Big Data

Dominic Nyhuis

1 Einführung

Zum Abschluss und als Ausblick auf aktuelle und künftige Entwicklungen wird in diesem Band das Thema *Big Data* angesprochen, welches für die Sozialwissenschaften zunehmend an Bedeutung gewinnt (Monroe 2013). Um uns der Frage zu nähern, welcher Stellenwert *Big Data* mittlerweile im Forschungsalltag zukommt und wie wir die einschlägigen Techniken verwenden können, ist es hilfreich, uns zunächst mit den wesentlichen Merkmalen des wenig geliebten *Big Data*-Begriffs zu beschäftigen.

Das naheliegende erste Distinktionsmerkmal ist bereits im Begriff *Big Data* angelegt: die Größe der zu analysierenden Daten. Verglichen mit Datensätzen der konventionellen sozialwissenschaftlichen Forschung kommen heute Datensätze in völlig anderen Größenordnungen vor. Denken Sie etwa an eine breit angelegte Umfrage mit mehreren Tausend Teilnehmern wie die Allgemeine Bevölkerungsumfrage der Sozialwissenschaften (ALLBUS) oder den European Social Survey (ESS). Solche Datensätze gehören zu dem Größten, was in der traditionellen Forschung Verwendung findet. Entsprechende Datenmengen werden durch Studien zu

D. Nyhuis (✉)
Leibniz Universität Hannover, Institut für Politikwissenschaft, Hannover, Deutschland
E-Mail: d.nyhuis@ipw.uni-hannover.de

M. Tausendpfund (Hrsg.), *Fortgeschrittene Analyseverfahren in den Sozialwissenschaften*, Grundwissen Politik,
https://doi.org/10.1007/978-3-658-30237-5_12

sozialen Netzwerken problemlos in den Schatten gestellt, wo schnell Datensätze mit Millionen Datenpunkten anfallen (King et al. 2013, 2017; Beauchamp 2017).

Dennoch wird der Begriff *Big Data* in den Sozialwissenschaften aus guten Gründen recht kritisch gesehen, da er häufig auch dann verwendet wird, wenn die Daten kaum das Etikett „groß" verdienen. Gerade im Vergleich zu „echten" *Big Data*-Anwendungen wird deutlich, dass sich die Sozialwissenschaften nur selten mit wirklich großen Daten beschäftigen. Zur Einordnung mag es helfen, wenn wir unsere Bemühungen vor dem Hintergrund sehen, in welchen Mengen und in welcher Geschwindigkeit die großen Online-Plattformen wie *YouTube* oder *Amazon* Daten verarbeiten. Selbst die Datenmengen, die beispielsweise von den Sensoren eines einzigen autonomen Fahrzeugs produziert und in Echtzeit ausgewertet werden müssen, übertreffen in Größe viele sozialwissenschaftliche Anwendungen.

Darüber hinaus gilt, dass die Wahrnehmung von großen Daten von den verfügbaren Rechen- und Speicherkapazitäten abhängt. In diesem Sinne verschiebt sich das Verständnis von *Big Data* mit jedem vergehenden Jahr. Während moderne Mobiltelefone aus dem Blickwinkel der 1960er-Jahre mit unerhört großen Daten hantieren, mag den heute noch großen Daten dieser Status schon in wenigen Jahren abhandenkommen. Grundsätzlich kann deshalb der folgende Maßstab gelten – wenn Sie Ihre Daten auf einer externen Festplatte speichern können, dann haben Sie es vermutlich nicht mit *Big Data* zu tun.

Um die einsetzende Ernüchterung aber direkt wieder einzufangen, sei bemerkt, dass derlei Daten – *Medium Data*™ – sozialwissenschaftlichen Interessen viel eher entsprechen. Die aus heutiger Sicht wirklich großen Forschungsdaten bringen Probleme ganz eigener Art mit sich und erfordern insbesondere auch die Auseinandersetzung mit gezielten Software-Lösungen, nur um den anfallenden Datenmengen Herr zu werden. Insofern ist der Vorteil eines mittelgroßen und vor allem abgeschlossenen Datensatzes, dass weniger hohe technische Hürden überwunden werden müssen, um sich mit den Daten auseinandersetzen zu können. Schließlich ist unser Anliegen die Datenauswertung, während die Datensammlung und -verarbeitung allein Mittel zum Zweck sind.

Ein zweites Merkmal von *Big Data* in den Sozialwissenschaften ist, dass entsprechende Daten häufig als Nebenprodukt einer nicht mit dem Ziel der sozialwissenschaftlichen Auswertung getätigten Handlung anfallen. Augenfälligstes Beispiel hierfür sind die zahllosen Analysen sozialer Medien (z. B. Ceron et al. 2014; Barberà 2015). Die Konsequenzen für die sozialwissenschaftliche Forschung sind in diesem Punkt viel weitreichender als auf den ersten Blick zu erkennen. Es ist ein entscheidendes Charakteristikum der digitalen Revolution, dass sich menschliches Verhalten viel leichter digitalisieren, mithin quantifizieren lässt. Zum ersten Mal werden somit viele Verhaltensmuster sichtbar, die für die Forschung vormals kaum

systematisch greifbar waren. Deshalb werden auch die genutzten Datenquellen um ein Vielfaches variabler.

Zwei Beispiele seien hier stellvertretend für die facettenreiche Forschungslandschaft herausgegriffen. Zum einen haben die Sozialwissenschaften damit begonnen, das massenhaft verfügbare digitale Bildmaterial zu analysieren. Ein interessanter Anwendungsfall ist etwa die Auswertung von Satellitenbildern in der Konfliktforschung. Wenn amtliche Daten zu kleinräumigem Wohlstand nicht verfügbar oder fehlerbehaftet sind, so wird in diesen Arbeiten argumentiert, dann können nächtliche Lichtemissionen als Wohlstands-Proxy herangezogen werden. Satellitenaufnahmen bieten folglich einen systematischen Zugriff auf lokale Wohlstandsvariationen, die mit Konfliktmustern in Beziehung gesetzt werden können (Cederman et al. 2015; Kuhn und Weidmann 2015; Weidmann und Schutte 2017). Dabei machen sich solche Analysen geokodierte Daten zunutze, denen in der *Big Data*-Literatur grundsätzlich eine ganz erhebliche Bedeutung zukommt (z. B. Egan und Mullin 2012; Hersh und Nall 2016).

Ein zweites Beispiel für eine innovative, aber durchaus kontroverse sozialwissenschaftliche Datenquelle, die sich die umfassenden Möglichkeiten zur Aufzeichnung menschlichen Handelns zunutze macht, sind die sogenannten Paradaten, die in der Survey-Forschung anfallen (Heerwegh 2003; Couper und Kreuter 2013; Kreuter 2013; Felderer et al. 2014). Während sich die klassische Survey-Forschung ausschließlich mit den Antworten der Respondenten beschäftigt, wurden gerade durch die immer umfassendere Nutzung von webbasierten Surveys Potenziale sichtbar, prozessgenerierte Daten mitzuerheben, um Rückschlüsse über die Survey-Teilnehmer zu ziehen. Der trivialste Fall solcher Paradaten sind ohne Zweifel die Antwortzeiten, die mittlerweile vielfach dokumentiert werden. Es können aber ebenso leicht weniger naheliegende Paradaten erhoben werden, etwa Tastatureingaben und -löschungen, Mausbewegungen, oder die Häufigkeit des Vor- und Zurückblätterns bei der Beantwortung eines Fragebogens. Entsprechende Daten sind zwar nicht im strengen Sinne Nebenprodukte einer nicht auf die sozialwissenschaftliche Auswertung ausgerichteten Handlung, da sie dezidiert im Rahmen der Forschung entstehen. Nebenprodukte sind sie jedoch in dem Sinne, dass sie nicht der offensichtliche Anlass der Forschung sind und womöglich sogar in Unkenntnis der Teilnehmer erhoben werden.

Ein drittes Merkmal von *Big Data* ist schließlich häufig die Komplexität der Daten, die sich nur schlecht in einer zweidimensionalen Datenmatrix abbilden lassen. Auch hier hilft zur Verdeutlichung der Vergleich mit klassischen Umfragedaten. Im Bereich der Survey-Forschung repräsentieren die Zeilen üblicherweise die Teilnehmer, während die Spalten die Variablen abbilden. Die Zellen beinhalten also die Ausprägungen der spezifischen Respondenten-Variablen-Kombinationen.

Dieses Schema ist kaum ausreichend, wenn wir das oben genannte Beispiel der Auswertung von Paradaten weiterdenken. Mausbewegungen und Tastatureingaben lassen sich nur mit erheblichen Informationsverlusten als eine fixe Ausprägung einer festen Zahl von Variablen ausdrücken. Die Daten müssen also zunächst in einem Format gespeichert werden, welches der Komplexität der Rohdaten gerecht wird. Erst zur Auswertung werden die Daten dann zum Beispiel in Metriken übersetzt, die sich für die Analyse mit klassischen statistischen Verfahren eignen. Entsprechende Metriken können etwa die mittlere Antwortzeit oder die Häufigkeit des Wechsels einer einmal gewählten Antwortkategorie sein.

In der sozialwissenschaftlichen Forschungspraxis ist *Big Data* nun vornehmlich verknüpft mit der Sammlung und Auswertung von Daten aus dem Internet. Die umfassende Digitalisierung aller Lebensbereiche ist aus Sicht der Sozialwissenschaften ein Glücksfall, werden durch diese Entwicklung doch Daten für die Bearbeitung zahlloser Fragestellungen verfügbar. Dabei ergeben sich für die praktische Arbeit vor allem zwei Herausforderungen. Zum einen ist es notwendig, sich mit dem technischen Rüstzeug vertraut zu machen, um die verfügbaren Datenbestände automatisch zu sammeln. Zum anderen sind für die Auswertung von *Big Data* spezielle Verfahren entwickelt worden, die den Besonderheiten solcher Daten gerecht werden.

Die Ziele dieses Kapitels sind entlang dieser beiden Herausforderungen strukturiert. Zum einen werden die Grundlagen der automatischen Webdatensammlung diskutiert. Zum anderen werden die grundsätzlichen Überlegungen zur Auswertung großer, wenig strukturierter Daten ausgeführt. Webdaten sind häufig Textdaten, weshalb der Schwerpunkt in der Datenauswertung auf Verfahren zur Handhabung großer Textbestände liegt. Letzteres schließt unmittelbar an die Überlegungen aus dem Kapitel zur Inhaltsanalyse an (siehe den Beitrag von Braun in diesem Band). Während dort jedoch die manuelle Quantifizierung von Text im Vordergrund steht, liegt der Fokus hier auf der computergestützten Quantifizierung. In vielen praktischen Anwendungen ist die automatische Auswertung zwingend, da die Masse des auszuwertenden Materials eine manuelle Vollkodierung ausschließt.

Im weiteren Verlauf dieses Kapitels widmen wir uns zunächst den technischen Grundlagen der webbasierten Datensammlung. Abschn. 3 deutet die Grundlagen der Auswertung webbasierter Datenbestände an, was sich insbesondere mit dem Begriff *Machine Learning* verknüpft. Abschn. 4 diskutiert ein Beispiel aus der aktuellen Forschungsliteratur. Schwerpunkt dieser Darstellung sind die notwendigen Schritte zur Replikation des Beispiels – sowohl der Datensammlung als auch der Analyse. Abschn. 5 schließt mit einer Bewertung von *Big Data* für die sozialwissenschaftliche Forschung.

2 Grundlagen der Webdatensammlung

2.1 Programmierschnittstellen

In diesem Abschnitt wollen wir uns mit den Grundlagen der Webdatensammlung auseinandersetzen. Dabei verzichten wir im Sinne des Einführungscharakters dieses Textes auf die Darstellung der technischen Details und beschäftigen uns stattdessen mit den wesentlichen Prinzipien. Für Leserinnen, die an der Anwendung der genannten Techniken interessiert sind, geben wir am Schluss dieses Abschnitts einige Literaturhinweise zur tiefergehenden und praktischen Beschäftigung mit Fragen der Webdatensammlung.

Zunächst können wir zwei grundsätzliche Arten der Webdatensammlung unterscheiden – zum einen die Sammlung von Datenmaterial aus dem Quelltext von Webseiten (*Web Scraping*), zum anderen die Datensammlung über Programmierschnittstellen (APIs, *Application Programming Interfaces*). Obwohl die zugrunde liegende technische Infrastruktur viele Gemeinsamkeiten aufweist, ist Letztere mit deutlich weniger Aufwand verbunden. Programmierschnittstellen werden von vielen Anbietern explizit aufgesetzt, um zum Beispiel den Datenaustausch mit Anwendern zu erleichtern. In der Praxis ist es so, dass Anbieter eine Reihe von zulässigen Operationen definieren, die an der Schnittstelle zwischen Anwendern und Anbietern erfolgen können. Schauen wir uns als Beispiel eine Anfrage an die Programmierschnittstelle der deutschen *Wikipedia* an:

https://de.wikipedia.org/w/api.php?action=query&titles=Bundestag&prop=pageviews&pvipdays=7&format=xml

Wir können aus dieser Anfrage verschiedene Dinge lernen. Zunächst fällt auf, dass die Anfrage das gewöhnliche URL-Format aufweist, welches wir aus der Interaktion mit Webseiten kennen. Wenn wir die Anfrage im Detail betrachten, dann können wir verschiedene Elemente unterscheiden. Die Anfrage beginnt mit der Festlegung des Protokolls (HTTPS), das verwendet wird, um Mitteilungen zwischen Nutzer (Client) und Anbieter (Server) auszutauschen. Der Inhalt solcher Mitteilungen ist hochgradig flexibel und kann beispielsweise der Quellcode einer HTML-Seite sein, aber beispielsweise auch ein JPEG, ein PDF oder, wie im vorliegenden Fall, ein XML-formatierter Datensatz. Das Protokoll definiert dabei die Struktur der Nachricht, die den Inhalt rahmt. Sie braucht uns hier nicht weiter zu beschäftigen.

Nach der Festlegung des zu verwendenden Protokolls wird die *Domain* genannt, also letztlich der anzusprechende Server, in diesem Fall „de.wikipedia.org“,

zusammen mit dem Pfad der gesuchten Ressource, in diesem Fall „/w/api.php“. Als letztes Element folgt schließlich die spezifische Anfrage, die wir an die API stellen und die durch ein Fragezeichen eingeleitet wird. Die Anfrage teilt sich auf in einzelne Parameter und Parameterwerte, die durch ein Gleichheitszeichen verbunden sind und durch ein „&“ voneinander getrennt werden. In seinen Grundzügen beschreibt diese Abfolge den Großteil der URLs, mit denen Sie jemals konfrontiert sein werden. Tab. 1 dokumentiert die spezifischen Parameter der Anfrage:

Die Anfrage lässt sich die Zahl der Seitenaufrufe (prop=pageviews) der *Wikipedia*-Seite des Deutschen Bundestags (titles=Bundestag) der letzten sieben Tage (pvipdays=7) im XML-Format ausgeben (format=xml). Wir wollen uns nicht weiter mit den spezifischen Parametern der Anfrage auseinandersetzen und verweisen bei Interesse auf die Überblicksseite der *Wikipedia*-API, die unter der Adresse https://de.wikipedia.org/w/api.php zu erreichen ist. Wichtig ist für uns lediglich, dass die möglichen Parameter vom Anbieter der API definiert werden, der somit die zulässigen Operationen festlegt. Dabei mag es Beziehungen zwischen den Parametern geben, sodass sich etwa der Parameter „pvipdays“ explizit auf den Parameterwert „pageviews“ des Parameters „prop“ bezieht, der den Zeitraum der Suchanfrage beschreibt. Neben den möglichen Parametern wird vom Datenanbieter auch festgelegt, welche Parameterwerte überhaupt zulässig sind. So können im Parameter „action“ und auch in den weiteren Parametern nur eine begrenzte Zahl von Operationen genannt werden, die vom Server akzeptiert werden.

Wenn Sie die oben genannte URL in Ihren Browser kopieren, dann sollten Sie vom Server eine Antwort erhalten, die sehr ähnlich der in Abb. 1 dargestellten ausschaut. Sie können leicht erkennen, an welcher Stelle das Dokument die Informationen über die Zahl der täglichen Seitenaufrufe enthält, beispielsweise 56 Aufrufe am 20. Juli 2018, 30 Aufrufe am 21. Juli 2018 und so weiter.

Aus der Antwort können wir eine Menge über die Sammlung von Webdaten lernen. Wenn Sie sich schon einmal mit dem Quell-Code einer normalen HTML-Seite beschäftigt haben, dann werden Sie feststellen, dass die Antwort eine sehr

Tab. 1 Parameter und Werte der Anfrage an die *Wikipedia*-API. Quelle: Eigene Darstellung

Parameter	Wert
action	query
titles	Bundestag
prop	pageviews
pvipdays	7
format	xml

```
▼<api batchcomplete="">
 ▼<query>
  ▼<pages>
   ▼<page _idx="2454605" pageid="2454605" ns="0" title="Bundestag">
    ▼<pageviews>
       <pvip date="2018-07-20" xml:space="preserve">56</pvip>
       <pvip date="2018-07-21" xml:space="preserve">30</pvip>
       <pvip date="2018-07-22" xml:space="preserve">45</pvip>
       <pvip date="2018-07-23" xml:space="preserve">49</pvip>
       <pvip date="2018-07-24" xml:space="preserve">52</pvip>
       <pvip date="2018-07-25" xml:space="preserve">46</pvip>
       <pvip date="2018-07-26" xml:space="preserve">40</pvip>
     </pageviews>
    </page>
   </pages>
  </query>
 </api>
```

Abb. 1 Antwort auf die API-Anfrage (Stand: 28. Juli 2018) Quelle: Eigene Darstellung

ähnliche Struktur wie eine übliche Webseite aufweist. Das ist kein Zufall, da wir die Daten mittels des Parameters „format“ im XML-Format angefordert haben. Das XML-Format ist eng verwandt mit dem HTML-Format und unterscheidet sich vornehmlich dadurch, dass es weniger Vorgaben hinsichtlich der zulässigen Strukturelemente macht, die in XML-verwandten Formaten durch die Vergleichszeichen (<) und (>) eingerahmt werden.

Bevor wir uns weiter unten mit den spezifischen Strukturelementen beschäftigen, seien zunächst einige Grundlagen des Dokuments angesprochen, das wir als Antwort erhalten. Bei einem XML-Dokument handelt es sich um nichts anderes als um ein Textdokument, das von Strukturelementen gegliedert wird. Bei der Färbung und Einrückung des Textes in Abb. 1 handelt es sich also bereits um eine Interpretation des Dokuments durch den Browser, die automatisch zum Zwecke der besseren Lesbarkeit vorgenommen wird. Wenn Sie das Dokument herunterladen und mit einem einfachen Texteditor öffnen, dann werden Sie feststellen, dass das Dokument schlicht aus einer Zeichenfolge besteht.

Grundsätzlich ist es so, dass im Bereich der Webdatensammlung viele Informationen als Zeichenfolge mit spezifischen Strukturelementen vorliegen. Neben XML- und HTML-Dokumenten gilt das beispielsweise für das JSON-Format, das uns in Abschn. 4 begegnen wird. Auch in diesem Fall wird die Zeichenfolge nach gewissen Regeln strukturiert, um eine Interpretation der darin enthaltenen Daten zu ermöglichen. Die *Wikipedia*-API ermöglicht es uns, die Daten nicht nur im XML-Format abzurufen, sondern ebenso im JSON-Format. Wenn Sie den Parameterwert des Parameters „format“ in der obigen Anfrage von „xml“ zu „json“

ändern und die Anfrage erneut in die Adresszeile Ihres Browsers kopieren, dann erhalten Sie denselben Datensatz wie zuvor – die Daten sind lediglich anders strukturiert.

Durch die testweise Anpassung des Parameterwertes liegt nun auf der Hand, wie wir größere Datensätze von der *Wikipedia*-API abfragen können. Wir können leicht den Wert des „titles"-Parameters variieren, um Datensätze zu verschiedenen Themen zu sammeln. Nehmen wir an, Sie interessieren sich für die Popularität der deutschen Großstädte und wollen die Zahl der täglichen Seitenaufrufe der entsprechenden *Wikipedia*-Seiten als Popularitätsindikator heranziehen. In diesem Fall brauchen Sie lediglich eine Liste der deutschen Großstädte – die Sie freilich ebenfalls von *Wikipedia* erhalten können –, um den entsprechenden Parameter in der Anfrage durch alle relevanten Ausprägungen zu ersetzen und die Anfrage an die API senden.

Ähnlich wie *Wikipedia* stellen zahllose Anbieter ihre Daten über Programmierschnittstellen zur Verfügung. Bei der Auseinandersetzung mit dem Beispiel aus der praktischen Forschung werden uns etwa Programmierschnittstellen von *Google* und *Twitter* begegnen. Der große Vorteil von APIs ist zunächst, dass wir nur die gesuchten Daten in einem Datenformat erhalten. Die Informationen müssen also nicht mühsam aus dem Quelltext einer Seite zusammengesammelt werden. Dies ist umso wertvoller, da gewisse Problemstellungen des *Web Scraping*, beispielsweise die Handhabung dynamischer Seiten, umgangen werden können. Gelegentlich ist es sogar so, dass über eine API Daten vom Server abgerufen werden können, die anderweitig nicht einzusehen sind. Aus Sicht wenig technikaffiner Anwender mag ein weiterer Vorteil von APIs sein, dass in vielen Programmiersprachen Zusatzmodule implementiert wurden, die eine Art Aufsatz für die gängigen Programmierschnittstellen bieten. In diesem Fall können API-Anfragen problemlos mit den vorhandenen Funktionen abgesetzt werden, ohne sich mit den technischen Details der APIs auseinandersetzen zu müssen.

Ein weiterer Unterschied zwischen dem *Web Scraping* und der Datensammlung mittels Programmierschnittstellen ist schließlich, dass Letztere eindeutig von den Datenanbietern vorgesehen ist. Damit schafft die Nutzung von APIs allerdings auch Abhängigkeiten, die ein nicht zu übersehender Nachteil der Datensammlung mittels Programmierschnittstellen sind. So werden dem Nutzer bei der Verwendung von APIs häufig Beschränkungen auferlegt. Noch vergleichsweise unproblematisch ist dies, wenn Datenanbieter eine Registrierung voraussetzen, um das individuelle Nutzerverhalten nachvollziehen zu können. Weitaus gewichtiger ist dagegen die ebenfalls häufig vorgenommene Beschränkung der Menge an Datenabfragen in einem gewissen Zeitraum oder aber die Beschränkung, welche Daten über die API eingesehen werden können. Datenanbieter versuchen auf diesem

Wege, die Belastung des Servers durch einzelne Nutzer verständlicherweise zu begrenzen. In der Praxis führt dies allerdings dazu, dass manche Programmierschnittstellen für gewisse Forschungsanwendungen nur wenig brauchbar sind.

2.2 Web Scraping

In einem solchen Fall und selbstverständlich auch dann, wenn keine API vorliegt, mag es helfen, auf das *Web Scraping* zurückzugreifen, also die Datenextraktion aus dem Quelltext einer Webseite. Die hierzu notwendigen technischen Grundlagen führen weit über die Dimensionen dieses Kapitels hinaus. Sie seien deshalb nur in ihren absoluten Grundzügen und auch nur für vergleichsweise simple Anwendungsfälle dargestellt. Für einen ausführlicheren Blick auf diese Themen bieten die Literaturhinweise am Schluss dieses Abschnitts eine gute Einstiegsmöglichkeit.

Kommen wir zur Darstellung der Grundlagen des *Web Scrapings* zu dem Dokument in Abb. 1 zurück. Da XML- und HTML-Dokumente gleich strukturiert sind, können wir auf Basis des Dokuments eine Reihe von Prinzipien lernen, die auch für das *Web Scraping* relevant sind, also für das Sammeln von Daten aus dem Quellcode einer Webseite. Deutlich zu erkennen ist, dass das Dokument hierarchisch aufgebaut ist und die einzelnen Strukturelemente ineinander verschachtelt sind. Weiterhin gilt, dass sich die Strukturelemente im Regelfall durch ein öffnendes (bspw. <query>) und ein schließendes Element (</query>), sogenannte *Tags*, auszeichnen. Dabei folgt das Öffnen und Schließen dem hierarchischen Aufbau des Dokuments, die inneren Elemente werden also vor den weiter außenliegenden Elementen geschlossen. Zusammengenommen werden der öffnende und schließende *Tag* in einem HTML-Dokument Knoten genannt. Schließlich können wir erkennen, dass die öffnenden *Tags* weitere Attribut-Wert-Paare enthalten können, beispielsweise das „date“-Attribut im „pvip“-*Tag*, das die einzelnen Datumsangaben enthält.

Der entscheidende Unterschied zwischen XML-formatierten Daten und HTML-Seiten ist die Bedeutung der Strukturelemente in HTML-Dokumenten. Während Datenanbieter die Tags und Attribute im XML-Format frei benennen können, sind die Strukturelemente bei HTML-Seiten fest vorgegeben. Dies ermöglicht die Interpretation des Quelltextes und die Darstellung einer interpretierten Version der HTML-Seite im Browser. Ein <a>-*Tag* beispielsweise wird in HTML-Dokumenten stets für Verlinkungen verwendet. Auch die zulässigen Attribute der *Tags* sind fest vorgegeben, sodass das Attribut „href“ stets das Ziel einer Verlinkung beschreibt. Wenn Sie im Quellcode einer HTML-Seite also den Ausdruck „Wikipedia (deutsch) finden“, dann wissen Sie jetzt, dass in

Ihrem Browser ein Link zur Startseite der deutschen Wikipedia sichtbar sein wird. In der interpretierten Version ist dabei nur der Text zwischen dem öffnenden und schließenden *Tag* sichtbar, also:

Wikipedia (deutsch)

Während es die Aufgabe des Browsers ist, eine interpretierte Version des Quelltextes darzustellen, beschäftigen wir uns beim *Web Scraping* üblicherweise mit dem Quelltext einer HTML-Seite. Dabei interessieren wir uns vor allem für die Frage, an welcher Stelle im Quelltext sich die gesuchten Informationen befinden und wie wir die relevanten Stellen extrahieren können. Hierzu machen wir uns eins von mehreren Werkzeugen zunutze, die zu diesem Zweck entwickelt wurden.

XPath beispielsweise bietet eine ausgesprochen flexible Syntax zur systematischen Beschreibung und Extraktion eines oder mehrerer Elemente aus einem HTML-Dokument. Dabei kann die Syntax aufgrund der Verwandtschaft zwischen HTML und XML gleichermaßen auf XML-Dokumente angewendet werden. Wenn wir beispielsweise alle „date“-Attribute aus dem Dokument in Abb. 1 extrahieren wollen, dann können wir den folgenden XPath-Ausdruck verwenden: „//pvip/@date“. Auch hier wollen wir uns nicht mit den Details der Syntax beschäftigen, sondern lediglich die grundsätzlichen Überlegungen darstellen, die uns zu diesem Ausdruck führen. Die ersten beiden Schrägstriche bedeuten, dass wir einen pvip-Knoten irgendwo im Dokument suchen, der ein date-Attribut enthält. Da dies auf alle gesuchten Knoten, aber auf keine weiteren zutrifft, können wir somit alle gesuchten Informationen auf einen Schlag aus dem Dokument ziehen.

Das Beispiel ist zugegebenermaßen sehr übersichtlich, das grundsätzliche Vorgehen unterscheidet sich aber nicht wesentlich von der ernsthaften Webdatensammlung. Auch wenn wir Informationen aus Dokumenten mit vielen Hundert Zeilen extrahieren, dann ist das Ziel stets die Formulierung eines Ausdrucks, der allgemein genug ist, um alle gesuchten Informationen zu fassen und gleichzeitig spezifisch genug, um ungewollte Elemente zu übergehen. Der Mehrwert eines solchen Vorgehens wird dabei auf den ersten Blick deutlich. Sobald das Extraktionsproblem für eine Seite gelöst ist und die Seitenstruktur sich nicht ändert, dann ist das Problem für alle Seiten gelöst. Nehmen wir beispielsweise an, dass Sie die Nachrichtentexte eines Nachrichtenportals auslesen wollen. Sobald Sie in der Lage sind, die gesuchten Elemente Titel, Untertitel, Veröffentlichungsdatum und Text aus dem Quelltext zu extrahieren, dann brauchen Sie lediglich alle relevanten Dokumente automatisch öffnen, um die gesuchten Informationen mit demselben Ausdruck zu extrahieren.

Die praktische Umsetzung der angedeuteten Techniken ist leichter, als es auf den ersten Blick erscheinen mag. Für die gängigen Programmiersprachen sind Zusatzmodule veröffentlicht worden, welche die komplexeren Aspekte der Interaktion zwischen Client und Server ähnlich wie der Browser automatisieren und spezielle Funktionen bereitstellen, um die gängigen Anwendungsfälle weiter zu vereinfachen.

Die größte Hürde für viele Anfänger ist vermutlich die Notwendigkeit, sich überhaupt mit einer Programmiersprache zu beschäftigen. Während es zweifellos nie schaden kann, sich mit einer Programmiersprache auseinanderzusetzen, mag die Hürde für Sozialwissenschaftlerinnen noch am niedrigsten sein, wenn sie für die Webdatensammlung die Programmiersprache R verwenden. Das Statistikprogramm R (https://www.r-project.org/) hat in den Sozialwissenschaften und darüber hinaus eine enorme Popularität erlangt, sodass einige Leserinnen womöglich bereits mit den Grundlagen des Programms vertraut sind und der Einstieg in die Webdatensammlung somit etwas leichter fällt.

Doch selbst bei nicht vorhandenen Programmkenntnissen erscheint das Erlernen von R für quantitativ orientierte Sozialwissenschaftler lohnenswert – und wie könnten Leser eines Kapitels zu *Big Data* kein Interesse an quantitativer Sozialforschung haben? –, da die erworbenen Fähigkeiten auch für die weiteren Aspekte der Datenanalyse genutzt werden können, sei es zur Aufbereitung und Verarbeitung von Daten oder zur Datenanalyse und -visualisierung. Wenn also ohnehin eine Programmiersprache erlernt werden muss, dann ist R die naheliegende Wahl, da es auch im sonstigen Forschungsalltag Verwendung finden kann und alle Schritte eines typischen Datenprojekts in derselben Umgebung vollzogen werden können. Dies gilt umso mehr, da viele statistische Techniken mittlerweile zuerst in R implementiert werden. Die Kenntnis des Programms erlaubt es somit, von neueren Entwicklungen unmittelbar zu profitieren. Der Vollständigkeit halber sei darauf hingewiesen, dass R aus nicht-sozialwissenschaftlicher Sicht vermutlich nicht das Werkzeug der Wahl für die Webdatensammlung ist. Da jedoch mittlerweile für alle üblichen Anwendungen in diesem Bereich Zusatzpakete in R geschrieben wurden, sind die Anreize zur Nutzung einer gängigeren Programmiersprache gering.

Einen ersten Eindruck der praktischen Umsetzung der genannten Techniken bietet der Beitrag von Munzert und Nyhuis (im Erscheinen), der die automatische Webdatensammlung sowohl im Bereich des *Web Scrapings* als auch in der Auseinandersetzung mit Programmierschnittstellen anhand verschiedener sozialwissenschaftlich orientierter Beispiele darstellt. Der Text enthält kleinere Code-Beispiele, die für eigene Forschungsprojekte angepasst werden können.

Deutlich umfassender ist die Einführung der Verfahren zur automatisierten Datensammlung bei Munzert et al. (2014). In dieser Arbeit werden die angedeuteten

Techniken in einem ersten Schritt umfassend eingeführt, um in einem zweiten Schritt verschiedene, teils komplexe Szenarien der Webdatensammlung durchzuspielen. Das Buch schließt mit ausgewählten Anwendungsbeispielen, die den gesamten Prozess datenorientierter Projekte vorstellen. Nicht verschwiegen sei allerdings, dass die rasante technische Entwicklung der vergangenen Jahre dem Buch etwas zu schaffen macht und einige mittlerweile zentrale Erweiterungsmodule für die Webdatensammlung in dem Buch noch nicht aufgegriffen werden konnten.

Ein zweites umfassendes Einführungswerk für die Webdatensammlung in R ist das Buch von Nolan und Temple Lang (2014). Die Autoren bieten ebenfalls eine Übersicht der Implementation zentraler Webtechnologien in R. Dabei unterscheidet sich das Werk insofern von der Arbeit von Munzert et al. (2014), als der Zuschnitt etwas anspruchsvoller und weniger anwendungsorientiert ist. Ein Vorteil des Buches von Nolan und Temple Lang ist sicher, dass es auch bei Grenzfällen Hilfestellungen bietet. Gleichzeitig ist es jedoch so, dass der Großteil der praktischen Datensammlungsaufgaben mit den grundlegenden Techniken geleistet werden kann. Im Übrigen gilt auch für die Arbeit von Nolan und Temple Lang (2014), dass eine Reihe von Standarderweiterungen in R für die Webdatensammlung bei der Veröffentlichung des Buches noch nicht verfügbar war und entsprechend nicht aufgegriffen wird.

Schließlich ist R wie bereits bemerkt bei weitem nicht die einzige Programmiersprache für die Sammlung von Webdaten. Die momentan vermutlich geläufigste Sprache ist Python. Python bietet deutlich mehr Funktionalität in diesem Bereich – auch durch seine Stärke in benachbarten Feldern der Webdatensammlung, zum Beispiel in Fragen des Umgangs mit Bildern oder natürlicher Sprache. Für interessierte Leser, die sich mit der Webdatensammlung in Python beschäftigen wollen, bietet Mitchell (2015) einen hervorragenden und umfassenden Überblick mit vielen Code-Beispielen.

3 Die Auswertung von Big Data

Auch in der Datenauswertung hat *Big Data* zu einigen Neuerungen geführt. Diese Neuerungen lassen sich vornehmlich auf die Merkmale der Daten zurückführen, die im Bereich der Webdatensammlung anfallen. Anders als bei der gewöhnlichen statistischen Analyse liegt der Fokus hier nicht auf dem Zusammenhang spezifischer Variablen, sondern auf der kategorialen Zugehörigkeit der zu untersuchenden Fälle.

Es wird also nicht der Zusammenhang zwischen den Variablen A, B und C im Rahmen eines statistischen Modells postuliert und geprüft, ob dieser Zusammenhang besteht. Stattdessen werden Datensätze mit vielen Variablen gebildet, um die Kategorie einer bestimmten Einheit vorherzusagen. Die hierzu verwendeten Modelle identifizieren dabei automatisch diejenigen Variablen, die sich am besten eignen, um zwischen der kategorialen Zugehörigkeit der Fälle zu unterscheiden. Die Summe der entsprechenden Techniken wird unter dem Stichwort *Machine Learning* diskutiert. Sie finden Anwendung in ganz unterschiedlichen Bereichen, beispielsweise in den Recommendation Engines von Amazon und YouTube, in der Spracherkennung in digitalen Assistenzsystemen, oder in der Gesichtserkennung in Videodaten.

Im Folgenden wollen wir uns beispielhaft mit der Nutzung des *Machine Learnings* im Bereich der Textanalyse als einem der häufigsten sozialwissenschaftlichen Anwendungsfälle beschäftigen. So fallen etwa in Nachrichtenportalen, Blogs und sozialen Medien vornehmlich wenig strukturierte Textdaten an. Solche Daten gehen mit besonderen Herausforderungen einher, die mit klassischen Analysestrategien nicht gut zu fassen sind und deshalb zu einem enormen Interesse an der computergestützten Auswertung von Text geführt haben. Dieser Abschnitt beschreibt zunächst die wesentlichen Überlegungen der relevanten Techniken, eine praktische Anwendung wird im folgenden Abschnitt dargestellt.

Im Bereich des *Machine Learnings* können wir zwischen zwei grundlegenden Varianten unterscheiden, dem *Unsupervised Learning* und dem *Supervised Learning*. In beiden Fällen ist es dabei so, dass die momentan üblichen Textanalyseverfahren die Syntax von Sprache nicht berücksichtigen und sich ausschließlich mit dem Auftreten oder Nicht-Auftreten von Begriffen beschäftigen. Diese Verfahren basieren somit auf der Annahme, dass allein die Verwendung bestimmter Begriffe Hinweise darüber liefert, wie ein Text klassifiziert werden sollte. Konkret werden die gesammelten Texte, zum Beispiel eine Sammlung von Nachrichtenartikeln, zunächst in eine Term-Dokument-Matrix umgewandelt. In einer solchen Matrix repräsentieren die Zeilen etwa die Texte, die Spalten die Begriffe, sodass die Zellen die Häufigkeit der Verwendung eines bestimmten Begriffs in einem Artikel ausweisen.[1]

[1] Aufgrund des Überblickscharakters dieses Kapitels und dem Fokus auf Prinzipien des *Machine Learnings* werden viele Feinheiten der Verarbeitung von Textdaten nicht berücksichtigt, etwa die Vorbereitung von Texten für die Analyse (Denny und Spirling 2018) oder Gewichtungsverfahren bei der Erstellung von Term-Dokument-Matrizen.

Das gegenwärtig am häufigsten verwendete Modell aus dem Bereich des *Unsupervised Learnings* ist die sogenannte *Latent Dirichlet Allocation* (LDA; Blei et al. 2003). Bei dieser Technik werden die Texte auf Basis von Ähnlichkeiten in den verwendeten Begriffen kategorisiert. Eine Besonderheit dieses Verfahrens ist, dass das LDA-Modell prinzipiell zulässt, dass mehrere Themen in einem Text auftreten. Allgemein wird die Kategorisierung bei Verfahren aus dem Bereich des *Unsupervised Learnings* einzig auf Basis von Systematiken in der Datenmatrix vorgenommen, eine menschliche Kodierung ist also nicht notwendig. Die einzige menschliche Modellvorgabe bei der LDA ist die Zahl der Themen, die für die Klassifizierung genutzt wird. Die Zahl der zu schätzenden Themen ist ausgesprochen folgenreich für die Ergebnisse und sollte sich deshalb aus theoretischen Überlegungen dazu ergeben, wie viele Themen im Textkorpus zu erwarten sind.

Während das LDA-Modell leicht zur Klassifizierung von Textdaten angewendet werden kann, erfordert es einen erheblichen Aufwand bei der Interpretation der Ergebnisse. Das Modell gibt eine Reihe von Begriffen aus, die besonders stark mit den einzelnen Themen zusammenhängen und es ist Aufgabe der Anwender, diese Begriffe für die Ergebnispräsentation mit einem Label zu verdichten. Häufig wird diese Aufgabe dadurch erschwert, dass manche Themen durch Begriffe charakterisiert werden, die sich nur schwerlich zu einer sinnvollen Kategorie verdichten lassen. Mindestens aber gibt es an diesem Punkt einen erheblichen und durchaus problematischen Interpretationsspielraum. Wichtig ist schließlich, dass die LDA ausschließlich für die thematische Kategorisierung von Texten verwendet werden kann. Das Verfahren eignet sich also nicht für die Bearbeitung von Forschungsfragen, die nicht nach der Themenorientierung von Texten fragen. Es liegt nahe, dass das angesprochene Thema stets die dominierende Kategorie ist, wenn das Kategorienschema automatisch aus dem Textkorpus heraus entwickelt wird.

Das LDA-Modell wurde von Roberts et al. (2014) zum *Structural Topic Model* (STM) weiterentwickelt. Dieses Modell ermöglicht Anwenderinnen die Verbesserung der Schätzung durch Einbeziehung textspezifischer Kovariate. Grundgedanke dieses Modells ist, dass wir häufig Strukturinformationen über die zu analysierenden Texte haben und diese Informationen sowohl mit der Auftretenswahrscheinlichkeit bestimmter Themen zusammenhängen als auch mit den Begriffen, die verwendet werden, um bestimmte Themen anzusprechen. Wenn uns etwa interessiert, welche Themen die Parteien in ihren parlamentarischen Anfragen ansprechen, dann könnten wir die unterzeichnende Partei als textspezifisches Kovariat verwenden, da die Parteien vermutlich systematisch unterschiedliche Themen in ihren Anfragen ansprechen, sodass beispielsweise eine Anfrage der Grünen eine höhere Grundwahrscheinlichkeit hat, sich mit dem Thema Umweltpolitik zu beschäftigen.

Beispiel für das *Unsupervised Learning*
Jankowski et al. (2019) untersuchen in ihrem Beitrag, welche Assoziationen die Kandidierenden zur Bundestagswahl 2013 mit dem Begriff „rechts" verbinden und ob es diesbezüglich Variationen innerhalb und zwischen den Parteien gibt. Zu diesem Zweck nutzen die Autoren die Kandidatenantworten auf eine offene Frage nach der Definition des Begriffs „rechts" im Rahmen einer Kandidatenbefragung. Methodisch greifen sie auf ein *Structural Topic Model* mit drei Themen zurück. Auf Basis der Begriffe, die am stärksten mit den drei geschätzten Themen zusammenhängen, vergeben sie zusammenfassende Labels. Dies sind „Freiheit und Verantwortung" (Eigenverantwortung, Verantwortung, Marktwirtschaft, Freiheit), „Rassismus und Intoleranz" (Rassismus, Intoleranz, Nationalismus, Kapital) und „Konservatismus und Status Quo" (Konservativ, National, Egoismus, Wirtschaft). Die hoch-indikativen Begriffe für die Themen sind in Klammern wiedergegeben. Die Autoren kommen zu dem Schluss, dass die drei Themen zu jeweils etwa einem Drittel in den Kandidatenantworten auftreten. Substantiell finden sie unter anderem, dass das Thema „Freiheit und Eigenverantwortung" deutlich häufiger in den Antworten der CDU/CSU- und AfD-Kandidaten auftritt als in den Antworten der Kandidaten linker Parteien.

Weitere praktische Anwendungen der LDA finden sich beispielsweise bei Jacobi et al. (2016), Lauderdale und Clark (2014) und Tzelgov (2014); Bauer et al. (2017) sowie Rothschild et al. (2019) verwenden das *Structural Topic Model*. Die genannten Verfahren sind zwar nicht die einzigen Modelle aus dem Bereich der *Unsupervised Classification* (Grimmer 2010; Quinn et al. 2010). Sie sind in der sozialwissenschaftlichen Forschung gegenwärtig jedoch am weitesten verbreitetet.

Anders gelagert ist das Vorgehen bei den Verfahren aus dem Bereich des *Supervised Learnings*. In diesen Fällen wird ein Teil des Materials, die Trainingsdaten, durch menschliche Kodierer handkodiert und die nicht-kodierten Texte werden auf Basis von Ähnlichkeiten in der Wortverwendung relativ zu den Referenztexten automatisch in die vorgegebenen Kategorien klassifiziert. Für diese Form der Klassifizierung sind eine ganze Reihe von Verfahren entwickelt worden. Das bekannteste Verfahren sind *Support Vector Machines* (D'Orazio et al. 2014). Der Vorteil der Methoden aus diesem Bereich ist zweifellos, dass auch Fragestellungen jenseits thematischer Kategorien bearbeitet werden können. Wenn wir uns beispielsweise

für Bewertungen interessieren, die in Texten ausgedrückt werden, dann können wir dieses Interesse durch das Trainingsmaterial vorgeben, sodass der Algorithmus diejenigen Textattribute identifiziert, die für das Vorliegen einer bestimmten Bewertung sprechen und etwa die thematischen Marker weniger stark gewichtet.

Beispiel für das *Supervised Learning*
Peterson und Spirling (2018) beschäftigen sich in ihrer Arbeit mit der Polarisierung im britischen Unterhaus, also letztlich mit der Frage, inwiefern sich die beiden großen Parteien als unversöhnliche Blöcke gegenüberstehen und wie sich dieses Verhältnis über Zeit gewandelt hat. Sie argumentieren, dass das Abstimmungsverhalten, anders als etwa bei Analysen des US-Kongresses, zur Untersuchung dieser Frage wenig brauchbar ist, da Abgeordnete aufgrund hoher Parteidisziplin nur äußerst selten von der Parteilinie abweichen. Hier ist also wenig Varianz über Zeit zu erwarten. Stattdessen nutzen die Autoren die Redebeiträge der Abgeordneten, um den Grad der Polarisierung im Parlament zu messen. Dabei verwenden sie einen Teil der Redebeiträge als Trainingsmaterial für verschiedene *Supervised Learning*-Algorithmen, wobei die Parteizugehörigkeit der Redner die korrekte Kategorie darstellt. In einem nächsten Schritt versuchen sie, die Parteizugehörigkeit der nicht im Trainingsdatensatz enthaltenen Reden zu klassifizieren. Den Anteil der korrekt klassifizierten Reden verwenden die Autoren als Maß für die parlamentarische Polarisierung: Je besser die Vorhersagbarkeit der Parteizugehörigkeit auf Basis der Redebeiträge ist, desto höher ist die Polarisierung und umgekehrt. Dem Maß liegt also die Vorstellung zugrunde, dass es in Zeiten hoher Polarisierung weniger Überschneidungen in den Redebeiträgen der beiden großen Parteien gibt, sodass die Algorithmen in den Daten leicht Attribute identifizieren können, die sich eindeutig für die korrekte Klassifizierung eignen.

Bei Interesse an weiteren Anwendungsbeispielen des *Supervised Learnings* sei auf die Arbeiten von Bonica (2018), Colleoni et al. (2014), Diermeier et al. (2011), Mendez (2017) und Yu et al. (2008) verwiesen.

Bevor wir uns dem Beispiel aus der Forschungspraxis zuwenden, sei schließlich noch darauf hingewiesen, dass es freilich auch simplere Textanalyseverfahren gibt, nicht zuletzt solche, die nicht auf statistischen Modellen basieren (Boumans und

Trilling 2016). Bei diktionärsbasierten Verfahren etwa können sowohl thematische als auch weitere Kategorien verwendet werden. So ist zum Beispiel die Sentiment-Analyse häufig nichts anderes (und entsprechend nur leidlich erfolgreich) als die Suche nach einer Reihe von Emotionsbegriffen in Texten (Tumasjan et al. 2011; Young und Soroka 2012; Balmas 2017).

Ebenso hat sich für die Analyse großer Textdaten das sogenannte *Crowdcoding* als hilfreiche Möglichkeit herauskristallisiert. In diesem Fall werden die Daten durch menschliche Kodierer online bearbeitet (Benoit et al. 2016; Haselmayer und Jenny 2017; Lind et al. 2017). Dabei gilt, dass selbst wenn der Textkorpus zu umfassend ist, um vollständig mittels *Crowdcoding* kategorisiert zu werden, dieses Verfahren doch hervorragend für die Bereitstellung von Trainingsdaten für das *Supervised Learning* geeignet ist.

Abschließend sei bemerkt, dass die angedeuteten Verfahren trotz ihres nicht zu leugnenden Potenzials für die sozialwissenschaftliche Forschung keinesfalls die traditionellen Verfahren ersetzen werden, da sich die angedeuteten Techniken für viele typische Fragestellungen nicht eignen. So interessiert uns, wie eingangs bemerkt, zumeist nicht die kategoriale Zugehörigkeit bestimmter Fälle, sondern der Zusammenhang spezifischer Variablen.

Greifen wir zur Illustration dieses Punktes ein weiteres Mal ein Beispiel aus der Survey-Forschung auf. So gibt es etwa in der politischen Soziologie eine Reihe klassischer Modelle, die bestimmte Erwartungen über das Wahlverhalten formulieren (Falter und Schoen 2014). Im Rahmen klassischer statistischer Modelle können wir die erwarteten Zusammenhänge als Regressionsgleichung ausdrücken, um etwa zu prüfen, ob die ideologische Nähe zwischen Wählern und Parteien das Wahlverhalten dominiert oder ob nicht-ideologische Faktoren einen stärkeren Einfluss haben. Viel seltener ist unser Anliegen dagegen, die Befragten auf Basis der Gesamtheit ihrer Survey-Antworten in distinkte Kategorien einzuordnen.

Interessant sind Verfahren aus dem Bereich des *Machine Learnings* deshalb vor allem, wenn wir mit ihrer Hilfe Variablen für theorietestende Analysen generieren können. Uns könnte etwa interessieren, welchen Einfluss soziale Netzwerke auf die politischen Wahrnehmungen von Nutzern haben. Zu diesem Zweck könnten wir beispielsweise ein klassisches Survey verwenden und die Nutzer um Angabe – sofern vorhanden – ihres *Twitter*-Profilnamens bitten. In diesem Fall könnten wir in einem ersten Schritt die thematischen Schwerpunkte des *Newsfeeds* der individuellen Nutzer mit *Machine Learning*-Verfahren auswerten und in einem zweiten Schritt die so generierte Variablen heranziehen, um Effekte auf die wahrgenommene Bedeutung bestimmter gesellschaftlicher Probleme zu untersuchen.

4 Beispiel aus der aktuellen Forschungspraxis

In diesem Abschnitt wollen wir uns mit einem Beispiel aus der aktuellen Forschungsliteratur beschäftigen. Die Arbeit verwendet Programmierschnittstellen (APIs) zur Datensammlung und -aufbereitung und eine der genannten Techniken aus dem Bereich des *Unsupervised Learnings* zur Datenauswertung. Der Schwerpunkt der Darstellung in diesem Abschnitt liegt weniger auf dem substanziellen Ergebnis, als vielmehr auf der Frage, wie die Arbeit in der Praxis repliziert werden könnte.

Der zu diskutierende Artikel „Computer-assisted text analysis for comparative politics“ wurde von Lucas et al. (2015) in der Zeitschrift *Political Analysis* veröffentlicht. Der Aufsatz beschreibt zwei Anwendungsbeispiele des *Structural Topic Models*, von denen wir uns mit dem zweiten beschäftigen wollen. Im Aufbau macht der Text zunächst einige allgemeine Bemerkungen zur Textanalyse und beschreibt dann im Detail einige übliche und weniger übliche Schritte zur Aufbereitung von Text, die der eigentlichen Datenauswertung vorgelagert sind. Wir wollen uns hier nicht weiter mit den Spezifika der vorbereitenden Schritte auseinandersetzen und verweisen für diese Fragen auf entsprechende Einführungstexte und selbstverständlich auch auf den Grundlagentext (Lucas et al. 2015; Welbers et al. 2017; Wilkerson und Casas 2017). Es sei allerdings das von den Autoren angesprochene Problem multilingualer Texte herausgegriffen, welches für das Anwendungsbeispiel von zentraler Bedeutung ist.

Wie im vorangegangenen Abschnitt dargelegt, basieren die meisten in den Sozialwissenschaften aktuell üblichen computergestützten Textanalyseverfahren auf einer Term-Dokument-Matrix. In solchen Matrizen wird schlicht vermerkt, wie häufig bestimmte Begriffe in einem Text auftreten, um daraus beispielsweise Rückschlüsse über den Inhalt der Texte zu ziehen. Wenig überraschend sollte es bei verschiedensprachigen Texten kaum bis keine Überschneidungen hinsichtlich der verwendeten Begriffe geben, sodass Verfahren zur vollautomatischen Themenklassifizierung stets die Sprache als das zentrale Unterscheidungsmerkmal identifizieren würden. Anders gesagt sind sich je ein deutscher und ein englischer Text zum Thema „Wirtschaft“ sowie je ein deutscher und ein englischer Text zum Thema „Innere Sicherheit“ zwingend ähnlicher im Hinblick auf die verwendete Sprache als im Hinblick auf die Substanz.

Um dem Problem der Mehrsprachigkeit zu begegnen, entscheiden sich die Autoren für die automatische Übersetzung des gesammelten Textkorpus ins Englische, um das gewählte Textanalyseverfahren auf einen unilingualen Korpus anwenden zu können. Lucas et al. (2015) unterscheiden hierbei zwischen der automatischen Übersetzung der Volltexte sowie der Einzelbegriffe aus der Term-Dokument-

Matrix. Diese Unterscheidung ist lediglich in praktischer Hinsicht relevant, da sich die Masse des zu übersetzenden Materials dramatisch reduziert, wenn jeder Begriff nur einmal statt viele Male übersetzt werden muss. Für die Übersetzung der Volltexte spricht hingegen, dass die Begriffe in ihrem Kontext übersetzt werden, was die Qualität der Übersetzung verbessern sollte. Da diese Unterscheidung jedoch keinen Einfluss auf das substanzielle Interesse des Textes hat – und nicht einmal auf die technischen Aspekte der Übersetzung –, soll sie uns hier nicht weiter beschäftigen. In der Folge werden wir deshalb ausschließlich das Ergebnis darstellen, welches auf den übersetzten Volltexten beruht.

Interessant ist das Thema der automatischen Übersetzung für uns vor allem deshalb, da die Autoren sich hierzu des Übersetzungsalgorithmus von *Google Translate* bedienen. Dabei ist es selbstverständlich nicht so, dass die Autoren jeden ihrer Texte mühsam in das Freitextfeld kopieren, um das Ergebnis dann wieder herauszukopieren. Stattdessen bedienen sie sich just der Techniken, die wir bereits im zweiten Abschnitt kennengelernt haben. Konkret machen sich die Autoren zunutze, dass *Google* die Funktionalität des Übersetzungsalgorithmus als API zur Verfügung stellt. Wir können also eine Übersetzungsanfrage als URL formuliert an den Server schicken, der auf unsere Anfrage nicht mit einem HTML-Dokument reagiert, sondern das Ergebnis der Übersetzungsaufgabe in einem Datenformat – in diesem Fall dem JSON-Format – bereitstellt. Um beispielsweise die Übersetzung des Satzes „big data ist leichter als erwartet“ zu erhalten, schicken wir die folgende Anfrage an *Google*:

https://www.googleapis.com/language/translate/v2?key=userkey&q=big%20data%20ist%20leichter%20als%20erwartet&source=de&target=en

und erhalten als Antwort das Folgende: „big data is easier than expected.“[2] Mit unserem bereits erworbenen Wissen sind wir in der Lage, die URL in ihre Bestandteile zu zerlegen. Abermals beginnt die URL mit der Festlegung des zu verwendenden Protokolls (HTTPS), gefolgt von der Server-Adresse (www.googleapis.com), dem Pfad (/language/translate/v2) und den Parametern der Anfrage, eingeleitet durch das Fragezeichen. In diesem Fall teilen wir dem Server vier Parameter mit:

[2] Interessierte Leserinnen werden feststellen, dass die Nutzung der *Google*-API etwas stärkeren Beschränkungen unterliegt als die *Wikipedia*-API aus dem Abschn. 2. Wie erwartet erhalten wir eine Antwort im JSON-Format, allerdings teilt uns eine knappe Botschaft mit, dass der API-Schlüssel ungültig ist. Wie bemerkt verlangen viele Datenanbieter eine Registrierung vor der Nutzung eines Datendienstes. Nach der Registrierung bei den Google-Datendiensten wird für den Nutzer ein individueller Schlüssel generiert, der den Wert „userkey“ des Parameters „key“ in dem abgedruckten Link ersetzt. Da uns hier vor allem das Prinzip und nicht die konkrete Umsetzung interessiert, sei zur Generierung eines Schlüssels auf die folgende Seite verwiesen: https://console.cloud.google.com/.

den individuellen Nutzerschlüssel „key“, den zu übersetzenden Satz „q“, die Ursprungssprache „source“ und die Zielsprache „target“.

Die Ersetzung der Leerzeichen durch die Zeichenfolge „%20“ ist die einzige wirkliche Neuheit. Bestimmte Zeichen sind in einer URL unzulässig oder sie haben eine besondere Bedeutung. Sie werden deshalb durch eine allgemein definierte Kombination zulässiger Zeichen ersetzt, ein Leerzeichen also beispielsweise durch die Zeichenfolge „%20“. Ein weiteres Beispiel ist das „&“, welches die einzelnen Anfrageparameter voneinander trennt und deshalb als Parameterwert durch „%26“ ersetzt wird. Der Parameterwert „big%20data%20ist%20leichter%20%26%20hilfreicher%20als%20erwartet“ im Parameter „q“ kann also verwendet werden, um die Übersetzung des Satzes „big data ist leichter & hilfreicher als erwartet“ zu erhalten.

In dem Anwendungsbeispiel interessieren sich Lucas et al. (2015) für die Reaktion in China und in der arabischen Welt auf die Veröffentlichung des US-Spionageprogramms durch Edward Snowden. Zu diesem Zweck erstellen sie eine Datenbank von Posts auf sozialen Medien – *Twitter* für arabische Nachrichten, das chinesische Netzwerk *Sina Weibo* für chinesische Nachrichten –, die den Begriff Snowden enthalten. Die Autoren sammeln Nachrichten zwischen dem 1. und dem 30. Juni 2013.[3] Sie weisen darauf hin, dass sie ihre Datenbasis von der Firma *Crimson Hexagon* bezogen haben, die Datenmaterial von sozialen Medien in großem Stil archiviert. Im Hinblick auf die grundsätzliche Replizierbarkeit der Analyse könnte die Datensammlung auch ohne einen solchen Mittelsmann geschehen, da sowohl *Twitter*[4] als auch *Sina Weibo*[5] ihre Daten über eine API zur Verfügung stellen. In der Praxis ist diese Möglichkeit allerdings begrenzt, da die rückwirkende Suche für ein kostenloses Nutzerkonto zeitlich beschränkt ist. Es ist dagegen ohne Weiteres denkbar und vermutlich auch lohnenswert, die Analyse für ein laufendes Ereignis zu replizieren, wenn die Daten also noch über die APIs verfügbar sind. Wir werden uns an dieser Stelle nicht weiter mit den APIs der beiden Unternehmen beschäftigen, da sich das Prinzip der Ansprache von APIs im Vergleich zum bereits Gesagten nicht wesentlich ändert. Es sei jedoch zumindest darauf hingewiesen, dass auch in diesen Fällen eine Registrierung bei dem jeweiligen Anbieter notwendig ist, um die APIs nutzen zu können.

Für die Analyse der Daten verwenden die Autoren das *Structural Topic Model*, das im Statistikprogramm R im Paket stm implementiert ist und deshalb leicht für

[3] Die erste Veröffentlichung des *Guardian* zum Thema der Spionageprogramme fand am 5. Juni 2013 statt.

[4] https://developer.twitter.com/en.html.

[5] http://open.weibo.com/wiki/API文档_V2/EN.

eine Replikation verwendet werden kann. Lucas et al. (2015) weisen nach, dass trotz der Übersetzung der chinesischen und arabischen Texte ins Englische sprachliche Besonderheiten der Ursprungssprache bei der Übersetzung bestehen bleiben, die das Ergebnis der Analyse unzulässig beeinflussen. Sie argumentieren, dass dieser Effekt im STM aufgefangen werden kann, indem die Ursprungssprache als Kovariat für den Inhalt einbezogen wird. Die Autoren schätzen ein Topic Model mit fünfzehn Themen, konzentrieren sich bei der Analyse aber auf drei Themen, die sie mit den Begriffen „Attack", „Human Rights" und „Asylum" zusammenfassen und wie folgt beschreiben:

> *„[Attack] deals with concerns about the United States attacking one's own country or society. [Human rights] deals with posts about the implication of the Snowden episode for American credibility on issues related to freedom and human rights. [Asylum] concerns news updates about Snowden's movements and whether or not he will be granted asylum and in which country.*" (Lucas et al. 2015, S. 271)

Für die Analyse fragen sie schließlich, ob sich die Diskussion zum Thema Snowden in den nationalen Diskursen systematisch unterscheidet. Hierzu prüfen sie, ob bestimmte Themen im chinesischen oder arabischen Textkorpus überzufällig häufig auftreten. Das Ergebnis dieser Analyse ist in Abb. 2 dargestellt. Die Abbildung

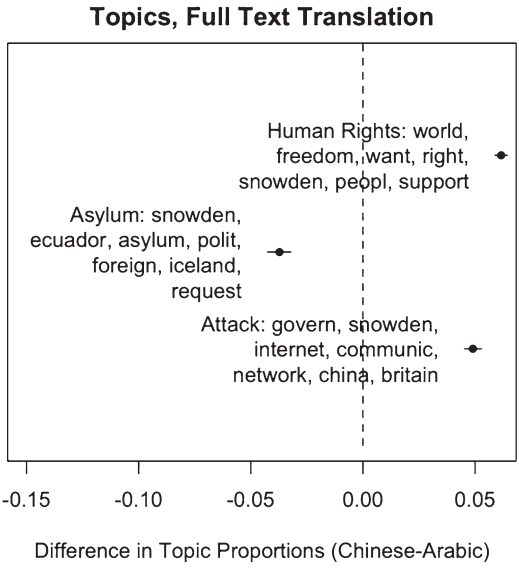

Abb. 2 Themenprävalenz zum Oberthema Edward Snowden in den chinesischen und arabischen sozialen Medien. Quelle: Lucas (2015, Abb. 8)

ist so zu lesen, dass Punkte rechts von der Nulllinie eher in den chinesischen Texten auftreten, Punkte links von der Nulllinie eher in den arabischen Texten. Sie schließen aus dem Ergebnis, dass die Veröffentlichung des US-Spionageprogramms durch Edward Snowden einen stärkeren und negativeren Effekt auf die Wahrnehmung der USA in China hatte, da zum einen ein mögliches Bedrohungsszenario herausgestrichen wurde (Fokus auf das Thema „Attack“) sowie zum anderen der Anspruch der USA als universelle Verteidigerin der Menschenrechte angezweifelt wurde (Fokus auf Thema „Human Rights“).

Während unser Anliegen hier nicht die Bewertung der substanziellen Ergebnisse ist, seien doch zumindest drei Punkte genannt, um die Befunde zu kontextualisieren – nicht zuletzt, da zwei der Punkte unmittelbar mit dem gewählten Textanalyseverfahren zusammenhängen. Wir haben bereits im vorangegangenen Kapitel darauf hingewiesen, dass *Unsupervised Models* stets eine erhebliche Interpretationsleistung und einen erheblichen Interpretationsspielraum mit sich bringen, insbesondere bei der Zuweisung der Labels zu den gefundenen Themen. Die für ein Thema indikativen Begriffe passen häufig nur leidlich gut zusammen und können deshalb nur schlecht auf einen gemeinsamen Nenner gebracht werden. In diesem Sinne ist das, was in den meisten Anwendungen als eindeutige Zuordnung dargestellt wird, eher selten zwingend. Diese Einschränkung ist umso gewichtiger, da in der vorliegenden Anwendung vergleichsweise weitreichende Schlüsse auf Basis des reinen Auftretens bestimmter Themen gezogen werden. Während die Themen also durch kontextlose Begriffe charakterisiert sind, wird das Auftreten bestimmter Themen in der vorliegenden Arbeit mit der Erwartung einer spezifischen Werthaltung im Diskurs verknüpft.

Schließlich ist es hilfreich, sich die Endpunkte der Skala in Abb. 2 zu vergegenwärtigen. Die theoretisch mögliche Skala bewegt sich zwischen −1,0, wenn ein Thema ausschließlich in den chinesischen sozialen Medien verhandelt wird, und +1,0, wenn ein Thema nur in den arabischen Posts auftritt. Vor diesem Hintergrund muss man festhalten, dass ein systematischer Zusammenhang zwischen der Themenprävalenz und dem nationalen Diskurs zwar in einem statistischen Sinne besteht, da die Konfidenzintervalle nicht die Nulllinie schneiden. Dieser Zusammenhang ist aber substantiell nicht so gewichtig, wie es auf den ersten Blick erscheinen mag. Wenn wir uns vor Augen führen, dass die Themen „Attack“ und „Human Rights“ zu etwa 52,5 % in den chinesischen Posts und zu rund 47,5 % in den arabischen Posts auftreten – ausreichend für den dargestellten Unterschied von ungefähr 5 Prozentpunkten –, dann könnte man das Ergebnis durchaus auch so lesen, dass es klare Überschneidungen in den nationalen Diskursen zum Thema Snowden gab.

5 Fazit

Die Digitalisierung aller Lebensbereiche hat zur Folge, dass wir uns gegenwärtig in einem *Golden Age of Data* befinden. An den Sozialwissenschaften ist diese Entwicklung nicht spurlos vorübergegangen. Am deutlichsten drückt sich die Datenaffinität der jüngeren sozialwissenschaftlichen Forschung womöglich in der Etablierung ganz neuer Forschungsfelder aus, die gelegentlich mit dem Begriff der *Computational Social Science* zusammengefasst werden (Alvarez 2016; Blätte et al. 2018). Auch der viel beschworene *Data Scientist* – ein bisschen Informatiker, ein bisschen Statistiker, ein bisschen Sozialwissenschaftler – ist eine offensichtliche Folge der massiven Datenverfügbarkeit. Gerade in diesem Punkt wird die Bedeutung von *Big Data* auch jenseits der Sozialwissenschaften deutlich, da die zunehmend datenorientierte Wirtschaft händeringend nach Fachkräften sucht, die technische Fähigkeiten und sozialwissenschaftliches Denken verknüpfen können.

Für die Sozialwissenschaften folgt daraus zweierlei. Zum einen sind wir gefordert, uns mit den einschlägigen Techniken vertraut zu machen, um den wandelnden Standards sozialwissenschaftlicher Forschung gerecht zu werden. Zum anderen sollten wir auch den wissenschaftlichen Nachwuchs in diesen Fragen ausbilden, um ihm Karrierepotenziale innerhalb wie außerhalb der Wissenschaft zu eröffnen (Munzert 2014, 2018). Das vorliegende Kapitel versteht sich als Teil dieser Bemühungen. Es wurde versucht, die Möglichkeiten und spezifischen Techniken zur Sammlung und Auswertung großer Datenbestände anzudeuten und darzustellen, wie eine vertiefende Auseinandersetzung mit diesen Themen stattfinden kann.

In diesem letzten Abschnitt soll schließlich eine kurze Bewertung der dargestellten Techniken in drei Punkten vorgenommen werden. Dabei gilt zunächst, dass *Big Data* weder Allheilmittel noch für jede Forschungsfrage geeignet ist. Insbesondere ist die beizeiten geäußerte Vorstellung, *Big Data* mache Theoriearbeit obsolet – die Daten mögen für sich sprechen –, verfehlt. Im besten Fall liefern theoriefreie Analysen interessante Spielereien, die jedoch zu keiner relevanten sozialwissenschaftlichen Fragestellung sprechen. Im schlimmeren Fall führen solche Anwendungen zu Fehlschlüssen, wenn die zugrunde liegenden Mechanismen nicht ausreichend durchdacht wurden. Dabei sollten wir uns gerade bei Web-Daten die Frage stellen, welche datengenerierenden Mechanismen den Beobachtungen zugrunde liegen und wie es um die Qualität der gewonnenen Daten bestellt ist. Große Datenmengen stellen keinesfalls sicher, dass das zu untersuchende Phänomen korrekt abgebildet wird.

Besonders einleuchtend ist dieser Punkt bei der Analyse von sozialen Netzwerken. Wenig überraschend stellen die Nutzer sozialer Netzwerke keinen repräsenta-

tiven Bevölkerungsquerschnitt dar, was Rückschlüsse von Präferenzäußerungen auf Einstellungsmuster in der Bevölkerung zumindest zweifelhaft erscheinen lässt (Mellon und Prosser 2017). Mehr noch – weder sollten wir erwarten, dass sich die Neigung, Mitteilungen in einem sozialen Netzwerk zu teilen, zufällig über die ohnehin schon verzerrte Nutzerbasis verteilt, noch ist klar, inwiefern öffentliche Äußerungen mit privaten Präferenzen zusammenhängen. Diese Punkte unterstreichen deutlich, dass der Validitätsnachweis für Web-Daten umso dringender geführt werden muss. Ein bekanntes Beispiel für die Unzuverlässigkeit sozialer Medien für die Analyse öffentlicher Meinung ist die Arbeit von Tumasjan et al. (2011). Die Autoren zeigen, dass der Anteil der Parteinennungen auf *Twitter* überraschend deutlich mit dem Wahlergebnis der Parteien bei der Bundestagswahl 2009 zusammenhängt. In einer Replik auf diese Arbeit nehmen Jungherr et al. (2012) die Piratenpartei in den Datensatz auf und finden, dass sich das Wahlergebnis bei Berücksichtigung der Piraten ganz erheblich von den *Twitter*-Nennungen unterscheidet. An diesem Punkt wird deutlich, dass sich Nutzer einem Medium nicht zufällig zuwenden.

Mit der Frage, ob große Daten das interessierende Phänomen abbilden, geht die Frage einher, inwiefern große Daten überhaupt notwendig sind, um eine bestimmte Fragestellung zu untersuchen. Gerade aufgrund der leichten Zugänglichkeit von Web-Daten ist unser erster Impuls häufig die Sammlung und Auswertung aller verfügbaren Daten. Während es zweifellos Anwendungen gibt, die eine Vollerhebung erforderlich machen, scheint es sinnvoll, sich vor der Datensammlung zunächst die Frage zu stellen, ob eine Stichprobe nicht den gleichen Erkenntniswert haben kann. Dabei mag man einwenden, dass das Ziehen einer Stichprobe bei webbasierten Daten in Unkenntnis der Grundgesamtheit nicht ganz trivial ist. Dieser Einwand unterstreicht allerdings lediglich die bereits angesprochene Validitätsproblematik und weniger den Wert einer Vollerhebung.

Drittens sei schließlich angesprochen, dass die sozialwissenschaftliche Forschungsagenda und die Datenverfügbarkeit häufig auseinanderfallen. Während bestimmte Bereiche durch einen außerordentlichen Datenreichtum gekennzeichnet sind, gibt es für andere Fragen nur wenig Datenmaterial. Somit liegt die Sorge auf der Hand, dass die Datenverfügbarkeit die Forschungsagenda treibt und relevante Fragen unbearbeitet bleiben. Dass diese Sorge nicht völlig unbegründet ist, können wir in der Forschung zu sozialen Medien beobachten. Die Plattform *Twitter* dominiert diese Literatur nicht deshalb, weil es von der Forschung als das einzig relevante Medium identifiziert wurde, sondern weil sich der Datenzugang hier deutlich leichter gestaltet als etwa bei *Facebook*.

Diesen Vorbehalten zum Trotz gilt jedoch, dass *Big Data* ganz erhebliche Potenziale für die Sozialwissenschaften birgt und uns den systematischen Zugriff auf Forschungsfragen ermöglicht, die noch vor wenigen Jahren nur schwerlich hätten

bearbeitet werden können. Nicht zuletzt ist dabei von enormem Vorteil, dass die Potenziale von *Big Data* auch durch individuelle Forscher ohne große Forschungsbudgets genutzt werden können – und hoffentlich hat dieses Kapitel dazu beigetragen, einige dieser Potenziale aufzuzeigen.

Literatur

Alvarez, Michael R., Hrsg. 2016. *Computational social science: Discovery and prediction*. Cambridge: Cambridge University Press.

Balmas, Meital. 2017. Bad news: The changing coverage of national leaders in foreign media of Western democracies. *Mass Communication and Society* 20(5): 663–685.

Barberà, Pablo. 2015. Birds of the same feather tweet together: Bayesian ideal point estimation using Twitter data. *Political Analysis* 23(1): 76–91.

Bauer, Paul C., Pablo Barberà, Kathrin Ackermann, und Aaron Venetz. 2017. Is the left-right scale a valid measure of ideology? Individual-level variation in associations with „left" and „right" and left-right self-placement. *Political Behavior* 39(3): 553–583.

Beauchamp, Nicholas. 2017. Predicting and interpolating state-level polls using Twitter textual data. *American Journal of Political Science* 61(2): 490–503.

Benoit, Kenneth, Drew Conway, Benjamin E. Lauderdale, und Michael Laver. 2016. Crowd-sourced text analysis: Reproducible and agile production of political data. *American Political Science Review* 110(2): 278–295.

Blätte, Andreas, Joachim Behnke, Kai-Uwe Schnapp, und Claudius Wagemann, Hrsg. 2018. *Computational Social Science. Die Analyse von Big Data*. Baden-Baden: Nomos.

Blei, David M., Andrew Y. Ng, und Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning* 3(4–5): 993–1022.

Bonica, Adam. 2018. Inferring roll-call scores from campaign contributions using supervised machine learning. *American Journal of Political Science* 62(4): 830–848.

Boumans, Jelle W., und Damian Trilling. 2016. Taking stock of the toolkit: An overview of relevant automated content analysis approaches and techniques for digital journalism scholars. *Digital Journalism* 4(1): 8–23.

Cederman, Lars-Erik, Nils B. Weidmann, und Nils-Christian Bormann. 2015. Triangulating horizontal inequality: Toward improved conflict analysis. *Journal of Peace Research* 52(6): 806–821.

Ceron, Andrea, Luigi Curini, Stefano M. Iacus, und Giuseppe Porro. 2014. Every tweet counts? How sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to Italy and France. *New Media and Society* 16(2): 340–358.

Colleoni, Elanor, Alessandro Rozza, und Adam Arvidsson. 2014. Echo chamber or public sphere? Predicting political orientation and measuring political homophily in Twitter using big data. *Journal of Communication* 64(2): 317–332.

Couper, Mick P., und Frauke Kreuter. 2013. Using paradata to explore item level response times in surveys. *Journal of the Royal Statistical Society A* 176(1): 271–286.

D'Orazio, Vito, Steven Landis, Glenn Palmer, und Philip Schrodt. 2014. Separating the wheat from the chaff: Applications of automated document classification using Support Vector Machines. *Political Analysis* 22(2): 224–242.

Denny, Matthew J., und Arthur Spirling. 2018. Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it. *Political Analysis* 26(2): 168–189.

Diermeier, Daniel, Jean-François Godbout, Yu Bei, und Stefan Kaufmann. 2011. Language and ideology in Congress. *British Journal of Political Science* 42(1): 31–55.

Egan, Patrick J., und Megan Mullin. 2012. Turning personal experience into political attitudes: The effect of local weather on American's perceptions about global warming. *Journal of Politics* 74(3): 796–809.

Falter, Jürgen W., und Harald Schoen, Hrsg. 2014. *Handbuch Wahlforschung*. Wiesbaden: Springer VS.

Felderer, Barbara, Alexandra Birg, und Frauke Kreuter. 2014. Paradaten. In *Handbuch Methoden der empirischen Sozialforschung*, Hrsg. Nina Baur und Jörg Blasius, 357–365. Wiesbaden: Springer VS.

Grimmer, Justin. 2010. A Bayesian hierarchical topic model for political texts: Measuring expressed agendas in Senate press releases. *Political Analysis* 18(1): 1–35.

Haselmayer, Martin, und Marcelo Jenny. 2017. Sentiment analysis of political communication: Combining a dictionary approach with crowdcoding. *Quality and Quantity* 51(6): 2623–2646.

Heerwegh, Dirk. 2003. Explaining response latencies and changing answers using client-side paradata from a web survey. *Social Science Computer Review* 21(3): 360–373.

Hersh, Eitan D., und Clayton Nall. 2016. The primacy of race in the geography of income-based voting: new evidence from public voting records. *American Journal of Political Science* 60(2): 289–303.

Jacobi, Carina, Wouter van Atteveldt, und Kasper Welbers. 2016. Quantitative analysis of large amounts of journalistic texts using topic modelling. *Digital Journalism* 4(1): 89–106.

Jankowski, Michael, Sebastian H. Schneider, und Markus Tepe. 2019. „... Deutschland eben". Eine Analyse zur Interpretation des Begriffs „rechts" durch Bundestagskandidaten auf Grundlage von Structural Topic Models. In *Identität – Identifikation – Ideologie. Analysen zu politischen Einstellungen und politischem Verhalten in Deutschland*, Hrsg. Markus Steinbrecher, Evelyn Bytzek und Ulrich Rosar, 141–179. Wiesbaden: Springer VS.

Jungherr, Andreas, Pascal Jürgens, und Harald Schoen. 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, M. „Predicting elections with Twitter: What 140 characters reveal about political sentiment. *Social Science Computer Review* 30(2): 229–234.

King, Gary, Jennifer Pan, und Margaret E. Roberts. 2013. How censorship in China allows government criticism but silences collective expression. *American Political Science Review* 107(2): 326–334.

King, Gary, Jennifer Pan, und Margaret E. Roberts. 2017. How the Chinese government fabricates social media posts for strategic distraction, not engaged argument. *American Political Science Review* 111(3): 484–501.

Kreuter, Frauke. 2013. *Improving surveys with paradata: Analytic uses of process information*. Hoboken: Wiley.

Kuhn, Patrick M., und und Nils B. Weidmann. 2015. Unequal we fight: Between- and within-group inequality and ethnic civil war. *Political Science Research and Methods* 3(3): 534–568.

Lauderdale, Benjamin E., und Tom S. Clark. 2014. Scaling meaningful political dimensions using texts and votes. *American Journal of Political Science* 58(3): 754–771.

Lind, Fabienne, Maria Gruber, und Hajo G. Boomgaarden. 2017. Content analysis by the crowd: Assessing the usability of crowdsourcing for coding latent constructs. *Communication Methods and Measures* 11(3): 191–209.

Lucas, Christopher, et al. 2015. Computer-assisted text analysis for comparative politics. *Political Analysis* 23(2): 254–277.

Mellon, Jonathan, und Christopher Prosser. 2017. Twitter and Facebook are not representative of the general population: Political attitudes and demographics of British social media users. *Research and Politics.* https://doi.org/10.1177/2053168017720008.

Mendez, Fernando. 2017. Modeling proximity and directional decisional logic: What can we learn from applying statistical learning techniques to VAA-generated data? *Journal of Elections, Public Opinion and Parties* 27(1): 31–55.

Mitchell, Ryan. 2015. *Web scraping with Python: Collecting data from the modern web*. Beijing: O'Reilly.

Monroe, Burt L. 2013. The five Vs of big data political science: Introduction to the virtual issue on big data in political science. *Political Analysis* 21(V5): 1–9.

Munzert, Simon. 2014. Big Data in der Forschung! Big Data in der Lehre? Ein Vorschlag zur Erweiterung der bestehenden Methodenausbildung. *Zeitschrift für Politikwissenschaft* 24(1–2): 205–220.

Munzert, Simon. 2018. Auf dem Weg zu einer fundierten Softwareausbildung in der Politikwissenschaft. In *Computational Social Science. Die Analyse von Big Data*, Hrsg. Andreas Blätte, Joachim Behnke, Kai-Uwe Schnapp und Claudius Wagemann, 379–402. Baden-Baden: Nomos.

Munzert, Simon, und Dominic Nyhuis. Im Erscheinen. Die Nutzung von Webdaten in den Sozialwissenschaften. In *Handbuch Methoden der Politikwissenschaft*, Hrsg. Claudius Wagemann, Achim Goerres und Markus Siewert. Wiesbaden: Springer VS.

Munzert, Simon, Christian Rubba, Peter Meißner, und Dominic Nyhuis. 2014. *Automated web data collection with R: A practical guide to web scraping and text mining*. Hoboken: Wiley.

Nolan, Deborah, und Duncan Temple Lang. 2014. *XML and web technologies for data sciences with R*. New York: Springer.

Peterson, Andrew, und Arthur Spirling. 2018. Classification accuracy as a substantive quantity of interest: Measuring polarization in Westminster systems. *Political Analysis* 26(1): 120–128.

Quinn, Kevin M., Burt L. Monroe, Michael Colaresi, Michael H. Crespin, und Dragomir R. Radev. 2010. How to analyze political attention with minimal assumptions and costs. *American Journal of Political Science* 54(1): 209–228.

Roberts, Margaret E., et al. 2014. Structural topic models for open-ended survey responses. *American Journal of Political Science* 58(4): 1064–1082.

Rothschild, Jacob E., Adam J. Howat, Richard M. Shafranek, und Ethan C. Busby. 2019. Pigeonholing partisans: Stereotypes of party supporters and partisan polarization. *Political Behavior* 41(2): 423–443.

Tumasjan, Andranik, Timm O. Sprenger, Philipp G. Sander, und Isabell M. Welpe. 2011. Election forecasts with Twitter: How 140 characters reflect the political landscape. *Social Science Computer Review* 29(4): 402–418.

Tzelgov, Eitan. 2014. Cross-cutting issues, intraparty dissent and party strategy: The issue of European integration in the House of Commons. *European Union Politics* 15(1): 3–23.

Weidmann, Nils B., und Sebastian Schutte. 2017. Using night light emissions for the prediction of local wealth. *Journal of Peace Research* 54(2): 125–140.

Welbers, Kasper, Wouter van Atteveldt, und Kenneth Benoit. 2017. Text analysis in R. *Communication Methods and Measures* 11(4): 245–265.

Wilkerson, John, und Andreu Casas. 2017. Large-scale computerized text analsis in political science: Opportunities and challenges. *Annual Review of Political Science* 20: 529–544.

Young, Lori, und Stuart Soroka. 2012. Affective news: The automated coding of sentiment in political texts. *Political Communication* 29(2): 205–231.

Yu, Bei, Stefan Kaufmann, und Daniel Diermeier. 2008. Classifying party affiliation from political speech. *Journal of Information Technology and Politics* 5(1): 33–48.