Der Zugang ist aber auf sehr wenige, von Twitter ausgewählte Projekte beschränkt.

Inwieweit sich in der Zukunft allgemein anerkannte Metriken etablieren und Twitter bei dieser Entwicklung positiv mitwirkt, bleibt offen.

Ähnlich wie bei den Metriken gibt es bisher auch in der Methodik keine Standards: der Erhebungszeitraum, die Suchterm-Logik (z.B. ein oder mehrere Hashtags, mit oder ohne Sprachfilter) und die Analysemethode variieren von Studie zu Studie. Selbst bei relativ strukturierten Vorgängen, wie der Sentiment-Analyse, gibt es eine Vielzahl an Ansätzen – von eigenen Skripten bis zu gängigen Analyse-Tools wie beispielsweise *WordStat* (Williams et al., 2013). Die Tatsache, dass einige Autor/-innen weder die angewandte Analysemethode, noch das Vorgehen zur Erstellung des Datensatzes detailliert beschreiben, erschwert sowohl einen Vergleich, als auch die Reproduktion von Ergebnissen.

5.6 Ethische und rechtliche Aspekte

Neben technischen und wissenschaftlichen Problemen bestehen auch ethische und rechtliche Einschränkungen. Grundsätzlich muss jeder Nutzer – und somit auch Forschende – die rechtlichen Vereinbarungen von Twitter einhalten. Die rechtlichen Rahmenbedingungen verteilen sich auf vier Dokumente: Den allgemeinen Nutzungsbestimmungen *Terms of Service* (Twitter, Inc., 2015a), der *Developer Policy* (Twitter, Inc., 2014a), dem *Developer Agreement* (Twitter, Inc., 2014b) für Entwickler und der Datenschutzrichtlinie *Privacy Policy* (Twitter, Inc., 2015i). Zudem sollten Forschende immer auch die nationalen Datenschutzbestimmungen sowie – bei Verwendung von Drittanbieter-Programmen – weitere individuelle Bestimmungen berücksichtigen.

Fasst man diese Regelungen zusammen, lassen sich einige Einschränkungen für die Twitter-Forschung erkennen: Grundsätzlich darf die Datennutzung den wirtschaftlichen Interessen von Twitter nicht schaden. Twitter verbietet somit generell die Erstellung, Anreicherung und Verbreitung großer Datenbanken mit Tweets – und das auch für nicht-kommerzielle Zwecke (Beurskens, 2014). Dies ist allein Twitter und –im begrenzten Umfang – privilegierten Vertragspartnern vorbehalten. Zudem unterliegen größere Datenbanken mit (angereicherten) Nutzerdaten den jeweiligen nationalen Datenschutz-bestimmungen (Beurskens, 2014, S. 130). Dies erschwert die Weitergabe vorhandener Daten an andere Wissenschaftler/-innen und begrenzt dadurch die Möglichkeiten zur Reproduktion von Ergebnissen.

Es ergibt sich somit ein diffuses Bild: Einerseits stellen Twitter-Nutzer automatisch ihre Tweets der Öffentlichkeit zur Verfügung (dies ist der Sinn von Tweets). Diese Nachrichten können über die interne Suche oder Google gefunden

werden. Jeder Nutzer sollte sich bewusst sein, dass prinzipiell jede Person die persönlich zur Verfügung gestellten Daten lesen kann. Andererseits behält sich Twitter die Rechte an diesen Tweets vor (zur Verarbeitung oder Weitergabe), erstellt aber gleichzeitig im Zuge einer Kooperation mit der amerikanischen *Library of Congress* ein öffentliches, historisches Archiv aller Tweets (Twitter, Inc., 2010). Folglich könnte man auch argumentieren, dass diese Twitter-Daten, die bereits veröffentlicht wurden, problemlos erneut (im Kontext einer Studie) verarbeitet und publiziert werden dürfen.

Da Twitter die Einhaltung seiner Regeln nur schwer kontrollieren kann, empfiehlt Beurskens (2014) eine pragmatische Vorgehensweise: Selbst erstellte Tweet-Sammlungen sollten nie veröffentlicht und der Zugriff auf Daten kontrolliert werden. Wäre eine Weitergabe von Daten für wissenschaftliche Zwecke notwendig, sollten diese immer anonymisiert werden, beispielsweise durch eine Pseudonymisierung oder ein Entfernen der entsprechenden, sensiblen Werte (name, screen_name, id). Eine Veröffentlichung/Weitergabe einzelner Tweets ist jedoch immer erlaubt und wird vor allem durch Medien ständig praktiziert

5.7 Relevanz und Zukunft des Portals

Insgesamt ist Twitter jedoch ein beachtenswertes Medium und eine nahezu unerschöpfliche Datenquelle für vielseitige Forschungsschwerpunkte. Das zeigt die Bandbreite an Studien: Die Meta-Analyse von Williams et al. (2013) analysierte über 1.000 wissenschaftliche Arbeiten, die im Zusammenhang mit Twitter stehen. Diese klassifizierten sie unter anderem in folgende Forschungsbereiche: Wirtschaft (v.a. PR, Marketing), Geografie (Bewegungsmuster, Aufenthaltsort), Gesundheit (Krankheiten), Kommunikation (Interpersonelle Kommunikation, Medien), Notfälle (Katastrophen), Klassifizierungen (Verhaltensmuster) und Sprache (Syntax, Semantik). Puschmann, Bruns, Mahrt, Weller und Burgess (2014, S. 426-427) begründen die Relevanz von Twitter für die Forschung mit der zunehmenden Bedeutung des Mikroblogging-Dienstes. Twitter sei ein globales Phänomen, das stetig an Nutzern und Tweets und somit auch an verwertbaren Daten wächst. Die immer stärkere Einbettung in die Medienökologie (durch Politik, Journalismus, Unternehmen und Privatnutzer) unterstreiche die aufstrebende Rolle in der Gesellschaft und den wachsenden Einfluss auf die Gesellschaft.

Dennoch ist die mittel- bis langfristige Zukunft von Twitter offen. Während die Zahl der monatlich aktiven Nutzer bis 2015 konstant wuchs, sank der Wert im vierten Quartal 2015 erstmals um 2 Millionen auf nun 305 Millionen (Twitter,

Inc., 2016). Zudem verbuchte das Unternehmen in den letzten Jahren erhebliche Verluste: Allein 2014 hatte Twitter einen Netto-Verlust von 578 Millionen US-Dollar (Twitter, Inc., 2015j).

Inwieweit das die zukünftige Architektur von Twitter hinsichtlich Funktionen und Zugriffsmöglichkeiten beeinflusst, ist noch unklar. Möglich wären restriktivere Datenzugänge (dies zeigt sich bereits in der Konzentration auf Gnip als alleinigen Daten-händler) – nicht nur für kommerzielle Nutzer, sondern auch für die Forschung. Zudem plant Twitter eine Ausweitung der Werbeanzeigen durch bessere Vermarktungsfunktionen auf der Plattform (Bragdon, 2015). Anfang 2016 verkündete Twitter mehrere tiefgreifende Veränderungen in der Systematik der Plattform. Zum einen überarbeitet das Unternehmen den Algorithmus zur Darstellung von Tweets im Feed: Anstelle einer streng chronologischen Auflistung von Tweets werden nun Tweets von Followees vorgefiltert und nach individueller Bedeutsamkeit und Popularität an den Anfang positioniert (Jahr, 2016). Diese Vorsortierung ist zunächst nicht verpflichtend, allerdings besteht die Möglichkeit, dass Twitter diese Funktion in absehbarer Zeit für eine lukrativere Positionierung von Werbung nutzen könnte.

Zum anderen wird überlegt, die Limitierung auf 140 Zeichen pro Tweet aufzuheben (Dorsey, 2016). Dies widerspräche der Grundidee von Twitter, Meldungen auf das Wesentliche zu konzentrieren und den Informationscharakter zu stärken. Die geringe Textlänge ist das wichtigste Merkmal der Kommunikation auf Twitter und ein Alleinstellungsmerkmal gegenüber anderen Plattformen, wie Facebook oder Instagram. Und nicht zuletzt ist die Zeichenbegrenzung ein wesentlicher Vorteil für die Textanalyse. Eine Erhöhung des Limits auf 10.000 Zeichen, wie es momentan geplant wird, verändert letztlich nicht nur die Kommunikationsmuster und -inhalte, sondern auch die Möglichkeit zur Datenanalyse.

Beiden Mitteilungen führten zu einer großen Resonanz bei Nutzern und Medien. Über das Hashtag *#RIPTwitter* kritisierten viele Nutzer die Abkehr von den Grundprinzipien des Dienstes (Steiner, 2016). Die langfristige Reaktion der Nutzer (z.B. als Einbruch der Nutzerzahlen) wird sich in der Zukunft zeigen. Letztlich ist das Internet sehr schnelllebig: Regelmäßig entstehen und verschwinden Online-Kommunikationsportale. Möglich wäre also ein ähnlich rapider Bedeutungsverlust von Twitter – auch aufgrund disruptiver Innovationen – wie er bei den sozialen Netzwerken StudiVZ und MySpace nach dem Erfolg von Facebook zu beobachten war.

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung - Nicht kommerziell 4.0 International Lizenz (http://creativecommons.org/licenses/by-nc/4.0/deed.de) veröffentlicht, welche für nicht kommerzielle Zwecke die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.
Etwaige Abbildungen oder sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende oder der Quellreferenz nichts anderes ergibt. Sofern solches Drittmaterial nicht unter der genannten Creative Commons Lizenz steht, ist eine Vervielfältigung, Bearbeitung oder öffentliche Wiedergabe nur mit vorheriger Zustimmung des betreffenden Rechteinhabers oder auf der Grundlage einschlägiger gesetzlicher Erlaubnisvorschriften zulässig.

6 Forschung mit Twitter – abschließende Bewertung

Ziel dieser Arbeit war die Darstellung mehrerer kostenloser Ansätze zum Sammeln, Speichern und Verarbeiten von Tweets. Während Kapitel 4.1 alle Möglichkeiten der Datenabfrage zeigen und gegenüberstellen konnte, verhinderte die Vielfalt an Ansätzen und Programmen sowie die individuelle Zielsetzung eine Darstellung *aller* Methoden zum Speichern und Analysieren. Deshalb beschränkt sich diese Arbeit auf einige, grundlegende und kostenlose Verfahren, mit deren Hilfe die Möglichkeiten und Einschränkungen nähergebracht werden sollten.

Die in dieser Arbeit angesprochenen Vor- und Nachteile für eine wissenschaftliche Nutzung von Twitter ermöglichen keine eindeutige Wertung des Dienstes als Forschungsinstrument. Kapitel 2 zeigte den umfassenden Forschungsstand und die große Bandbreite an Wegen zur Nutzung von Twitter-Daten. Dennoch zeigten sich bereits hier Einschränkungen: Die typische Internet-Sprache mit ihren Charakteristika, wie Abkürzungen, Mehrsprachigkeit und Neologismen, verhindert eine zuverlässige, automatische Inhaltsanalyse. Die eingeschränkte Möglichkeit, Tweets zu vernetzen (etwa bei Konversationen), blendet den für eine inhaltliche Analyse unter Umständen sehr wichtigen Kontext eines Tweets aus.

Der große Datenumfang, die Offenheit in der Kommunikation und das Set an detaillierten Metadaten (Kapitel 3) erlauben dennoch vielseitige Betrachtungswinkel und ermöglichen zudem problemlos auch langfristige Studien. Forschende profitieren auch davon, dass Twitter bereits strukturierte Daten über seine APIs oder seinen Datenhändler Gnip zur Verfügung stellt. Dies vereinfacht und beschleunigt das Filtern, Strukturieren und Analysieren.

Wie in Kapitel 4 dargestellt, stehen dabei mehrere Möglichkeiten der Datensammlung zur Verfügung. Diese unterscheiden sich jedoch grundlegend hinsichtlich Zeithorizont, Datenumfang und Preis, sodass je nach Forschungsabsicht und finanziellem Rahmen die geeignete Datenquelle gewählt werden muss. Hierbei gilt jedoch immer zu beachten, dass eine Vollständigkeit der Daten theoretisch nur beim kostenpflichtigen Datenhändler Gnip gewährleistet werden kann. Alle anderen Zugänge (APIs und Drittanbieter) haben unterschiedliche Einschränkungen,

© Der/die Autor(en) 2016
F. Pfaffenberger, *Twitter als Basis wissenschaftlicher Studien*,
DOI 10.1007/978-3-658-14414-2_6

sodass unter Umständen eine Kombination mehrerer Ansätze zum Sammeln von Tweets notwendig ist.

Zudem gilt bei Twitter, wie bei allen sozialen Online-Diensten, dass die verfügbaren Daten nie repräsentativ und verlässlich sind. Da die Nutzerstruktur kein Abbild der Bevölkerung ist, sind Prognosen für Ereignisse (wie Wahlen) auf dieser Datenbasis nur eingeschränkt möglich. Effekte der Selbstselektion und sozialen Erwünschtheit beschränken zudem die Aussagekraft von Tweets ein. Die Zuverlässigkeit von Aussagen (im Hinblick auf die Äußerung von Meinungen und Informationen) und Meta-Daten (wie Standort und Sprache) kann kaum verifiziert werden. Nicht nur die Identifikation von Twitter-Spam, sondern auch die Erkennung bewusster Propaganda-Tweets wird somit eine zukünftige Herausforderung sein.

Die ebenfalls in Kapitel 4 besprochenen Ansätze zur Datensammlung und-analyse setzen alle grundlegende technische Kenntnisse voraus. Eine sinnvolle Untersuchung von Twitter-Daten erfordert nicht nur eine Reduktion auf wesentliche Daten, sondern auch eine Bereinigung und Vorverarbeitung. Besonders die spezielle Sprache auf Twitter eignet sich zunächst nicht für eine automatisierte Inhaltsanalyse. Deswegen muss der Text zuvor aufwändig bereinigt, gefiltert und unter Umständen mit anderen Informationen angereichert werden. Dennoch ist eine verlässliche, automatisierte Dateninterpretation nicht gewährleistet. Der fehlende Kontext von Tweets macht eine korrekte Bewertung von Aussagen schwer. Der fehlende Mechanismus zum Sammeln von (über Mentions und Replies) verknüpften Tweets führt zu einer ungewollten Beschränkung des Datensatzes auf Tweets, die den jeweiligen Suchterm enthalten.

Für die zukünftige Forschung könnte es von Interesse sein, anerkannte Standards für Metriken und Methoden der Twitter-Forschung zu etablieren. Das Fehlen von Regeln, Normen und Maßzahlen könnte sich auf das noch recht junge Alter der Twitter-Forschung zurückführen lassen. Oftmals sind auch die Anwendungsfälle derart spezifisch, dass Forschende eigene Ansätze beziehungsweise Programme zum Sammeln und Analysieren entwickeln. Im Hintergrund der großen Bandbreite an Verwendungszwecken und Forschungsrichtungen für Twitter-Daten wird es schwer sein, allgemein gültige Verfahren zu konzipieren. Deshalb empfiehlt sich zumindest die Etablierung von Metriken, um eine Vergleichbarkeit der Ergebnisse zu gewährleisten.

Die Vergleichbarkeit und Reproduzierbarkeit wird wiederum durch ethische und vor allem rechtliche Bestimmungen eingeschränkt. Twitter behält sich das Monopol auf „seine“ Daten. Folgt man den Bestimmungen, wäre ein Sammeln

und Anreichern von großen Datensätzen nicht rechtmäßig. Ein praktikabler Mittelweg zwischen der Einhaltung strikter Regeln und der Veröffentlichung für wissenschaftliche Zwecke wäre die Weitergabe pseudonymisierter Daten.

Trotz aller Einschränkungen ist Twitter eine interessante und beachtenswerte Datenquelle. Die offene und schnelle Kommunikation erlaubt unter Umständen detaillierte Einblicke in die Interessen, Meinung und Stimmung von Nutzern. Die Datenerhebung erfolgt dabei nahezu automatisiert. Selbst bei der Verwendung kostenloser Datenquellen erhalten Forschende Zugriff auf eine riesige Datenmenge. Berücksichtigt man alle Einschränkungen und Fallstricke (z.B. bezüglich Datenverfügbarkeit und Vollständigkeit), ergeben sich hinsichtlich Fragestellung und Zeithorizont nahezu unbegrenzte Forschungsmöglichkeiten: Von der Echtzeit-Erkennung von Epidemien oder Unglücken über die Stimmungsanalyse während medialer Großereignisse bis zur Erstellung von Bewegungsprofilen, Stimmungsverläufen oder Interaktionen ausgewählter Nutzer. Inwieweit Twitter auch in Zukunft von wissenschaftlichem Interesse sein wird und kann, vor allem hinsichtlich der Datenstruktur und -verfügbarkeit, hängt dabei vorrangig von den zukünftigen Entscheidungen des Unternehmens ab.

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung - Nicht kommerziell 4.0 International Lizenz (http://creativecommons.org/licenses/by-nc/4.0/deed.de) veröffentlicht, welche für nicht kommerzielle Zwecke die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.
Etwaige Abbildungen oder sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende oder der Quellreferenz nichts anderes ergibt. Sofern solches Drittmaterial nicht unter der genannten Creative Commons Lizenz steht, ist eine Vervielfältigung, Bearbeitung oder öffentliche Wiedergabe nur mit vorheriger Zustimmung des betreffenden Rechteinhabers oder auf der Grundlage einschlägiger gesetzlicher Erlaubnisvorschriften zulässig.

Literaturverzeichnis

Acar, A., & Muraki, Y. (2011). Twitter for crisis communication: lessons learned from Japan's tsunami disaster. *International Journal of Web Based Communities, 7* (3), 392.

Aramaki, E., Maskawa, S., & Morita, M. (Hrsg.). (2011). *Twitter catches the flu: detecting influenza epidemics using Twitter.* Association for Computational Linguistics.

ARD-Werbung Sales & Services GmbH. (2015). *Media Perspektiven Basisdaten 2014. Onlineanwendungen im Zeitvergleich* (Media Perspektiven Basisdaten). Zugriff am 23.07.2015. Verfügbar unter http://www.media-perspektiven.de/fileadmin/user_upload/media-perspektiven/pdf/2015/Basisdaten_2014_komplett_verlinkt.pdf

Benevenuto, F., Magno, G., Rodrigues, T., & Almeida, V. (2010). Detecting spammers on twitter. In *Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*.

Beurskens, M. (2014). Legal questions of Twitter research. In K. Weller (Hrsg.), *Twitter and society* (Digital Formations, Volume 89, S. 123–133). New York: Peter Lang Publishing.

Bifet, A., & Frank, E. (2010). Sentiment knowledge discovery in Twitter streaming data. In B. Pfahringer, G. Holmes & A. Hoffmann (Hrsg.), *Discovery Science* (Lecture Notes in Computer Science, Bd. 6332, S. 1–15). Heidelberg: Springer VS. Verfügbar unter http://dx.doi.org/10.1007/978-3-642-16184-1_1

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python* (1st Ed.). Beijing: O'Reilly.

Boie, J. (2011, 03. Januar). *Das Zwitschern der Weinkönigin.* Zugriff am 22.07.2015. Verfügbar unter http://www.sueddeutsche.de/politik/twitter-und-der-bundespraesident-das-zwitschern-der-weinkoenigin-1.441573

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science, 2* (1), 1–8.

Bollen, J., Pepe, A., & Mao, H. (2011). Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media* (S. 450–453). Menlo Park, California: AAAI Press.

boyd, d., & Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication, 13* (1), 210–230.

boyd, d., Golder, S., & Lotan, G. (2010). Tweet, tweet, retweet: conversational aspects of retweeting on Twitter. In *Proceedings of the 43rd Hawaii International Conference on System Sciences (HICSS-43)* (S. 1–10).

© Der/die Autor(en) 2016
F. Pfaffenberger, *Twitter als Basis wissenschaftlicher Studien*,
DOI 10.1007/978-3-658-14414-2

Bragdon, A. (2015, 02. Juli). *Introducing new audience insights for brands,* Twitter, Inc. Zugriff am 18.07.2015. Verfügbar unter https://blog.twitter.com/2015/introducing-new-audience-insights-for-brands

Bruns, A., & Burgess, J. (2012). Notes towards the scientific study of public communication on Twitter. *Science and the Internet,* 159–169.

Bruns, A., Highfield, T., & Burgess, J. (2013). The Arab Spring and social media audiences: English and arabic Twitter users and their networks. *American Behavioral Scientist, 57* (7), 871–898.

Bruns, A., & Liang, Y. E. (2012). Tools and methods for capturing Twitter data during natural disasters. *First Monday, 17* (4).

Bruns, A., & Moe, H. (2014). Structural layers of communication on Twitter. In K. Weller (Hrsg.), *Twitter and society* (Digital Formations, Volume 89, S. 15–28). New York: Peter Lang Publishing.

Bruns, A., & Stieglitz, S. (2012). Quantitative approaches to comparing communication patterns on Twitter. *Journal of Technology in Human Services, 30* (3-4), 160–185.

Bruns, A., & Stieglitz, S. (2013). Towards more systematic Twitter analysis. Metrics for tweeting activities. *International Journal of Social Research Methodology, 16* (2), 91–108.

Bruns, A., & Stieglitz, S. (2014). Metrics for understanding communication on Twitter. In K. Weller (Hrsg.), *Twitter and society* (Digital Formations, Volume 89, Bd. 89, S. 69–82). New York: Peter Lang Publishing.

Busemann, K. (2013). Wer nutzt was im Social Web? Ergebnisse der ARD/ZDF-Onlinestudie 2013. *Media Perspektiven* (7-8), 391–399. Zugriff am 30.03.2015. Verfügbar unter http://www.ard-zdf-onlinestudie.de/index.php?id=436

Carter, S., Tsagkias, M., & Weerkamp, W. (2011). Semi-supervised priors for microblog language identification. In *Dutch-Belgian information retrieval workshop (DIR 2011).* Amsterdam.

Carter, S., Weerkamp, W., & Tsagkias, M. (2013). Microblog language identification. Overcoming the limitations of short, unedited and idiomatic text. *Language Resources and Evaluation, 47* (1), 195–215.

Cha, M., Haddadi, H., Benevenuto, F., & Gummadi, P. K. (2010). Measuring user influence in Twitter: The million follower fallacy. *ICWSM, 10* (10-17), 30.

Chen, G. M. (2011). Tweet this: A uses and gratifications perspective on how active Twitter use gratifies a need to connect with others. *Computers in Human Behavior, 27* (2), 755–762.

Cheng, Z., Caverlee, J., & Lee, K. (2010). You are where you tweet. In J. Huang, N. Koudas, G. Jones, X. Wu, K. Collins-Thompson & A. An (Hrsg.), *The 19th ACM international conference* (S. 759).

Chodorow, K. (2013). *MongoDB. The definitive guide* (Second edition). Beijing: O'Reilly.

Christensen, C. (2011). Twitter revolutions? Addressing social media and dissent. *The Communication Review, 14* (3), 155–157.

ComScore (2015). *Altersverteilung der aktiven Nutzer von Twitter im Jahr 2014,* comScore. Zugriff am 01.07.2015. Verfügbar unter http://de.statista.com/statistik/daten/studie/77439/umfrage/nutzer-von-twitter-in-ausgewaehlten-altersklassen/

Conover, M. D., Ratkiewicz, J., Francisco, M., Goncalves, B., Menczer, F., & Flammini, A. (2011). Political polarization on Twitter. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media* (S. 89–96). Menlo Park, California: AAAI Press.

Courtois, C., & D'heer, E. (2012). Second screen applications and tablet users: constellation, awareness, experience, and interest. In *Proceedings of the 10th European conference on Interactive tv and video* (S. 153–156).

Davidson, M. (2014, 06. November). *Open sourcing Twitter emoji for everyone,* Twitter, Inc. Zugriff am 01.05.2015. Verfügbar unter https://blog.twitter.com/2014/open-sourcing-twitter-emoji-for-everyone

Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM, 51* (1), 107–113.

DeGeneres, Ellen [TheEllenShow] (2014, 02. März). If only Bradley's arm was longer. Best photo ever. #oscars [Tweet]. Abgerufen von: https://twitter.com/TheEllenShow/status/440322224407314432/photo/1

Diakopoulos, N. A., & Shamma, D. A. (2010). Characterizing debate performance via aggregated twitter sentiment. In *Proceedings of ACM CHI 2010 Conference on Human Factors in Computing Systems*.

Diginomy Pty Ltd. (2015). *Twitonomy,* Diginomy Pty Ltd. Zugriff am 06.07.2015. Verfügbar unter www.twitonomy.com

Dorsey, J. [Jack] (05.01.2016). https://pbs.twimg.com/media/CX_R94ZUkAACYLE.jpg [Tweet]. Abgerufen von: https://twitter.com/jack/status/684496529621557248

Duggan, M., Ellison, N. B., Lampe, C., Lenhart, A., & Madden, M. (2015, 09. Januar). *Social media update 2014*. Pew Research Center. Zugriff am 06.04.2015. Verfügbar unter http://www.pewinternet.org/2015/01/09/social-media-update-2014/

Dusch, A., Gerbig, S., Lake, M., Lorenz, S., Pfaffenberger, F., & Schulze, U. (2015). Post, reply, retweet – Einsatz und Resonanz von Twitter im Bundestagswahlkampf 2013. In C. Holtz-Bacha (Hrsg.), *Die Massenmedien im Wahlkampf. Die Bundestagswahl 2013* (S. 275–294). Wiesbaden: Springer VS.

Earle, P. S., Bowden, D. C., & Guy, M. (2012). Twitter earthquake detection: earthquake monitoring in a social world. *Annals of Geophysics, 54* (6).

Ebersbach, A., Glaser, M., & Heigl, R. (2008). *Social web.* Stuttgart: UTB.

Edwards, J. (2013, 06. November). Twitter's 'dark pool': IPO doesn't mention 651 million users who abandoned Twitter. Business Insider. Zugriff am 06.07.2015. Verfügbar unter: http://www.businessinsider.com/twitter-total-registered-users-v-monthly-active-users-2013-11

Elter, A. (2013). Interaktion und Dialog? Eine quantitative Inhaltsanalyse der Aktivitäten deutscher Parteien bei Twitter und Facebook während der Landtagswahlkämpfe 2011. *Publizistik, 58* (2), 201–220.

Ethority (2014). *Social Media Prisma. Version 6 - German Edition*. Verfügbar unter http://ethority.de/smp/RGB.jpg

Fahrstuhlprofi [Paddy1FCN] (2015, 15. juni). Allmächd na, am Sunndoch is ja scho widda Dräningsaafdakd beim Glubb #FCN [Tweet]. Abgerufen von: https://twitter.com/Paddy1FCN/status/610379763279073281

Gadde, V. (2014, 26. März). *Challenging the access ban in Turkey,* Twitter, Inc. Zugriff am 23.07.2015. Verfügbar unter https://blog.twitter.com/2014/challenging-the-access-ban-in-turkey

Gaffney, D., & Puschmann, C. (2014). Data collection on Twitter. In K. Weller (Hrsg.), *Twitter and society* (Digital Formations, Volume 89, S. 55–67). New York: Peter Lang Publishing.

Gainous, J., & Wagner, K. M. (2014). *Tweeting to power: The social media revolution in American politics:* Oxford University Press.

Gayo-Avello, D. (2012). "I Wanted to predict elections with Twitter and all I got was this lousy paper" – a balanced survey on election prediction using Twitter data. *arXiv preprint arXiv:1204.6441*.

GlobalWebIndex (2015). *Altersverteilung der Nutzer von Facebook weltweit im 4. Quartal 2014,* GlobalWebIndex. Zugriff am 01.07.2015. Verfügbar unter http://de.statista.com/statistik/daten/studie/39471/umfrage/nutzer-von-facebook-nach-alter/

Google, Inc. (2015, 26. Mai). *Translate API,* Google Inc. Verfügbar unter https://cloud.google.com/translate/docs

Gottron, T., & Lipka, N. (2010). A comparison of language identification approaches on short, query-style texts. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell et al. (Hrsg.), *Advances in Information Retrieval* (Lecture Notes in Computer Science, Bd. 5993, S. 611–614). Berlin, Heidelberg: Springer Berlin Heidelberg.

Graham, M., Hale, S. A., & Gaffney, D. (2014). Where in the world Are you? Geolocation andl Identification in Twitter. *The Professional Geographer, 66* (4), 568–578.

Grant, W. J., Moon, B., & Busby Grant, J. (2010). Digital dialogue? Australian politicians' use of the social network tool Twitter. *Australian Journal of Political Science, 45* (4), 579–604.

Halavais, A. (2014). Structure of Twitter. Social and technical. In K. Weller (Hrsg.), *Twitter and society* (Digital Formations, Volume 89, S. 29–41). New York: Peter Lang Publishing.

Halstead, N. (2015, 11. April). *Twitter ends its partnership with DataSift – firehose access expires on August 13, 2015,* Datasift. Zugriff am 17.04.2015. Verfügbar unter http://blog.datasift.com/2015/04/11/twitter-ends-its-partnership-with-datasift-firehose-access-expires-on-august-13-2015

Hawelka, B., Sitko, I., Beinat, E., Sobolevsky, S., Kazakopoulos, P., & Ratti, C. (2014). Geo-located Twitter as proxy for global mobility patterns. *Cartography and Geographic Information Science, 41* (3), 260–271.

Heverin, T., & Zach, L. (2010). Microblogging for crisis communication: examination of Twitter use in response to a 2009 violent crisis in the Seattle-Tacoma, Washington, area. In *Proceedings of the 7th International ISCRAM Conference*.

Hofer-Shall, Z. (2015, 10. April). *Working directly with the Twitter data ecosystem.* Zugriff am 01.07.2015. Verfügbar unter https://blog.gnip.com/twitter-data-ecosystem/

Honeycutt, C., & Herring, S. C. (2009). Beyond microblogging: conversation and collaboration via Twitter. In *Proceedings of the 42nd Hawaii International Conference on System Sciences* (S. 1–10).

Huberman, B. A., Romero, D. M., & Wu, F. (2008). Social networks that matter: Twitter under the microscope. *SSRN Electronic Journal*.

International Organization for Standardization (2015). *Date and time format - ISO 8601,* International Organization for Standardization. Zugriff am 07.07.2015. Verfügbar unter http://www.iso.org/iso/home/standards/iso8601.htm

J. [justingretzer] (2015, 12. April). Ich fand den #Tatort ja ned so arch gut. [Tweet]. Abgerufen von: https://twitter.com/justingretzer/status/587377530035908609

Jahr, M. (2016, 10. Februar). *Never miss important Tweets from people you follow*. Zugriff am: 02.03.2016. Verfügbar unter: https://blog.twitter.com/2016/never-miss-important-tweets-from-people-you-follow

Jansen, B. J., Zhang, M., Sobel, K., & Chowdury, A. (2009). Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology, 60* (11), 2169–2188.

Java, A., Song, X., Finin, T., & Tseng, B. (2007). Why we twitter: understanding the microblogging effect in user intentions and communities. In *WebKDD.* San Jose, CA. Verfügbar unter http://workshops.socialnetworkanalysis.info/websnakdd2007/papers/submission_21.pdf

Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. (2007). Target-dependent Twitter sentiment classification. In *Proceedings of the 45th annual meeting of the Association for Computational Linguistics* (S. 151–160).

Johnson, P. R. & Yang, S. (2009). Uses and gratifications of Twitter: An examination of user motives and satisfaction of Twitter use. In *Annual conference of the Association for Education in Journalism and Mass Communication*.

Jokolove [Klaas_joko] (2015, 01. Juni). Omg freuen mich schon so auf @halligalli sleep&greet bei @damitdasklaas + spontan wenn ich Du wäre 22:10! Uhr -@ProSieben [Tweet]. Abgerufen von: https://twitter.com/Klaas__joko/status/605358311895117826

Jungherr, A. (2015). *Analyzing political communication with digital trace data. The role of Twitter messages in social science research* (Contributions to political science). Switzerland: Springer International Publishing.

Jungherr, A., Jürgens, P., & Schoen, H. (2012). Why the pirate party won the German election of 2009 or the trouble with predictions: a response to Tumasjan, A., Sprenger, T. O., Sander, P. G., & Welpe, I. M. "Predicting elections with Twitter: what 140 characters reveal about political sentiment". *Social Science Computer Review, 30* (2), 229–234.

Jürgens, P., & Jungherr, A. (2011). Wahlkampf vom Sofa aus: Twitter im Bundestagswahlkampf 2009. In E. J. Schweitzer & S. Albrecht (Hrsg.), *Das Internet im Wahlkampf. Analysen zur Bundestagswahl 2009* (S. 201–225). Wiesbaden: VS Verlag für Sozialwissenschaften.

Karger, D. R., & Quan, D. (2004). What would it mean to blog on the semantic web? In S. McIlraith, D. Plexousakis & F. van Harmelen (Hrsg.), *The Semantic Web – ISWC 2004* (Lecture Notes in Computer Science, Bd. 3298, S. 214–228). Berlin, Heidelberg: Springer VS. Verfügbar unter http://dx.doi.org/10.1007/978-3-540-30475-3_16

Kazim, H. (2015, 22. Juli). *Terroranschlag in Suruc: Türkische Provider blockierten Twitter.* Zugriff am 23.07.2015. Verfügbar unter http://www.spiegel.de/netzwelt/netzpolitik/tuerkei-provider-sperren-twitter-nach-anschlag-in-suruc-a-1044782.html

Khondker, H. H. (2011). Role of the new media in the Arab Spring. *Globalizations, 8* (5), 675–679.

Konert, B., & Hermanns, D. (2002). Der private Mensch in der Netzwelt. In R. Weiß & J. Groebel (Hrsg.), *Privatheit im öffentlichen Raum* (S. 415–505). Heidelberg: Springer VS.

Krikorian, R. (2010). *Map of a tweet.* Zugriff am 20.03.2015. Verfügbar unter http://de.scribd.com/doc/30146338/map-of-a-tweet#scribd

Krikorian, R. (2014, 17. April). *Twitter #DataGrants selections,* Twitter, Inc. Zugriff am 13.07.2015. Verfügbar unter https://blog.twitter.com/2014/twitter-datagrants-selections

Kumar, S., Morstatter, F., & Liu, H. (2014). *Twitter data analytics* (SpringerBriefs in Computer Science). New York: Springer.

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is Twitter, a social network or a news media? In M. Rappa, P. Jones, J. Freire & S. Chakrabarti (Hrsg.), *the 19th international conference* (S. 591).

Larsson, A. O., & Moe, H. (2012). Studying political microblogging: Twitter users in the 2010 Swedish election campaign. *New Media & Society, 14* (5), 729–747.

Lasorsa, D. L., Lewis, S. C., & Holton, A. E. (2012). Normalizing Twitter. Journalism practice in an emerging communication space. *Journalism Studies, 13* (1), 19–36.

Liu, I. L., Cheung, C. M., & Lee, M. K. (2010). Understanding Twitter usage: what drive people continue to tweet. In *PACIS 2010 Proceedings*.

Loper, E., & Bird, S. (2002). NLTK: the Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics - Volume 1* (S. 63–70). Philadelphia, Pennsylvania: Association for Computational Linguistics.

Lotan, G., Graeff, E., Ananny, M., Gaffney, D., Pearce, I., & boyd, d. (2011). The Arab Spring | The revolutions were tweeted: information flows during the 2011 Tunisian and Egyptian revolutions. *International Journal Of Communication, 5* (31). Verfügbar unter http://ijoc.org/index.php/ijoc/article/view/1246

Lui, M., & Baldwin, T. (2012). langid. py: An off-the-shelf language identification tool. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.

Lui, M., & Baldwin, T. (2014). Accurate language identification of twitter messages. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)* (S. 17–25).

Mamic, L. I., & Almaraz, I. A. (2014). How the larger corporations engage with stakeholders through Twitter. *International Journal of Market Research, 55* (6), 851.

Marres, N., & Weltevrede, E. (2013). Scraping the social? Issues in live social research. *Journal of Cultural Economy, 6* (3), 313–335.

Marturana, L. (2015, 12. März). *MongoDB Showdown: Aggregate vs Map-Reduce,* Draios, Inc. Zugriff am 22.07.2015. Verfügbar unter https://sysdig.com/mongodb-showdown-aggregate-vs-map-reduce

McCord, M., & Chuah, M. (2011). Spam detection on Twitter using traditional classifiers. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell et al. (Hrsg.), *Autonomic and Trusted Computing* (Lecture Notes in Computer Science, Bd. 6906, S. 175–186). Berlin, Heidelberg: Springer Berlin Heidelberg.

Mendoza, M., Poblete, B., & Castillo, C. (2010). Twitter under crisis. In *SOMA '10 Proceedings of the First Workshop on Social Media Analytics* (S. 71–79).

Metaxas, P. T., Mustafaraj, E., & Gayo-Avello, D. (Hrsg.). (2011). *How (Not) to predict elections*. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom)* (S. 165-171). IEEE.

Millman, K. J., & Aivazis, M. (2011). Python for scientists and engineers. *Computing in Science & Engineering, 13* (2), 9–12.

Mocanu, D., Baronchelli, A., Perra, N., Gonçalves, B., Zhang, Q., Vespignani, A. et al. (2013). The Twitter of babel: mapping world languages through microblogging platforms. *PloS one, 8* (4), e61981.

Morstatter, F., Pfeffer, J., Liu, H., & Carley, K. M. (2013). Is the sample good enough? comparing data from twitter's streaming api with twitter's firehose. *arXiv preprint arXiv:1306.5204*.

Neumann, S. (2013, 22. Dezember). *Dieser Afrika-Tweet kostete eine PR-Agentin den Job.* Zugriff am 17.07.2015. Verfügbar unter http://www.welt.de/vermischtes/article123207732/Dieser-Afrika-Tweet-kostete-eine-PR-Agentin-den-Job.html

Noffke, O. (2013, 21. Dezember). *In 64 Zeichen zur meistgehassten Frau des Internets.* Zugriff am 19.07.2015. Verfügbar unter http://www.stern.de/digital/online/rassistischer-tweet-in-64-zeichen-zur-meistgehassten-frau-des-internets-3642070.html

Norris, P. (2001). *Digital divide: civic engagement, information poverty, and the internet worldwide:* Cambridge University Press.

Orita, A., & Hada, H. (2009). Is that really you? In E. Bertino, T. Gross & K. Takahashi (Hrsg.), *The 5th ACM workshop* (S. 17–20).

Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)* (S. 1320–1326).

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Found. Trends Inf. Retr., 2* (1-2), 1–135.

Papacharissi, Z., & de Fatima Oliveira, M. (2012). Affective news and networked publics: The rhythms of news storytelling on #Egypt. *Journal of Communication, 62* (2), 266–282.

Park, J., Barash, V., Fink, C., & Cha, M. (2013). *emoticon style: interpreting differences in emoticons across cultures*. Verfügbar unter https://www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/view/6132

Parmelee, J. H., & Bichard, S. L. (2012). *Politics and the Twitter revolution. How tweets influence the relationship between political leaders and the public* (Lexington studies in political communication). Lanham, Md: Lexington Books.

Paul, M. J., & Dredze, M. (2011). You are what you tweet: analyzing Twitter for public health. In AAAI Press (Hrsg.), *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media* (S. 265–272). Menlo Park, California.

Perera, R., Anand, S., Subbalakshmi, K. P., & Chandramouli, R. (2010). Twitter analytics: architecture, tools and analysis. In *The 2010 Military Communications Conference* (S. 2186–2191). Piscataway, NJ: IEEE.

Perez, S. (2014, 24. Juni). *Twitter officially launches its "retweet with comment" feature*. Verfügbar unter http://techcrunch.com/2015/04/06/retweetception/

Pilkington, E. (2013, 22. Dezember). *Justine Sacco, PR executive fired over racist tweet, 'ashamed'.* Zugriff am 18.07.2015. Verfügbar unter http://www.theguardian.com/world/2013/dec/22/pr-exec-fired-racist-tweet-aids-africa-apology

Police Fédérale [PolFed_presse] (2015, 22. November). Par sécurité, veuillez respecter le silence radio sur les médias sociaux concernant les opérations de police en cours à #Bruxelles. Merci [Tweet]. Abgerufen von: https://twitter.com/polfed_presse/status/668525939324813312

ProSieben [ProSieben] (2015, 17. Juni). +++EILMELDUNG+++ Eine Ära geht zu Ende. Stefan Raab beendet Ende 2015 seine TV-Karriere. @TVtotal #TVtotal [Tweet]. Abgerufen von: https://twitter.com/prosieben/status/611265301397811200

Puschmann, C., Bruns, A., Mahrt, M., Weller, K., & Burgess, J. (2014). Epilogue. Why we study Twitter. In K. Weller (Hrsg.), *Twitter and society* (Digital Formations, Volume 89, S. 425–432). New York: Peter Lang Publishing.

Remus, R., Quasthoff, U., & Heyer, G. (2010). SentiWS - a publicly available German-language resource for sentiment analysis. In *Proceedings of the 7th International Language Resources and Evaluation (LREC'10)* (S. 1168–1171).

Rogers, K. (2015, 23. November). Twitter cats to the rescue in Brussels lockdown. *The New York Times Magazine.* Zugriff am 01.02.2016. Verfügbar unter http://www.nytimes.com/2015/11/24/world/europe/twitter-cats-to-the-rescue-in-brussels-lockdown.html

Rogers, R. (2014). Debanalising Twitter. the transformation of an object of study. In K. Weller (Hrsg.), *Twitter and society* (Digital Formations, Volume 89, S. IX). New York: Peter Lang Publishing.

Ronson, J. (2015, 12. Februar). How one stupid tweet blew up Justine Sacco's life. *The New York Times Magazine.* Zugriff am 22.07.2015. Verfügbar unter http://www.nytimes.com/2015/02/15/magazine/how-one-stupid-tweet-ruined-justine-saccos-life.html

Roomann-Kurrik, A. (2013, 13. Februar). *Introducing new metadata for Tweets,* Twitter, Inc. Zugriff am 13.07.2015. Verfügbar unter https://blog.twitter.com/2013/introducing-new-metadata-for-tweets

Ross, C., Terras, M., Warwick, C., & Welsh, A. (2011). Enabled backchannel: conference Twitter use by digital humanists. *Journal of Documentation, 67* (2), 214–237.

Runkler, T. A. (2000). Der Datenanalyse-Prozess. In *Information Mining* (Computational Intelligence, S. 1–4). Vieweg+Teubner Verlag. Verfügbar unter https://dx.doi.org/10.1007/978-3-322-89158-7_1

Russell, M. A. (2013). *Mining the social web: Data mining Facebook, Twitter, LinkedIn, Google+, GitHub, and more:* O'Reilly Media.

Sacco, J. [JustineSacco] (2013, 20. Dezember). Going to Africa. Hope I don't get AIDS. Just kidding. I'm white! [Tweet].

Sakai, T., Okazaki, M., & Matsuo, Y. (2010). Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web* (S. 851–860). New York: ACM.

Sang, E. T. K., & Bos, J. (2012). Predicting the 2011 dutch senate election results with twitter. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (S. 53–60). Stroudsburg, PA: ACL.

Sanner, M. F. (1999). Python: a programming language for software integration and development. *J Mol Graph Model, 17* (1), 57–61.

Scanfeld, D., Scanfeld, V., & Larson, E. L. (2010). Dissemination of health information through social networks: twitter and antibiotics. *American journal of infection control, 38* (3), 182–188.

Seifert, S. [RegSprecher] (2015, 11. März). #Japan: Our Embassy in #Tokyo commemorates the victims of #Quake, #Tsunami + #NuclearDisaster 4 years ago. #311Day [Tweet]. Abgerufen von: https://twitter.com/GermanyDiplo/status/575676432262774784

Semiocas. (2012). *Semiocast — Twitter reaches half a billion accounts — More than 140 millions in the U.S.,* Semiocast. Zugriff am 30.03.2015. Verfügbar unter http://semiocast.com/en/publications/2012_07_30_Twitter_reaches_half_a_billion_accounts_140m_in_the_US

Semiocast (2013). *Language data reveals Twitter's global reach,* Semiocast. Zugriff am 30.03.2015. Verfügbar unter http://www.technologyreview.com/graphiti/522376/the-many-tongues-of-twitter/

Signorini, A., Segre, A. M., & Polgreen, P. M. (2011). The use of Twitter to track levels of disease activity and public concern in the U.S. during the influenza A H1N1 pandemic. *PloS one, 6* (5), e19467.

SimilarWeb. (2014). *Woher die Twitter-User kommen. Verteilung des Traffics auf Twitter.com im Februar 2014 nach Ländern,* SimilarWeb. Zugriff am 02.07.2015. Verfügbar unter http://de.statista.com/infografik/2040/verteilung-des-traffics-auf-twittercom-im-februar-2014-nach-laendern/

Southall, A. (2013, 20. Dezember). *A Twitter message about aids, followed by a firing and an apology.* Zugriff am 18.07.2015. Verfügbar unter http://thelede.blogs.nytimes.com/2013/12/20/a-twitter-message-about-aids-africa-and-race/

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., & Demirbas, M. (2010). Short text classification in twitter to improve information filtering. In F. Crestani, S. Marchand-Maillet, H.-H. Chen, E. N. Efthimiadis & J. Savoy (Hrsg.), *Proceeding of the 33rd international ACM SIGIR conference* (S. 841–842).

Steiner, A. (10.02.2016). Algorithmische Timeline. Twitter sortiert seine Tweets neu. *Frankfurter Allgemeine Zeitung Online*. Abgerufen am: 23.02.2016. Verfügbar unter: http://www.faz.net/aktuell/wirtschaft/netzwirtschaft/twitter-fuehrt-die-algorithmische-timeline-ein-trotz-dementi-14062730.html

Stelter, B. (2013, 22. Dezember). *Company parts ways with PR exec after AIDS in Africa tweet.* Zugriff am 18.07.2015. Verfügbar unter http://edition.cnn.com/2013/12/21/us/sacco-offensive-tweet/

Stone, B. (2009). *Project retweet: phase one,* Twitter, Inc. Zugriff am 15.03.2015. Verfügbar unter https://blog.twitter.com/2009/project-retweet-phase-one

Suh, B., Hong, L., Pirolli, P., & Chi, E. H. (2010). Want to be retweeted? Large scale analytics on factors impacting retweet in Twitter network. In *IEEE Second International Conference on Social Computing (SocialCom)* (S. 177–184). Minneapolis, Minnesota: IEEE.

Thimm, C., Einspänner, J., & Dang-Anh, M. (2012). Twitter als Wahlkampfmedium. *Publizistik, 57* (3), 293–313.

Trelle, T. (2014). *MongoDB. Der praktische Einstieg* (1. Aufl). Heidelberg: dpunkt.

Tugores, A., & Colet, P. (2013). Mining online social networks with Python to study urban mobility. In P. de Buyl & N. Varoquaux (Hrsg.), *Proceedings of the 6th European Conference on Python in Science (EuroSciPy 2013)* (S. 21–28). Verfügbar unter http://arxiv.org/pdf/1404.6966

Tumasjan, A., Sprenger, T. O., Sandner, P. G., & Welpe, I. M. (2010). Election forecasts with Twitter: How 140 characters reflect the political landscape. *Social Science Computer Review,* 0894439310386557.

Tweet Archivist, Inc. (2015a). *Tweet Archivist,* Tweet Archivist Inc. Zugriff am 11.07.2015. Verfügbar unter https://de.tweetarchivist.com/

Tweet Archivist, Inc. (2015b). *Häufige Fragen,* Tweet Archivist, Inc. Zugriff am 11.07.2015. Verfügbar unter https://de.tweetarchivist.com/about/faq

Twitter, Inc. (2010, 14. April). *Tweet preservation,* Twitter, Inc. Zugriff am 16.07.2015. Verfügbar unter https://blog.twitter.com/2010/tweet-preservation

Twitter, Inc. (2014a). *Entwicklerrichtlinie,* Twitter, Inc. Zugriff am 15.07.2015. Verfügbar unter https://dev.twitter.com/de/overview/terms/policy

Twitter, Inc. (2014b). *Entwicklervertrag,* Twitter, Inc. Zugriff am 15.07.2015. Verfügbar unter https://dev.twitter.com/de/overview/terms/agreement

Twitter, Inc. (2015, 11. Juni). *Removing the 140 character limit from direct messages,* Twitter, Inc. Zugriff am 01.07.2015. Verfügbar unter https://twittercommunity.com/t/removing-the-140-character-limit-from-direct-messages/41348

Twitter, Inc. (2015a). *Allgemeine Geschäftsbedingungen von Twitter,* Twitter, Inc. Zugriff am 15.07.2015. Verfügbar unter https://twitter.com/tos?lang=de

Twitter, Inc. (2015b). *Anzahl der monatlich aktiven Nutzer von Twitter vom 1. Quartal 2010 bis 1. Quartal 2015 (in Millionen),* Twitter, Inc. Zugriff am 01.07.2015. Verfügbar unter http://de.statista.com/statistik/daten/studie/232401/umfrage/monatlich-aktive-nutzer-von-twitter-weltweit-zeitreihe/

Twitter, Inc. (2015c). *GET search/tweets. Parameters,* Twitter, Inc. Zugriff am 06.07.2015. Verfügbar unter https://dev.twitter.com/rest/reference/get/search/tweets

Twitter, Inc. (2015d). *GET statuses/user_timeline,* Twitter, Inc. Zugriff am 20.06.2015. Verfügbar unter https://dev.twitter.com/rest/reference/get/statuses/user_timeline

Twitter, Inc. (2015e). *POST statuses/filter,* Twitter, Inc. Zugriff am 23.03.2015. Verfügbar unter https://dev.twitter.com/streaming/reference/post/statuses/filter

Twitter, Inc. (2015f). *Streaming API request parameters,* Twitter, Inc. Zugriff am 12.04.2015. Verfügbar unter https://dev.twitter.com/streaming/overview/request-parameters#track

Twitter, Inc. (2015g). *The Search API,* Twitter, Inc. Zugriff am 24.03.2015. Verfügbar unter https://dev.twitter.com/rest/public/search

Twitter, Inc. (2015h). *The Search API. Query operators,* Twitter, Inc. Zugriff am 13.04.2015. Verfügbar unter https://dev.twitter.com/rest/public/search

Twitter, Inc. (2015i). *Twitter Datenschutzrichtlinie,* Twitter, Inc. Zugriff am 15.07.2015. Verfügbar unter https://twitter.com/privacy?lang=de

Twitter, Inc. (2015j, 05. Februar). *Twitter reports fourth quarter and fiscal year 2014 results*, San Francisco, Kalifornien. Zugriff am 18.07.2015. Verfügbar unter http://files.shareholder.com/downloads/AMDA-2F526X/3927982790x0x807274/82E8F435-FC2C-4470-88D3-AF94E7BB059E/2014_Q4_Earnings_Release.pdf

Twitter, Inc. (2015k, 31. März). *Twitter usage / company facts,* Twitter, Inc. Zugriff am 01.07.2015. Verfügbar unter https://about.twitter.com/company

Twitter, Inc. (2016, 10. Februar). *Twitter Q4 and fiscal year 2015 shareholder letter*, San Francisco, Kalifornien. Zugriff am 01.02.2016. Verfügbar unter http://files.shareholder.com/downloads/AMDA-2F526X/1152561686x0x874448/7F88ED74-4727-4B42-8568-60B0C0DA92C7/Q4_15_Shareholder_Letter.pdf

Unicode, Inc. (2015). *Full emoji data,* Unicode, Inc. Zugriff am 15.07.2015. Verfügbar unter http://unicode.org/emoji/charts/full-emoji-list.html

United States Securities and Exchange Commission [SEC] (2014, 30. Juni). *Twitter, Inc. Quarterly report persuant to section 13 or 15(d) of the Securities Exchange Act of 1934.* Zugriff am 01.07.2015. Verfügbar unter http://www.sec.gov/Archives/edgar/data/1418091/000156459014003474/twtr-10q_20140630.htm?_ga=1.264558604.2072063068.1396271368

Vaccari, C., Valeriani, A., Barberá, P., Bonneau, R., Jost, J. T., Nagler, J. et al. (2013). Social media and political communication: a survey of Twitter users during the 2013 Italian general election. *Rivista italiana di scienza politica, 43* (3), 381–410.

Varoufakis, Y. (2015, 05. Juli). *Minister No More!* Zugriff am 22.07.2015. Verfügbar unter https://twitter.com/yanisvaroufakis/status/617928716262486016?lang=de

Vieweg, S., Hughes, A. L., Starbird, K., & Palen, L. (2010). Microblogging during two natural hazards events. In *ACM Conference on Human Factors in Computing Systems CHI2010* (S. 1079–1088). Atlanta, Georgia: ACM.

Wang, H., Can, D., Kazemzadeh, A., Bar, F., & Narayanan, S. (2012). A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics* (S. 115–120). Stroudsburg, PA: ACL

Weil, K. (2014). *Coming soon to Twitter.* Zugriff am 15.03.2015. Verfügbar unter https://blog.twitter.com/2014/coming-soon-to-twitter

Weller, K. (2014). What do we get from Twitter—and what not? a close look at Twitter research in the social sciences. *Knowledge Organization, 41* (3), 238–248.

Weller, K., Bruns, A., Burgess, J., Mahrt, M., & Puschmann, C. (2014). Twitter and society: an introduction. In K. Weller (Hrsg.), *Twitter and society* (Digital Formations, Volume 89, S. xxix). New York: Peter Lang Publishing.

Wendling, M. (2015, 17. November). How the Paris attacks unfolded on social media. BBC trending. Zugriff am 25.02.2016. Verfügbar unter: http://www.bbc.com/news/blogs-trending-34836214

Williams, S. A., Terras, M. M., & Warwick, C. (2013). What do people study when they study Twitter? Classifying Twitter related academic papers. *Journal of Documentation, 69* (3), 384–410.

Wiltshire, L. (2014, 14. Juli). *The roar of the crowd for the #WorldCupFinal,* Twitter, Inc. Zugriff am 03.07.2015. Verfügbar unter https://blog.twitter.com/2014/the-roar-of-the-crowd-for-the-worldcupfinal

Zhang, X., Fuehres, H., & Gloor, P. A. (2011). Predicting stock market indicators through Twitter "I hope it is not as bad as I fear". *Procedia - Social and Behavioral Sciences, 26,* 55–62.

Zhao, D., & Rosson, M. B. (2009). How and why people twitter: the role that micro-blogging plays in informal communication at work. In *Proceedings of the ACM 2009 international conference on Supporting group work* (S. 243–252). New York: ACM.

Anhang A – Objekte und Eigenschaften der Twitter APIs

A.1 Wichtige User-bezogene Datenfelder

FELD	BESCHREIBUNG
CREATED_AT	Zeitpunkt der Accounterstellung
DESCRIPTION	Die nutzerspezifische Account-Beschreibung
ENTITIES	→ siehe Anhang A.3 Entites
FAVOURITES_COUNT	Anzahl an Tweets, die dieser User mit einem Favorite versehen hat
FOLLOWERS_COUNT	Anzahl an Followers, die dieser Account momentan hat
FRIENDS_COUNT	Anzahl an Nutzer/-innen, denen dieser Account folgt
ID / ID_STR	Account ID (einzigartig)
LANG	Eingestellte Account-Sprache
LOCATION	Definierter Account-Standort (z.B. Berlin)
NAME	"Echter" Name des Accountinhabers
SCREEN_NAME	Nickname (z.B. @user123)
STATUSES_COUNT	Anzahl an verfassten Tweets bis zu diesem Zeitpunkt
TIME_ZONE	Definierte Zeitzone

Für eine vollständige Liste siehe https://dev.twitter.com/overview/api/users.

© Der/die Autor(en) 2016
F. Pfaffenberger, *Twitter als Basis wissenschaftlicher Studien*,
DOI 10.1007/978-3-658-14414-2

A.2 Wichtige Tweet-bezogene Datenfelder

FELD	BESCHREIBUNG
CREATED_AT	Zeitpunkt der Tweet-Erstellung
ENTITIES	→ siehe Anhang A.3 Entites
FAVOURITES_COUNT	Anzahl an momentanen Favorites durch andere Nutzer
ID / ID_STR	ID des Tweets (einzigartig)
IN_REPLY_TO_SCREEN_NAME	Bei Retweet: screen_name des ursprünglichen Nutzers, sonst: *null*
IN_REPLY_TO_STATUS_ID	Bei Retweet: ID des ursprünglichen Tweets, sonst: *null*
IN_REPLY_TO_USER_ID	Bei Retweet: ID des ursprünglichen Nutzers, sonst: *null*
LANG	Erkannte Sprache des Tweets
PLACE	Ermittelter Ort des Tweets
RETWEET_COUNT	Anzahl momentaner Retweets
RETWEETED_STATUS	Bei Retweet: Entities des ursprünglichen Tweets, sonst: *null*
TEXT	Twitter-Text

Für eine vollständige Liste siehe https://dev.twitter.com/overview/api/tweets.

A.3 Wichtige Entities eines Tweets

FELD	BESCHREIBUNG
HASHTAGS	Hashtags mit Positionsangabe innerhalb des Textes (Anfang und Ende)
MEDIA	Eingebettete Bilder und Videos
URLS	Eingebettete Links in Kurz- (t.co) und Langform
USER_MENTIONS	Eingebettete Mentions mit `name`, `screen_name` und ID

Für eine vollständige Liste siehe https://dev.twitter.com/overview/api/entities

A.4 Einschränkungen der REST APIs

TITLE	RESOURCE FAMILY	USER AUTH	APP AUTH
GET APPLICATION/RATE_LIMIT_STATUS	application	180	180
GET FAVORITES/LIST	favorites	15	15
GET FOLLOWERS/IDS	followers	15	15
GET FOLLOWERS/LIST	followers	15	30
GET FRIENDS/IDS	friends	15	15
GET FRIENDS/LIST	friends	15	30
GET FRIENDSHIPS/SHOW	Friendships	180	15
GET LISTS/LIST	lists	15	15
GET LISTS/MEMBERS	lists	180	15
GET LISTS/MEMBERS/SHOW	lists	15	15
GET LISTS/MEMBERSHIPS	lists	15	15
GET LISTS/OWNERSHIPS	lists	15	15
GET LISTS/SHOW	lists	15	15
GET LISTS/STATUSES	lists	180	180
GET LISTS/SUBSCRIBERS	lists	180	15
GET LISTS/SUSCRIBERS/SHOW	lists	15	15
GET LISTS/SUBSCRIPTIONS	lists	15	15
GET SEARCH/TWEETS	search	180	450
GET STATUSES/LOOKUP	statuses	180	60
GET STATUSES/RETWEETERS/IDS	statuses	15	60
GET STATUSES/RETWEETS/:ID	statuses	15	60
GET STATUSES/SHOW/:ID	statuses	180	180
GET STATUSES/USER_TIMLINE	statuses	180	300
GET TRENDS/AVAILABLE	trends	15	15
GET TRENDS/CLOSEST	trends	15	15
GET TRENDS/PLACE	trends	15	15
GET USERS/LOOKUP	users	180	60
GET USERS/SHOW	users	180	180

Siehe auch https://dev.twitter.com/rest/public/rate-limits

Anhang B – Programmcode zur Inhaltsanalyse des Franken-Tatorts aus Kapitel 4.3.3.2

```
from pymongo import MongoClient as MC
import datetime, nltk
import pandas as p
import matplotlib.pyplot as mpl
from pandas.tseries.resample import TimeGrouper
from pandas.tseries.offsets import DateOffset
from nltk import FreqDist
from nltk.corpus import stopwords
from nltk.collocations import *


# Verbindung zur MongoDB
def mdb_conn(host, port, username, password, db):
 mongo_uri = "mongodb://{0}:{1}@{2}:{3}/{4}".format(
             username, password, host, port, db)
 conn = MC(mongo_uri)
    return conn[db]

def mdb_reader(db, collection, query={}, host="<Host-IP>",
               port=27017, username="<DB-Benutzer>",
               password="<Passwort>", no_id=True):
    db = mdb_conn(host=host, port=port, username=username,
                  password=password, db=db)
    cursor = db[collection].find(query)
    frame =  p.DataFrame(list(cursor))
    if no_id:
        del frame["_id"]
    return frame

dbase = "tatort"
coll = "dt_tweets"


# Korpus-Reader für Collection
tweetkorp = mdb_reader(dbase, coll)
```

© Der/die Autor(en) 2016
F. Pfaffenberger, *Twitter als Basis wissenschaftlicher Studien*,
DOI 10.1007/978-3-658-14414-2

```
# Plotte Tweets nach Zeit
tweetkorp.set_index("created_at", inplace=True)
tweetkorp.index = tweetkorp.index.tz_localize('GMT').
                  tz_convert('CET')
tweetkorp.index.name = "Zeit"
tweetkorp.index
tweetkorp.tail(10)
hashtag = "Tatort"

tweetsperminute = tweetkorp['text'].resample('1t',
                                             how='count')
mpl.figure(figsize=(16,6))
tweetsperminute.plot()
mpl.grid(None)
mpl.savefig("tweetsperminute")

#Tokenisierung
stop_en = stopwords.words('english')
stop_de = stopwords.words('german')
text = tweetkorp[u'text']

stop_en = stopwords.words('english')
stop_de = stopwords.words('german')
customstopwords = ["tatort", "warum", "dass", "immer",
                   "schon", "gleich", "gerade", "jetzt",
                   "mal", "heute", "erst", "macht",
                   "eigentlich", "gibt", "gar", "beim",
                   "ganz", "wer", "mehr", "wohl"]
tokens = []
sentences = []
for txt in text.values:
    sentences.append(txt.lower())
    tokens.extend([t.lower().encode('utf-8').strip(":,.!?")
for t in txt.split()])

# Wortfilter
mentions = [wrd for wrd in tokens if wrd.startswith("@")]
hashtags = [wrd for wrd in tokens if wrd.startswith("#")]
links = [wrd for wrd in tokens if wrd.startswith("www") or
         wrd.startswith("http")]
```

```
sel_tokens = [wrd for wrd in tokens \
                  if not wrd in (customstopwords and
                                 stop_en and stop_de and
                                 mentions and hashtags and
                                 links) \
                  and wrd.isalpha() \
                  and not len(wrd)<3]

# Wortfilter 2 (ohne Entities)
sel_tokens2 = [wrd for wrd in tokens \
                  if wrd.isalpha() \
                  and not wrd in (mentions and links)]

# Plotte Top 20 Wörter
freqdist = nltk.FreqDist(sel_tokens)
freqdist.plot(20)

# Konkordanzen für "gut"
tweettokens = nltk.wordpunct_tokenize(unicode(sentences).
                                      encode('utf-8'))
rawtweettext = nltk.Text(tweettokens)
rawtweettext.concordance("gut")

# Kollokationen
tweettext = nltk.Text(sel_tokens2)
tweettext.collocations()

# Bigramme, die "gut" enthalten
bigram_measures = nltk.collocations.BigramAssocMeasures()
gut_filter = lambda *wrd: "gut" not in wrd
bigrams = BigramCollocationFinder.from_words(tweettext)
bigrams.apply_freq_filter(10)
bigrams.apply_ngram_filter(gut_filter)
print bigrams.nbest(bigram_measures.likelihood_ratio, 20)


# Sentiment-Analyse - NBK trainieren
training_set=[]
import csv
posWrds = csv.reader(open("<Pfad>/sentiws_pos.txt", 'rb'),
                     delimiter='|')
training_set.extend([(pos[0].lower(), 'positive') for pos in
                     posWrds])
negWrds = csv.reader(open("<Pfad>/sentiws_neg.txt", 'rb'),
                     delimiter='|')
```

```
training_set.extend([(neg[0].lower(), 'negative') for neg in
                     negWrds])

posTweet = ["Das war ein guter", \
            u"super quote fürn Franken-Dadord", \
            "Spitze", \
            "ich mochte den tatort gestern", \
            "bassd scho", \
            "Guter", \
            "gelungener Start", \
            "spannend bis zum schluss", \
            "bester tatort seit langem", \
            "top", \
            "Ein skurriler tatort heut #ilike", \
            "erstaunlich guter", \
            u"Wirklich überraschend dieser", \
            "Grandios"]

training_set.extend([(posTweets.lower(), 'positiv') for
                     posTweets in posTweet])

negTweet = ["Langweilig", \
            u"Blöder", \
            u"einfach lächerlich! umschalten", \
            "er soll sterben", \
            "laien-theater", \
            u"tatort war echt enttäuschend", \
            "absolute einschlafhilfe", \
            "abschalten bei dem mist", \
            u"absehbar, wer der mörder ist", \
            "was will dieser justin bieber", \
            u"ich hab so ein ungutes gefühl", \
            "justin bieber", \
            "so ein praktikant mit der waffe", \
            "was will dieser bubi", \
            u"Warum tragen die frauen im heutigen #tatort \
              so miese perücken?", \
            u"die Musik stört", \
            "Ich schalte um"]

training_set.extend([(negTweets.lower(), 'negativ') for
                     negTweets in negTweet])
```

```
samples = freqdist.keys()[:3000]

# Feature-Extraktion
all_words = nltk.FreqDist(w.lower() for w in tweettokens)
samples = list(all_words)[:2000]

def tweet_features(tweet):
    features={}
    for word in samples:
        features["contains({})".format(word)] =
                (word in tweet)
    return features

training = [(tweet_features(word), sentiment) for (word,
            sentiment) in training_set]
classifier = nltk.NaiveBayesClassifier.
            train(training)
classifier.show_most_informative_features(14)

positivTweets = []
negativTweets = []

for twt in tweetkorp.text:
    twts = classifier.classify(tweet_features(twt))
    if twts=='positiv':
        positivTweets.append(twt)
    else:
        negativTweets.append(twt)

def classifytweet(dataframe):
    return classifier.classify(tweet_features(
                                dataframe.text))

tweetkorp['sentiment'] = tweetkorp.apply(classifytweet,
axis=1)

# Plotte Stimmung auf Zeit
mpl.figure(figsize=(20,6))
pd.ewma(tweetkorp.sentiment.str.match('positiv').
        resample('1t'), 2).plot(color='#aaaaaa',
        linewidth=1.5)
mpl.ylim(0,1)
mpl.grid(None)
mpl.axhline(0.5)
```

```
mpl.axvline(datetime(2015, 4, 12, 18, 15, 0, 0), color='r',
            linewidth=4.0)
mpl.axvline(datetime(2015, 4, 12, 19, 45, 0, 0), color='r',
            linewidth=4.0)
mpl.savefig('sentiment-%s.png' % hashtag,
            box_inches='tight', dpi=300)
```