

BRAT Rapid Annotation Tool

Christian Strippel / Laura Laugwitz / Sünje Paasch-Colberg / Katharina Esau / Annett Heft*

*Im Kontext interdisziplinärer Zusammenarbeit insbesondere mit Kolleg*innen aus der Informatik wird die Kommunikations- und Medienforschung seit geraumer Zeit mit einer großen Bandbreite an Forschungssoftware konfrontiert, mit der sie bislang nur wenig Erfahrungen hat. Neben Programmiersprachen wie Python oder R sind dies etwa spezifische Tools zur Textanalyse, die eine Alternative zu bisherigen Varianten der computergestützten Inhaltsanalyse darstellen. Mit dem brat rapid annotation tool (BRAT) stellen wir in diesem Beitrag eine solche Alternative vor und rezensieren sie vor dem Hintergrund unserer Erfahrungen im Umgang mit ihr. BRAT ist ein webbasiertes Open-Source-Tool zur Textannotation, das vor rund zehn Jahren von einem internationalen Team aus Informatiker*innen entwickelt wurde. Der Beitrag stellt das Tool und seine wichtigsten Funktionen vor, zeigt anhand von drei Fallstudien mögliche Anwendungsbeispiele zu dessen Einsatz in qualitativen und quantitativen Inhaltsanalysen auf und bewertet es schließlich mit Blick auf entsprechende Potenziale und Schwierigkeiten.*

Schlagwörter: Forschungssoftware, Inhaltsanalyse, Textanalyse, Textannotation, Rezension

BRAT Rapid Annotation Tool

In the context of interdisciplinary collaboration, especially with colleagues from computer science, communication and media research has for some time been confronted with a wide range of research software with which it has had little prior experience. In addition to programming languages such as Python or R, these include specific tools for text analysis that represent an alternative to previous variants of computer-assisted content analysis. With the brat rapid annotation tool (BRAT) we present such an alternative in this paper and review it against the background of our experience in using it. BRAT is a web-based open-source text annotation tool that was developed by an international team of computer scientists about ten years ago. The article introduces the tool and its most important features, presents examples for its use in qualitative and quantitative

* Christian Strippel, M. A., Weizenbaum-Institut für die vernetzte Gesellschaft, Hardenbergstraße 32, 10623 Berlin, Deutschland, christian.strippel@weizenbaum-institut.de, ORCID: <https://orcid.org/0000-0002-7465-4918>;

Laura Laugwitz, M. Sc., Universität Hamburg, Von-Melle-Park 5, 20146 Hamburg, Deutschland, laura.laugwitz@uni-hamburg.de, ORCID: <https://orcid.org/0000-0001-8527-2504>;

Dr. Sünje Paasch-Colberg, Deutsches Zentrum für Integrations- und Migrationsforschung (DeZIM), Mauerstraße 76, 10117 Berlin, Deutschland, paasch-colberg@dezim-institut.de, ORCID: <https://orcid.org/0000-0002-0771-9646>;

Dr. Katharina Esau, Queensland University of Technology, Digital Media Research Centre, Creative Industries Precinct Z9, Kelvin Grove, QLD 4059, Brisbane, Australia, katharina.esau@qut.edu.au, ORCID: <https://orcid.org/0000-0001-7450-6010>;

Dr. Annett Heft, Weizenbaum-Institut für die vernetzte Gesellschaft, Berlin, und Freie Universität Berlin, Institut für Publizistik- und Kommunikationswissenschaft, Garystr. 55, 14195 Berlin, Deutschland, annett.heft@fu-berlin.de, ORCID: <https://orcid.org/0000-0001-6637-795X>.

content analyses on the basis of three case studies, and finally evaluates it with regard to potentials and difficulties for the field.

Keywords: research software, content analysis, text analysis, text annotation, software review

1. Einleitung

Im Zuge der verstärkten interdisziplinären Zusammenarbeit mit Kolleg*innen aus der Informatik werden aktuell eine Vielzahl unterschiedlicher Software-Anwendungen an die Kommunikations- und Medienforschung herangetragen (Hepp et al., 2021a). Neben Programmiersprachen wie Python oder R, auf die „umgestellt“ zu haben fast schon zu einer Frage der guten wissenschaftlichen Praxis geworden ist, sind dies etwa Tools zur Textanalyse, die eine ernstzunehmende Alternative zu jener Software darstellen, mit der im Fach seit Jahren Inhaltsanalysen maschinenlesbar dokumentiert werden: Microsoft Excel (Feiks, 2016). Auch wenn das Tabellenkalkulationsprogramm nicht gleich das erste Tool ist, an das wir bei Wörtern wie „Forschungssoftware“ oder „computational turn“ (Hepp et al., 2021b) denken, es im Gegenteil sogar wie ein Werkzeug aus einer fernen, fast schon vor-digitalen Ära erscheint, so müssen sich neue Anwendungen doch gerade als Alternativen zu eben diesem Oldtimer bewähren.

Mit dem *brat rapid annotation tool* (BRAT) stellen wir in diesem Beitrag eine solche Alternative vor und rezensieren sie vor dem Hintergrund unserer Erfahrungen im Umgang mit ihr. BRAT ist ein webbasiertes Open-Source-Tool zur Textannotation, das vor rund zehn Jahren von einem internationalen Team aus Informatiker*innen entwickelt wurde (Stenetorp et al., 2012). Was unter „Textannotation“ zu verstehen ist, wie sie sich zu gängigen Verfahren der Inhaltsanalyse verhält und welche Funktionen BRAT dafür bereitstellt, um diese Fragen geht es dabei zuerst. Es folgen Anwendungsbeispiele aus drei Forschungsprojekten, an denen wir in der Vergangenheit beteiligt waren und die uns schließlich als Grundlage zur Bewertung des Tools dienen. Im Fazit des Beitrags fassen wir die Potenziale und Schwierigkeiten der Software noch einmal zusammen und empfehlen konkrete Einsatzmöglichkeiten in der Kommunikations- und Medienforschung. Unser Ziel ist dabei, Abwägungen über die Arbeit mit diesem oder ähnlichen Tools im Vorfeld zukünftiger Projekte zu erleichtern.

2. Beschreibung der Software

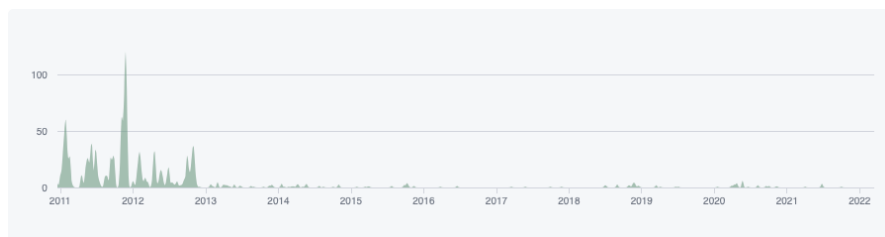
BRAT ist ein browserbasiertes Tool mit visueller Nutzeroberfläche zur strukturierten und kollaborativen Annotation von Texten in allen Sprachen. Browserbasiert meint dabei, dass BRAT über einen Server bereitgestellt und mithilfe eines Internet-Browsers bedient wird. Strukturierte Textannotation bedeutet, dass Textdokumente – oder Bestandteile davon – markiert und mit zuvor festgelegten Labels versehen werden. Es handelt sich also um eine Kombination aus induktivem Textlabeling, wie wir es aus der qualitativen Inhaltsanalyse kennen, und deduktiver Zuweisung zuvor festgelegter Kategorien wie bei der Codierung in standardisierten Inhaltsanalysen. Kollaborative Annotation heißt schließlich, dass mehrere Personen gleichzeitig an einem Dokument arbeiten und die Änderungen der anderen Annotierenden¹ in Echtzeit verfolgen können.

1 In der inhaltsanalytischen Kommunikations- und Medienforschung wird die Technik des Zuweisens von Kategorien zu Textsegmenten klassischerweise „Codieren“ bzw. „Codierung“

Dadurch ist BRAT vergleichbar mit etablierten Programmen zur computergestützten, qualitativen Text- bzw. Inhaltsanalyse wie Atlas.ti, NVivo oder MAXQDA sowie mit anderen Tools zur Textannotation wie WebAnno (Eckart de Castilho et al., 2016), INCEption (Klie et al., 2018) oder LightTag (Perry, 2021). Von Ersteren unterscheidet sich BRAT dadurch, dass es unter einer Open-Source-Lizenz (MIT) veröffentlicht wurde und so kostenlos genutzt und ohne Einschränkung kopiert, modifiziert, verteilt, unterlizenziert oder gar verkauft werden kann. Von Letzteren hebt sich BRAT dadurch ab, dass es auch über zehn Jahre nach seiner Entwicklung noch immer „one of the most popular tools for the manual annotation of documents“ (Neves & Ševa, 2021, S. 153) ist; und einige der aktuelleren Tools zum Teil sogar auf dessen Code basieren (Biemann et al., 2017, S. 245).

Auf ihrer englischsprachigen Webseite² stellen die Entwickler*innen – neben der Möglichkeit zum Herunterladen der Software – ein ausführliches Manual, Nutzungsbeispiele, eine Demo-Version, Tutorials für die Installation und Konfiguration, Erläuterungen zu den Features sowie zahlreiche Screenshots zur visuellen Dokumentation bereit. Über die Zeit ist so eine kleine Community entstanden, die auf GitHub³ auf Probleme der Software hinweist, Änderungen vorschlägt und – wenn auch nur in begrenztem Maße – Anpassungen vornimmt (siehe Abb. 1).

Abbildung 1: Anpassungen (Git Commits) für BRAT seit 2011



Quelle: <https://github.com/nlplab/brat/graphs/contributors>

2.1 Features

Hinsichtlich der Features lassen sich bei BRAT die Stammfunktionen („primitives“), also die Basis-Features zur Annotation, von weiteren Features zur Verbesserung der Nutzerfreundlichkeit unterscheiden. Die Stammfunktionen umfassen dabei die folgenden sechs Features:

1. Das *Annotieren* („text span annotation“), das aufgrund der visuellen Nutzerfläche von BRAT ähnlich wie die Kommentarfunktion von Microsoft Word funktioniert: Ein zu annotierender Textabschnitt wird markiert und durch Auswahl einer der zuvor festgelegten Kategorien mit einem Label versehen (siehe Abb. 2).

genannt und die entsprechenden Personen werden als „Codierer*innen“ bezeichnet. In der Informatik wird unter diesen Begriffen allerdings etwas anderes verstanden, weshalb BRAT nicht als Codier-, sondern als Annotationstool bezeichnet wird. Vor diesem Hintergrund arbeiten auch wir im Folgenden mit den Begriffen „Annotation“ und „Annotierende“. Forschungspraktisch können diese Bezeichnungen aber synonym verstanden werden.

2 <https://brat.nlplab.org>.

3 <https://github.com/nlplab/brat>.

Definiert werden diese Textstellen als „Entitäten“ („entities“). In der Computerlinguistik dient die Definition solcher Entitäten beispielsweise der „Named-entity recognition“, mit der Algorithmen lernen, welche Eigennamen in einem Text zusammengehören (z. B. „Angela“ und „Merkel“).

2. Durch *Relationieren* („relation annotation“) lassen sich annotierte Entitäten dann in Beziehung zueinander setzen. Dabei wird per Drag-and-drop eine Verbindung zwischen zwei Entitäten gezogen und diese Relation dann – ebenfalls anhand zuvor festgelegter Kategorien – mit einem Label versehen (siehe Abb. 4). Diese Relationen können dabei gerichtet oder ungerichtet sein. Auf diese Weise können Algorithmen später lernen, in welcher Beziehung bestimmte Entitäten bzw. Wörter miteinander stehen und wann etwa ein Pronomen im Text welche vorher erwähnte Person meint.
3. Durch die Definition von *Ereignissen* („event annotation“) können über binäre Relationen hinaus auch mehrstufige Beziehungen („n-ary associations“) annotiert werden. Dazu werden mehrere Entitäten verbunden und die Relationen je mit einer spezifischen Rolle in diesem Ereignis versehen (siehe Abb. 4). Um unvollständig definierte Ereignisse zu vermeiden, kann im Vorfeld bestimmt werden, welche Elemente obligatorisch definiert sein müssen. Diese Vorgaben („constraints“) werden dann während der Eingabe automatisch überprüft und es wird auf unvollständige oder fehlerhafte Ereignisse hingewiesen.
4. Mit Hilfe von *Attributen* („attributes“), die ebenfalls im Vorfeld festgelegt werden müssen, lassen sich alle annotierten Entitäten, Relationen und Ereignisse weiter qualifizieren (siehe Abb. 3). Dabei kann sowohl mit binären Angaben (realisiert über Checkboxes) als auch mit der Auswahl aus mehreren Ausprägungen (über Dropdown-Menüs) gearbeitet werden.
5. Für jede Annotation besteht darüber hinaus die Möglichkeit, *offene Anmerkungen* zu ergänzen. Im Prinzip kann diese Funktion zur offenen, induktiven Textannotation genutzt werden, in Kombination mit den anderen genannten Features können hier aber vor allem Unsicherheiten bei den vorgenommenen Annotationen ergänzt werden.
6. Schließlich wird die *Normalisierung von Entitäten* durch sogenannte „normalization features“ unterstützt. Die als Entitäten annotierten Textstellen (z. B. „Bundeskanzlerin“) lassen sich mit den Einträgen realer Entitäten („real-world entities“; hier: „Angela Merkel“) in externen Ressourcen wie der Wikipedia oder dem Google Knowledge Graph assoziieren. Auf diese Weise werden die entsprechenden Textstellen konkretisiert und mit weiteren Informationen über diese Entitäten verknüpft und angereichert.

Daneben gibt es eine Vielzahl weiterer Features, die die Nutzung des Tools unterstützen und vereinfachen. Dazu zählen:

- das *automatische Speichern* aller Annotationen durch unverzügliche Übermittlung aller Veränderungen an den Server;
- das *detaillierte Monitoring* durch Protokollierung aller Vorgänge im Annotationsprozess;
- der direkte *Vergleich von Annotationen* durch eine „side-by-side visualization“;
- eine *Suchfunktion* für alle Textdokumente und Annotationen;
- das *Verlinken von Annotationen* durch eigene Uniform Resource Identifiers und
- die *Unterstützung der manuellen Annotation durch automatisierte Verfahren*, etwa durch externe Tools oder halbautomatisiert durch ein integriertes Lernverfahren („rapid mode“).

2.2 *Installation und Konfiguration*

Um BRAT einmal auszuprobieren, kann die Demo-Version auf der Webseite der Entwickler*innen genutzt werden; für eine projektbezogene Anwendung muss allerdings eine eigene Instanz des Tools aufgesetzt werden. Dafür wird BRAT von der Webseite heruntergeladen und auf einem Server installiert und konfiguriert. Von den Annotierenden wird es dann über einen Internet-Browser (empfohlen werden Chrome oder Safari) genutzt. Diese Client-Server-Architektur ist klassisch für Webanwendungen und bietet den Vorteil, dass BRAT von den Annotierenden auf beliebigen (Desktop-)Umgebungen aufgerufen werden kann. Mobile Geräte mit berührungsempfindlichen Bildschirmen wie Smartphones oder Tablets werden bislang allerdings nicht unterstützt.

Serverseitig ist BRAT ein Python-Programm, das auf einer UNIX-ähnlichen Umgebung läuft, im besten Fall auf einem Webserver, der CGI (Common Gateway Interface) unterstützt (z. B. Apache). Die Installation kann also im Rahmen bestehender Infrastrukturen auf Servern einer Universität oder Forschungseinrichtung oder auf dafür gemieteten (Cloud-)Servern erfolgen. Das Aufsetzen der Anwendung geschieht einmalig, sodass dafür auch IT-Expert*innen und Admins angefragt werden können. Der Arbeitsaufwand für Informatiker*innen sollte nicht mehr als einen halben Arbeitstag betragen. Doch auch technisch versierte Nicht-Informatiker*innen können der englischsprachigen Installationsanleitung⁴ folgen oder das Installationskript nutzen.

Nach der Installation von BRAT kann die laufende Anwendung über eine URL aufgerufen werden. Daraufhin liefert die Anwendung zunächst nur Testdaten, die projektspezifischen Daten und Annotationsmöglichkeiten müssen also erst noch konfiguriert werden. Dafür wird in der Datei „config.py“ auf dem Server ein User angelegt, der über Admin-Rechte in dem BRAT-Verzeichnis verfügt. Die Konfiguration verläuft dann über ein Terminal⁵ und erfahrungsgemäß iterativ. Forschende, die sich im Terminal nur begrenzt sicher fühlen, können alle Textdateien auch auf dem eigenen Rechner mit einem Editor erstellen und müssen diese dann lediglich über das Terminal in die entsprechenden Ordner kopieren.

Konfiguriert werden vier Dimensionen in je einer gesonderten Textdatei: die Annotationstypen (annotation.conf), die visuelle Darstellung (visual.conf), technische Erweiterungen (tools.conf) und Tastatur-Shortcuts (kb_shortcuts.conf).⁶ Das projektspezifische Untersuchungsmaterial wird in Form einzelner Textdateien in UTF-8-Codierung⁷ mit der Endung „.txt“ (z. B. „artikel1.txt“, „artikel2.txt“) in den „data“-Ordner der Anwendung importiert. Dabei bietet es sich an, separate Ordner und User-Accounts für die Annotierenden zu erstellen. Des Weiteren wird analog zu jeder angelegten Textdatei eine weitere, leere Textdatei mit demselben Namen und der Dateierweiterung „.ann“ erstellt und hochgeladen (Beispiel: „artikel1.txt“ und „artikel1.ann“). In diesen Dateien werden die Annotationen später abgespeichert und für die Analyse dokumentiert.

4 <https://brat.nlplab.org/installation.html>.

5 Das Terminal bietet mit seinem „command-line interface“ die Möglichkeit, textbasiert Befehle an einen Computer oder ein Programm weiterzugeben. Das ermöglicht in der Regel mehr Funktionen als das graphische Interface, mit dem heutzutage die meisten Computer bedient werden.

6 Eine Anleitung zur Konfiguration findet sich hier: <https://brat.nlplab.org/configuration.html>.

7 UTF-8 ist die am weitesten verbreitete Codierung für Unicode-Zeichen.

3. Anwendungsbeispiele

Unsere Einschätzung, welche Potenziale und Herausforderungen BRAT für Inhaltsanalysen bereithält, stützt sich im Wesentlichen auf unsere Erfahrungen aus drei Studien, in denen wir mit diesem Tool gearbeitet haben. Untersucht wurden in diesen Studien (1) die Wirkungsbeziehung zwischen Kommunikationsformen und Deliberationsdynamik in Online-Diskussionsthreads auf Teilnehmungsplattformen, (2) Hassrede in Nutzerkommentaren zum Thema Flucht und Migration sowie (3) Anti-Elitismus in der digitalen Kommunikation rechtspopulistischer und rechtsradikaler Parteien im Kontext der Wahlen 2019 zum Europaparlament.

3.1 Studie 1: *Kommunikationsformen und Deliberationsdynamik in Diskussionsthreads auf Online-Teilnehmungsplattformen (2016–2020)*

Bei der ersten Studie handelt es sich um eine quantitative, relationale Inhaltsanalyse von deutschsprachigen Nutzerkommentaren auf Online-Teilnehmungsplattformen. Konkret ging es um Verfahren zur Beteiligung von Bürger*innen in Berlin zur Gestaltung des Tempelhofer Feldes sowie in Nordrhein-Westfalen zur Zukunft des Braunkohleabbaus. Im Fokus der Analyse standen die Wirkungsbeziehungen zwischen unterschiedlichen Kommunikationsformen (z. B. Argumentation, Narration) in den Ausgangskommentaren und Formen der Reziprozität im weiteren Diskussionsverlauf (z. B. Quantität und Qualität der Antwort-Kommentare). Die Studie war dabei Teil eines Dissertationsprojekts an der Heinrich-Heine-Universität Düsseldorf im Rahmen des NRW-Fortschrittskollegs Online-Partizipation (Esau, 2018, 2022; Esau & Friess, 2022) sowie einer interdisziplinären Zusammenarbeit zwischen den Instituten Informatik und Sozialwissenschaften (Escher et al., 2016; Liebeck et al., 2016, 2017).

Für die Durchführung der relationalen Inhaltsanalyse wurde sich für BRAT entschieden, da die annotierten Textdaten so direkt als Trainingsdaten für automatisierte Textanalyseverfahren im Rahmen der interdisziplinären Kooperation eingesetzt werden konnten. In einer mehrtägigen Schulung wurden die theoretischen Konstrukte diskutiert und das zuvor entwickelte Codebuch anhand mehrerer Probeannotationen gemeinsam fortentwickelt. Die Intercoder-Reliabilität, die anhand von 143 Nutzerkommentaren getestet wurde, lag bei den meisten Variablen bei einem Krippendorff's α zwischen .80 und .90. Die Variable Respekt fiel mit .66 am schlechtesten aus.

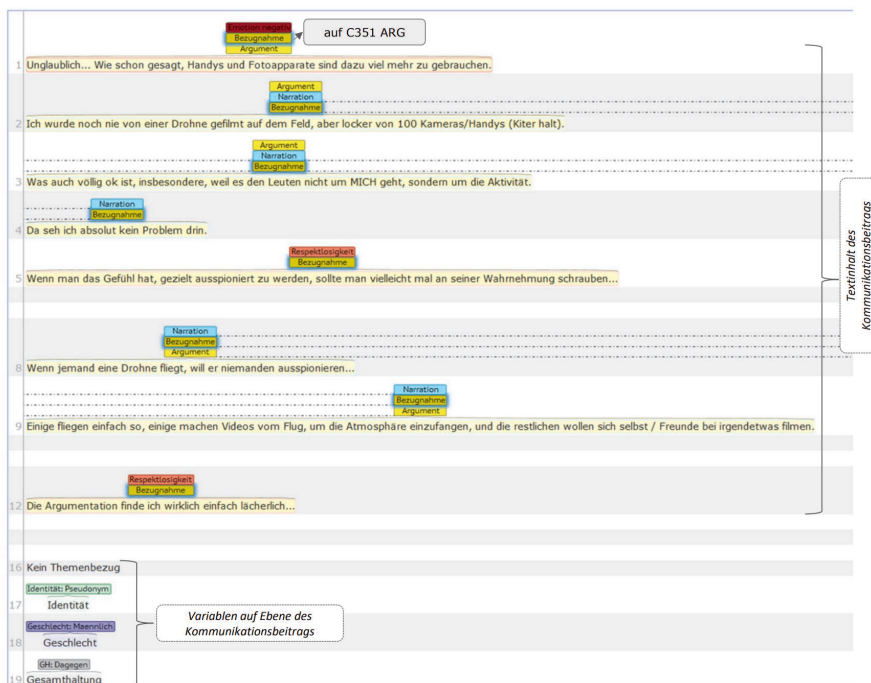
Die Annotation des Untersuchungsmaterials wurde im August und September 2017 durchgeführt. Zuerst wurde der gesamte Diskussionsstrang gelesen, erst dann wurden alle Kommentare in der Reihenfolge ihres Erscheinens in der Baumstruktur des Threads annotiert. Gearbeitet wurde dabei mit zwei parallel geöffneten Browserfenstern, sodass der zu annotierende Nutzerkommentar in BRAT und der gesamte Diskussionsstrang in einer HTML-Ansicht zeitgleich eingesehen werden konnten.

In einem ersten Schritt wurden für jeden Nutzerkommentar Variablen auf Kommentarebene annotiert (z. B. Gesamthaltung des Kommentars, Geschlecht im Nutzernamen). Um dazu allerdings nicht gleich den gesamten Text markieren zu müssen, wurden die entsprechenden Variablen unter jeder Textdatei eingefügt, sodass die Annotierenden sie einfach nur markieren mussten, um die entsprechenden Ausprägungen als Entitäten zu annotieren (siehe Abb. 2, Zeilen 16–19). Anschließend wurden dann auf Äußerungsebene die Kommunikationsformen (Argumentation, Narration, Emotionsäußerung und Humor) sowie Merkmale deliberativer und nicht-deliberativer Kommunikation (Konstruktivität, Reflexivität, Empathie und Respekt) ebenfalls als Entitäten annotiert. Für eine bessere Übersicht wurden die annotierten Textstellen in

einer zur Kategorie passenden Farbe angezeigt und mit dem entsprechenden Label versehen (Argumente gelb, negative Emotionen rot usw.; siehe Abb. 2, Zeile 1). Überlappende Annotationen waren grundsätzlich zulässig, da eine Äußerung eine Kombination aus verschiedenen Kommunikationsformen (z. B. Narration und Argumentation) sein konnte.

Die Relationen zwischen einzelnen Aussagen in den Nutzerkommentaren wurden in BRAT unter Benutzung des Kommentarfeldes annotiert (siehe Abb. 2, Zeile 1). Das Kommentarfeld erwies sich dafür als geeignete Lösung, da ein Relationieren von Entitäten über verschiedene Kommentare hinweg mit der „relation annotation“-Funktion in BRAT nicht möglich ist. Stattdessen wurden im Kommentarfeld für jede annotierte Aussage, die sich auf eine Aussage in einem vorangegangenen Kommentar bezieht, zuvor festgelegte Codes eingetragen. Ein Beispiel: In dem Nutzerkommentar C352 bezieht sich eine Aussage, die als negative Emotion (EMON) und Argument (ARG) annotiert wurde, auf eine als Argument annotierte Aussage in dem vorangegangenen Kommentar C351 (siehe Abb. 2, Zeile 1). Für die Aussage in C352 wurde deshalb zusätzlich noch eine Bezugnahme annotiert und in dem Kommentarfeld zu dieser Annotation „auf C351 ARG“ notiert. Durch dieses Vorgehen lagen nach der Annotation Daten auf drei verschiedenen Ebenen vor: auf Kommentar-, Aussagen- und Relations-ebene. Sie wurden als Excel-Dateien für spätere Regressions- und Sequenzanalysen aus BRAT extrahiert und gespeichert.

Abbildung 2: Annotation der Kommunikationsformen und Antwort-Relationen in einem beispielhaften Nutzer-Kommentar und in BRAT



3.2 Studie 2: Hate Speech in Nutzerkommentaren zu Flucht und Migration (2018–2020)

Die zweite Studie wurde im Rahmen des interdisziplinären Forschungsprojekts NO-HATE durchgeführt, in dem Hate Speech in deutschsprachigen Nutzerkommentaren zum Thema Flucht und Migration untersucht wurde.⁸ Ziel des Projekts war es, die Prävalenz und inhaltliche Dimensionierung von Hasskommentaren in partizipativen Medienangeboten im Internet zu analysieren und eine Schnittstelle zur automatisierten Hate-Speech-Erkennung zu entwickeln. In einer manuellen Inhaltsanalyse wurden dafür drei Teilstichproben von Nutzerkommentaren aus den Kommentarspalten ausgewählter journalistischer Webseiten und diverser Social-Media-Angebote annotiert (siehe Paasch-Colberg et al., 2021a, 2021b). Die Annotationen wurden dann zum einen statistisch ausgewertet (Paasch-Colberg et al., 2021c, 2022) und dienten zum anderen als Trainingsdatensatz für die Implementierung und Evaluation eines Machine-Learning-Modells zur automatischen Hate-Speech-Erkennung.

Im Zuge der Kommentaranalyse wurde Hate Speech indirekt mit Hilfe verschiedener theoretisch abgeleiteter Indikatoren auf der Ebene von Bewertungen erfasst. Auf der Basis eines Codebuches wurden dafür zunächst Bewertungen in den zu untersuchenden Kommentaren identifiziert, als Entität „Bewertung“ annotiert und dann jeweils mit einer Reihe von zuvor festgelegten Attributen (z. B. Bewertungsrichtung, Bewertungsobjekt, Gruppenbezüge, Entmenschlichung) weiter qualifiziert (siehe Abb. 3; für Details siehe Paasch-Colberg et al. 2021b). Sofern alle in der Konfiguration als obligatorisch definierten Auswahlentscheidungen getroffen wurden, erschien die Textstelle in der Benutzer*innen-Oberfläche als grün markiert und mit einem Kurz-Label versehen, das den Entitäten-Typ und die Anzahl der annotierten Attribute anzeigt. Um den Annotierenden die Möglichkeit zu geben, offene Anmerkungen zu ihren Annotationen zu ergänzen, wurde eine zweite Entität „Anmerkung“ als offenes Textfeld konfiguriert.

Ursprünglich war geplant, weitere Konstrukte in BRAT zu annotieren und dafür auch komplexere Annotationsformen zu nutzen. So war angedacht, Sprecher*innen als „Akteur“-Entität zu annotieren und über die Relationierungsfunktion mit Bewertungen in Bezug zu setzen. Zudem war geplant, Frames in Nachrichtenartikeln und Social-Media-Postings als Entität „Ereignis“ zu annotieren. Auf diese Weise wäre es zum Beispiel möglich gewesen, einer Problemdefinition Sprecher*innen und als verantwortlich identifizierte Akteure zuzuweisen (siehe Abb. 4). Da sich in den Probenläufen jedoch herausstellte, dass solche komplexen Annotationen sehr zeitaufwendig sind und die Ressourcen des Projekts vor allem in die Annotation von Hate Speech fließen mussten, wurden letztendlich keine Sprecher*innen identifiziert und Problemdefinitionen ab der zweiten Teilstichprobe nicht mehr annotiert.

8 In dem BMBF-geförderten Verbundprojekt „NOHATE – Bewältigung von Krisen öffentlicher Kommunikation im Themenfeld Flüchtlinge, Migration, Ausländer“ haben Informatiker*innen der Berliner Hochschule für Technik, Kommunikationswissenschaftler*innen der Freien Universität Berlin und Computerlinguist*innen von VICO Research & Consulting GmbH zusammengearbeitet (Förderkennzeichen: 01UG1735AX).

<https://doi.org/10.5771/1615-634X-2022-4-446>, am 17.05.2024, 08:48:29

Abbildung 3: Annotation einer negativen Bewertung im NOHATE-Projekt

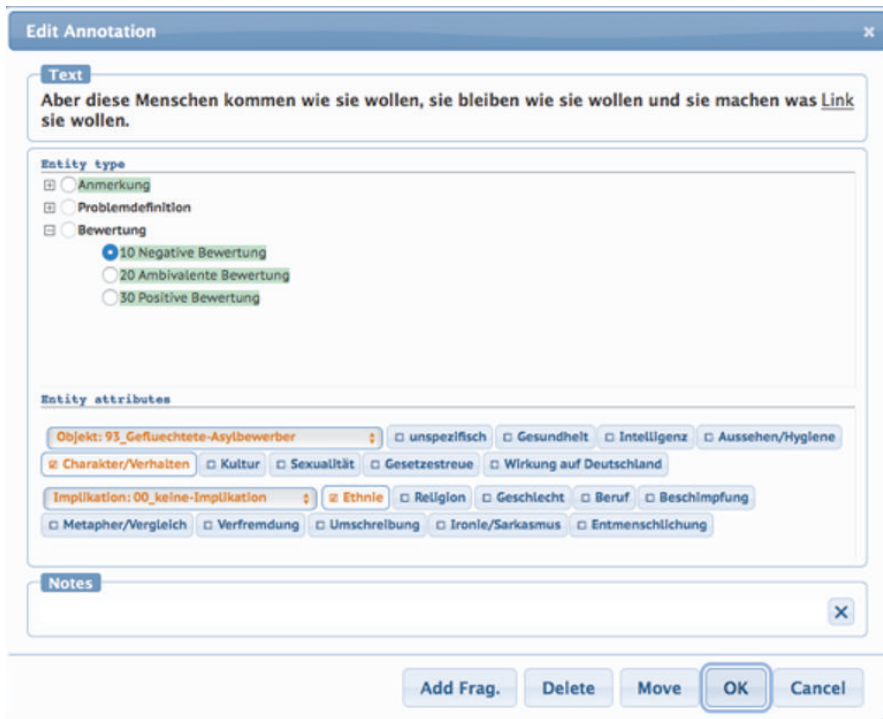
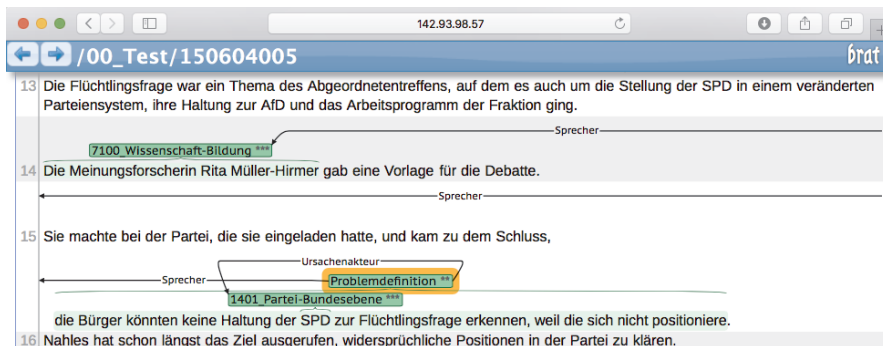


Abbildung 4: Annotation von Problemdefinitionen in BRAT als „Ereignis“



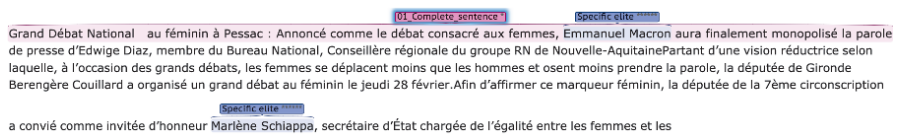
3.3 Studie 3: Anti-Elitismus in der digitalen Kommunikation rechtspopulistischer und rechtsradikaler Parteien im Wahlkontext (2020–2021)

Schließlich kam BRAT in einer dritten Studie zum Einsatz, in der die Typen und Charakteristika verschiedener Formen von Anti-Elitismus in der digitalen Kommunikation rechtspopulistischer und rechtsradikaler Parteien aus sechs europäischen Ländern im Kontext der Wahlen zum Europaparlament 2019 untersucht wurden (Vaughan & Heft, 2022). Gegenstand der strukturierten Annotation mit BRAT waren Facebook-Postings und Beiträge auf den Webseiten der Parteien, die in sechs Sprachen vorlagen. Die Studie differenziert konzeptionell zwischen zwei Typen von Anti-Elitismus: spezifischem Anti-Elitismus, der sich gegen konkrete, auf verschiedener Grundlage machtvolle Eliten richtet („Angela Merkel“), und generellem Anti-Elitismus, der sich auf breitere diskursive Konstrukte bezieht („die Elite“, „die herrschende Klasse“).

Auf Basis eines standardisierten Codebuchs wurden spezifische und generelle Referenzen auf Eliten annotiert (siehe Abb. 5). Hier erwies sich die Nutzung von BRAT bereits für die Tests und Weiterentwicklung des Codebuchs als hilfreich. Ausgangspunkt waren in der Literatur gut dokumentierte Konstrukte, beispielsweise ein Verständnis von Eliten als Gruppen oder Einzelpersonen mit regelmäßigem und erheblichem Einfluss auf wichtige Entscheidungen innerhalb einer Organisation oder einer Gesellschaft (Hoffman-Lange, 2018). Die Tests mit verschiedenen Annotierenden und die Diskussion der Annotationen direkt am markierten Text ermöglichte es auf effiziente Weise, das Codebuch mit detaillierten Beispielen und spezifischen Regeln anzureichern, die Handreichungen für die Entscheidung von Zweifelsfällen bereitstellten.

Das Codebuch enthielt darüber hinaus weitere Variablen, die mit jeder Referenz auf eine spezifische oder allgemeine Eliten-Nennung verbunden waren. Der „Elitetyp“ erfasste, ob die Elite eine Einzelperson oder eine Organisation war. „Sektor“ erfasste die Zugehörigkeit des Akteurs zu verschiedenen gesellschaftlichen Sektoren wie Politik, Medien, Wirtschaft, Recht und Kultur/Wissenschaft. „Parteizugehörigkeit“ codierte die spezifische Parteizugehörigkeit aller politischen Akteure. Schließlich erfasste „Scope“ die geografische Reichweite der Eliten und kontrastierte nationale Eliten im Heimatland des Sprechers mit nationalen Eliten in anderen Ländern sowie Eliten, die auf transnationaler Ebene (wie der EU) aktiv sind. In Bezug auf die Bewertung der referenzierten Elite wurden zwei Schlüsselvariablen codiert. Die Variable „Sentiment“ erfasste, ob die Referenz positiv, negativ oder neutral war, basierend auf der Einbettung einer Referenz im jeweiligen Kontext eines Satzes. Schließlich wurden verschiedene Arten von Anti-Elitismus annotiert, wofür sich die Studie an der Konzeptionalisierung und dem Codebuch von Ernst et al. (2017) orientierte, die zwischen discrediting, blaming, und detaching differenzieren.

Abbildung 5: Annotation von Elite-Referenzen in französischem Text



Mit Bezug auf generellen Anti-Elitismus wurde BRAT in dieser Studie in einer zweiten Weise genutzt. Es zeigte sich, dass genereller Anti-Elitismus im unstrukturierten

Text des Untersuchungsmaterials der Studie viel seltener vorkam, als dass die manuelle Annotation ressourcenschonend hätte eingesetzt werden können. Daher wurde die BRAT-Annotation in einer Stichprobe der Daten genutzt, um ein Diktionär mit potenziell generellen Anti-Elitismus repräsentierenden Termen zu erstellen. Die Diktionärsentwicklung erfolgte in mehreren Schritten, ausgehend von Annotationen im Material für Schulungs- und Pretest-Codierungen. Die Annotationen wurden sämtlich manuell geprüft, sich herauskristallisierende „Schlüsselbegriffe“ in alle Sprachen übersetzt und das resultierende Diktionär auf den gesamten Korpus der Studie angewandt. Auf diese Weise konnten *potenzielle* Referenzen auf allgemeinen Anti-Elitismus im Gesamtkorpus leichter identifiziert und für eine anschließende manuelle Codierung aufbereitet werden (für die Methodendokumentation siehe Vaughan & Heft, 2022).

4. Bewertung

Legen wir zur Bewertung von BRAT zunächst die von Haim (2021) formulierten Gütekriterien für die Entwicklung von Forschungssoftware entlang der Dimensionen Transparenz, Replizierbarkeit, Validierbarkeit und Nachhaltigkeit zugrunde, ergibt sich insgesamt ein positives Bild: Als Open-Source-Software kann BRAT einfach und kostenlos von der Webseite der Entwickler*innen heruntergeladen und mit wenig Vorwissen selbstständig installiert werden, es ist durch viel hilfreiches Material (Einführung, Anwendungsbeispiele, Manual etc.) sehr gut dokumentiert und hat sich in zahlreichen Forschungsprojekten bereits bewährt. Der Code der Software baut zudem auf der Programmiersprache Python auf, ist frei lizenziert auf GitHub verfügbar und kann so durch andere Wissenschaftler*innen gepflegt und weiterentwickelt werden. Aus Perspektive der Softwareentwicklung erfüllt BRAT damit die meisten der von Haim genannten Gütekriterien.

Im Zentrum unserer Bewertung soll im Folgenden allerdings die Nutzung des Tools stehen, insbesondere seine Potenziale für die Inhaltsanalyse sowie bemerkenswerte Schwierigkeiten und Herausforderungen im Umgang mit der Software.

4.1 Vorteile und Potenziale

Der wohl wichtigste Vorteil von BRAT gegenüber anderen (softwaregestützten) Varianten der Inhaltsanalyse ist, dass es die Datenerhebung forschungspraktisch erheblich vereinfacht, ohne dass dafür nennenswerte Kosten anfallen. Verglichen mit der eingangs erwähnten Codierung von Texten mit Hilfe von Microsoft Excel, bei der die Codierer*innen mit ihrem Blick immer wieder zwischen Codebuch, Untersuchungsmaterial und Eingabemaske hin- und herspringen (müssen), integriert BRAT zumindest die letzten zwei Bestandteile übersichtlich in einem Bildschirmfenster (in diesem Punkt besser: *angrist*, siehe Wettstein, 2015). Die jeweiligen Codes müssen nicht mehr auswendig gelernt und händisch notiert werden, sondern werden durch die Auswahl und Vergabe der Labels im Zuge der Annotation automatisch zugewiesen, wodurch das Annotieren sehr viel weniger fehleranfällig wird. Das schlichte Design und die Einfachheit der visuellen Benutzeroberfläche machen die Nutzung von BRAT darüber hinaus übersichtlich, intuitiv verständlich und unkompliziert.

Damit verbunden ist ein zweiter wichtiger Vorteil, der insbesondere die Schulung der Annotierenden betrifft: Durch den Einsatz von BRAT können Unsicherheiten mit Annotationsentscheidungen und Fragen zum Codebuch direkt an Beispielen aus dem Untersuchungsmaterial diskutiert werden. Unterstützt wird dies vor allem dadurch, dass jede Annotation durch die automatische Vergabe von Uniform Resource

Identifiers verlinkt und mit anderen Schulungsteilnehmenden geteilt werden kann. Zudem erlaubt die Funktion der „side-by-side visualization“, zwei Annotationen auf einem Bildschirm gegenüberzustellen und direkt miteinander zu vergleichen. Auch wenn sich die Annotierenden zu Beginn der Schulung mit der Bedienung der neuen Software vertraut machen müssen, so, scheint uns, fördern der visuelle Zugang und die hohe Nutzerfreundlichkeit des Tools den Lernfortschritt weitaus besser als der iterative Abgleich von in Tabellen eingetragenen Zahlencodes.

Ein dritter Vorteil betrifft schließlich die Weiterverarbeitung und Analyse der annotierten Daten: So erlaubt die Kombination aus induktivem Textlabeling und standardisierter Code-Zuweisung in BRAT eine sowohl quantitative als auch qualitative Datenauswertung. In der ersten Studie wurde dies etwa für die Kombination aus Regressions- und Sequenzanalyse genutzt; in der zweiten Studie für eine qualitative Typenbildung einerseits (Paasch-Colberg et al., 2021a) und eine quantitative Auswertung andererseits (Paasch-Colberg et al., 2022). Darüber hinaus eignet sich das Tool weitaus besser als Microsoft Excel oder MaxQDA für eine interdisziplinäre Zusammenarbeit mit der Informatik und/oder der Computerlinguistik – etwa zur Generierung von Diktationären oder dem Trainieren algorithmischer Modelle im Bereich der automatisierten Textanalyse. Nicht nur sind Annotationstools wie BRAT in diesen Disziplinen weit verbreitet und bekannt, die kostenlose Nutzbarkeit und das einfache, flexible Datenformat bergen darüber hinaus große Potenziale für die computationale Weiterverarbeitung. Die Nachnutzung erhobener Daten etwa zur Replikation der Datenerhebung oder zur Weiterentwicklung automatisierter Verfahren werden so auch über disziplinäre Grenzen hinweg vereinfacht.

4.2 Schwierigkeiten und Herausforderungen

Den Vorteilen der BRAT-Nutzung stehen Herausforderungen gegenüber, die vor einem Einsatz der Software zur Unterstützung inhaltsanalytischer Forschungsprojekte berücksichtigt werden sollten. Bemerkenswert ist dabei, dass einige dieser Herausforderungen direkt mit den Vorteilen der Software selbst verbunden sind, das eine also nicht ganz ohne das andere zu haben ist.

Deutlich wird dies etwa bei den Schulungen: So eignet sich BRAT aufgrund seiner visuellen Oberfläche und der einfachen Bedienbarkeit zwar sehr gut für die Textannotation, doch ist diese Methode der Datenerhebung selbst recht anspruchsvoll und voraussetzungsreich. Nicht nur muss im Vorfeld das Codebuch in Aufbau und Struktur an der Verfahrenslogik der Annotation ausgerichtet werden (die oben vorgestellten Projekte liefern dafür erste Beispiele, siehe Paasch-Colberg et al., 2021b sowie den Anhang zu Vaughan & Heft, 2022), vor allem ist es eine große Herausforderung, die Annotierenden sowohl im reliablen Zuweisen von Labels zu bestimmten Textstellen als auch in der reliablen Identifikation eben dieser Textstellen zu schulen. Da die Untersuchungseinheiten bei annotationsbasierten Inhaltsanalysen in der Regel nicht auf der Ebene des gesamten Textdokuments (einem Beitrag, einem Post etc.) liegen, sondern auf der Ebene einzelner Aussagen, Sätze oder Wörter, ist die Schulung entsprechend aufwendig. Dies gilt insbesondere bei der recht komplexen Annotation von Ereignissen (siehe oben).

Auch wenn diese Schwierigkeiten auf die Methode der Textannotation und nicht auf BRAT selbst zurückzuführen sind, sollten sie bei der Abwägung eines möglichen Einsatzes dieses Tools zur Unterstützung einer Inhaltsanalyse eine entscheidende Bedeutung spielen. Ausschlaggebend ist dabei die Frage nach der semantischen Komplexität und Kontextabhängigkeit jener Konstrukte, die von den Annotierenden in

den Texten als Entitäten identifiziert werden müssen. So haben die hier vorgestellten Beispielstudien gezeigt, dass Akteure, Argumente, Emotionen, Bewertungen und Referenzen auf Eliten gut und durchaus reliabel identifiziert und weiter qualifiziert werden können, während etwa Frames und die ihnen zugrunde liegenden Problemdefinitionen sowie andere relationale Konstrukte, für die verschiedene Aspekte im Text einheitlich identifiziert und miteinander in Beziehung gesetzt werden müssen, schon sehr viel schwieriger zu annotieren sind und entsprechend zusätzliche Ressourcen dafür eingeplant werden müssen.

Relevant ist in diesem Zusammenhang auch die Frage, ob die Identifikation dieser Konstrukte im Nachgang der Annotation automatisiert werden soll. Da für eine solche Automatisierung inhaltsanalytischer Verfahren sehr große Mengen zuvor klassifizierter Texte benötigt werden, muss der hohe Aufwand einer reliablen Annotation im Vorfeld sorgsam abgewogen werden. Gegenüber der einfachen Codierung von Konstrukten auf der Ebene des gesamten Textes bringt die Annotation allerdings den großen Vorteil mit sich, diejenigen Passagen eines Textes, in denen das Konstrukt identifiziert wird, sehr viel genauer bestimmen zu können. Für das Trainieren algorithmischer Detektionsverfahren ist dies nicht unerheblich, da durch eine genaue Annotation der relevanten Textstellen vermieden werden kann, dass die für eine Automatisierung nicht relevanten Textstellen mit in die Mustererkennung einbezogen werden.

Von diesen methodischen Herausforderungen abgesehen, bringt der Einsatz von BRAT spezifische Schwierigkeiten mit sich, die mit dem oben als Vorteil beschriebenen Umstand verbunden sind, dass es sich bei BRAT um ein Open-Source-Tool handelt. Denn ohne eine kommerzielle Verwertung der Software fehlen die Ressourcen, um sie dauerhaft in Stand zu halten und weiterzuentwickeln. Zwar reagieren die Entwickler*innen und einige Nutzer*innen auf die im „issues“-Forum auf Github⁹ eingestellten Fragen und Probleme, doch gibt es keinen technischen Support, der zeitnah reagiert. Auch wenn es in allen drei Anwendungsbeispielen keine größeren Probleme gab, die Software also insgesamt ausgereift ist, kann dies für Forschende, die technisch nicht so versiert sind und auch auf keine IT-Unterstützung zurückgreifen können, zumindest bei der Erstanwendung eine Hürde sein. Dies gilt – trotz der sehr guten Dokumentation und Anleitung auf der Webseite von BRAT – bereits für die Installation und Konfiguration der Software, die immerhin voraussetzungsreicher ist als die Nutzung häufig bereits installierter Software wie Microsoft Excel; vor allem aber wird dies relevant, wenn das gesamte Spektrum der oben gelisteten Funktionalitäten ausgenutzt werden soll.

5. Fazit

Ist BRAT nun also eine ernst zu nehmende Alternative für (computergestützte) Inhaltsanalysen in der Kommunikations- und Medienforschung? Mit Blick auf die zahlreichen Funktionalitäten des Tools, die insbesondere für die Annotationsschulungen viele Vorteile mit sich bringen, seine freie Verfügbarkeit, die nutzungsfreundliche Bedienung und die interdisziplinäre Anschlussfähigkeit insbesondere in der Informatik und (Computer-)Linguistik können wir diese Frage eindeutig bejahen. Angesichts der methodischen Herausforderungen der annotationsbasierten Inhaltsanalyse scheint uns ein Einsatz allerdings nur dann empfehlenswert, wenn es gilt, Konstrukte auf Aussage-, Satz- oder Wortebene zu klassifizieren und ggf. zu relationieren – etwa

⁹ <https://github.com/nlplab/brat/issues>.

bei Mehrebenenanalysen, der Kombination von qualitativer und quantitativer Inhaltsanalyse oder eben bei der Entwicklung von Machine-Learning-Verfahren zur automatisierten Klassifikation dieser Konstrukte in Texten. In diesen Fällen lohnt es sich aus unserer Sicht, den anfänglichen Aufwand bei der Implementierung des Tools auf sich zu nehmen. Für die Klassifikation ganzer Texte, wie sie in kommunikations- und medienwissenschaftlichen Inhaltsanalysen üblich sind, ist das Tool hingegen weniger geeignet als der klassische Rückgriff auf Excel.

Über die Frage nach sinnvollen Einsatzmöglichkeiten von BRAT hinaus stellt sich aus unserer Sicht allerdings noch eine weitere Frage: danach nämlich, wie sich die Bedingungen für ihren Einsatz verbessern lassen. Hier sehen wir zwei mögliche Ansatzpunkte: Zum einen wäre es wünschenswert, die annotationsbasierte Inhaltsanalyse mit Tools wie BRAT im Methodenkanon des Fachs stärker zu berücksichtigen. Dies wäre nicht nur angesichts der Zunahme interdisziplinärer Kooperationen mit der Informatik wichtig, sondern auch, um die Potenziale dieser Methode sowie entsprechender Tools zur Bearbeitung inhaltsanalytischer Fragestellungen besser auszuschöpfen. Zum anderen muss der vielerorts geforderte Wechsel von kommerzieller Forschungssoftware hin zu kostenlosen Open-Source-Tools wie BRAT nachhaltig gestaltet und institutionell abgesichert werden (siehe dazu Hepp et al., 2021a). Dies gilt nicht nur für die Pflege und Weiterentwicklung von Software, sondern auch für die Beratung und Unterstützung ihrer Nutzung. Forschungsk Kooperationen mit der Informatik bieten dafür eine gute Gelegenheit: Bei Anträgen entsprechender Forschungsprojekte, in denen Open-Source-Tools wie BRAT eingesetzt werden sollen, ließen sich etwa Mittel für deren Weiterentwicklung mit einplanen. Zudem sollten Forschungs- und Infrastruktureinrichtungen prüfen, ob sie die Nutzung von Tools wie BRAT nicht durch freie Zugänge zu eigenen Instanzen, die Bereitstellung entsprechender Serverkapazitäten sowie eine niedrigschwellige Beratung oder zumindest die Koordinierung von Expertise im Umgang mit solcher Software unterstützen könnten. Wir hoffen, mit der vorliegenden Besprechung einen (weiteren) Anstoß dafür geben zu können.

Quellenverzeichnis

- Biemann, C., Bontcheva, K., Eckart de Castilho, R., Gurevych, I., & Yimam, S. M. (2017). Collaborative web-based tools for multi-layer text annotation. In N. Ide & J. Pustejovsky (Hrsg.), *Handbook of Linguistic Annotation* (S. 229–256). Springer.
- Eckart de Castilho, R., Mújdricza-Maydt, É., Yimam, S. M., Hartmann, S., Gurevych, I., Frank, A., & Biemann, C. (2016). A web-based tool for the integrated annotation of semantic and syntactic structures. In *Proceedings of the LT4DH workshop at the 25th International Conference on Computational Linguistics (COLING 2016)*, Osaka, Japan. <https://aclanthology.org/W16-4011> [18.10.2022].
- Ernst, N., Engesser, S., & Esser, F. (2017). Bipolar populism? The use of anti-elitism and people-centrism by Swiss parties on social media. *Swiss Political Science Review*, 23(3), 253–261. <https://doi.org/10.1111/spsr.12264>.
- Esau, K. (2018). Capturing citizens' values: On the role of narratives and emotions in digital participation. *Analyse & Kritik*, 40(1), 55–72. <https://doi.org/10.1515/auk-2018-0003>.
- Esau, K. (2022). *Kommunikationsformen und Deliberationsdynamik. Eine relationale Inhalts- und Sequenzanalyse von Online-Diskussionen auf Beteiligungsplattformen*. Nomos.
- Esau, K., & Friess, D. (2022). What creates listening online? Exploring reciprocity in online political discussions with relational content analysis. *Journal of Deliberative Democracy*, 18(1), 1–16. <https://doi.org/10.16997/jdd.1021>.
- Escher, T., Frieß, D., Esau, K., Sieweke, J., Tranow, U., Dischner, S., Hagemeyer, P., Mauve, M. (2016). Online deliberation in academia: Evaluating the quality and legitimacy of cooperative-

- ly developed university regulations. *Policy & Internet*, 9(1), 133–164. <https://doi.org/10.1002/poi3.119>.
- Feiks, M. (2016). *Datenerhebung mit Excel: Eine Anleitung zur Umsetzung von Inhaltsanalysen und Befragungen*. Springer.
- Haim, M. (2021). Gütekriterien und Handlungsempfehlungen für die Entwicklung von Forschungssoftware in der Kommunikations- und Medienwissenschaft. *Medien & Kommunikationswissenschaft*, 69(1), 65–79. <https://doi.org/10.5771/1615-634X-2021-1-65>.
- Hepp, A., Hohmann, F., Belli, A., Boczek, K., Haim, M., ..., & Unkel, J. (2021a). *Forschungssoftware in der Kommunikations- und Medienwissenschaft: Stand, Herausforderungen und Perspektiven*. https://www.dgpuk.de/sites/default/files/DGPuK%20Positionspapier%20-%20Forschungssoftware%20in%20der%20Kommunikations-%20und%20Medienwissenschaft_0.pdf.
- Hepp, A., Loosen, W., & Hasebrink, U. (2021b). Jenseits des Computational Turn: Methodentwicklung und Forschungssoftware in der Kommunikations- und Medienwissenschaft. *Medien & Kommunikationswissenschaft*, 69(1), 4–24. <https://doi.org/10.5771/1615-634X-2021-1-3-1>.
- Hoffmann-Lange, U. (2018). Methods of elite identification. In H. Best, J. Higley, M. Cotta et al. (Hrsg.), *The Palgrave Handbook of Political Elites* (S. 79–92). Palgrave Macmillan.
- Klie, J.-C., Bugert, M., Boulosa, B., Eckart de Castilho, R., & Gurevych, I. (2018). The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of System Demonstrations of the 27th International Conference on Computational Linguistics (COLING 2018)*, Santa Fe, New Mexico, USA. <https://aclanthology.org/C18-2002> [18.10.2022].
- Liebeck, M., Esau, K., Conrad, S. (2016). What to do with an airport? Mining arguments in the German online participation project Tempelhofer Feld. *Proceedings of the 3rd Workshop on Argument Mining*, ACL 2016, Berlin, 7.–12. August, 144–153. <https://doi.org/10.18653/v1/W16-2817>.
- Liebeck, M., Esau, K., Conrad, S. (2017). Text Mining für Online-Partizipationsverfahren: Die Notwendigkeit einer maschinell unterstützten Auswertung. *HMD Praxis der Wirtschaftsinformatik, Online Participation*, 54(4), 544–562. <https://doi.org/10.1365/s40702-017-0321-6>.
- Neves, M., & Ševa, J. (2021). An extensive review of tools for manual annotation of documents. *Briefings in Bioinformatics*, 22(1), 146–163. <https://doi.org/10.1093/bib/bbz130>.
- Paasch-Colberg, S., Strippel, C., Trebbe, J., & Emmer, M. (2021a). From insult to hate speech: Mapping offensive language in German user comments on immigration. *Media and Communication*, 9(1), 171–180. <https://doi.org/10.17645/mac.v9i1.3399>.
- Paasch-Colberg, S., Strippel, C., Laugwitz, L., Emmer, M., & Trebbe, J. (2021b). *Codebuch: Hate Speech im Internet 3/2019. Annotationsbasierte Inhaltsanalyse von Nutzerkommentaren zum Thema Flucht und Migration aus März 2019*. <https://doi.org/10.31235/osf.io/c59kn>.
- Paasch-Colberg, S., Strippel, C., Laugwitz, L., Emmer, M., & Trebbe, J. (2021c). *Field Report: Hate Speech im Internet 3/2019. Annotationsbasierte Inhaltsanalyse von Nutzerkommentaren zum Thema Flucht und Migration aus März 2019*. [Methodenbericht]. Freie Universität Berlin. <https://doi.org/10.31235/osf.io/7n45g>.
- Paasch-Colberg, S., Trebbe, J., Strippel, C., & Emmer, M. (2022). Insults, criminalization, and calls for violence: Forms of hate speech and offensive language in German user comments on immigration. In A. Monnier, A. Boursier, & A. Seoane (Eds.), *Cyberhate in the context of migrations* (pp. 137–163). Palgrave Macmillan.
- Perry, T. (2021). LightTag: Text annotation platform. <https://arxiv.org/abs/2109.02320>.
- Stenetorp, P., Pyysalo, S., Topic, G., Ohta, T., Ananiadou, S., & Tsujii, J. I. (2012). BRAT: A web-based tool for NLP-assisted text annotation. *Proceedings of the 13th Conference of the European Chapter of the ACL*, Avignon, France. <https://aclanthology.org/E12-2021> [18.10.2022].
- Vaughan, M., & Heft, A. (2022). Anti-elitism in the European radical right in comparative perspective. *JCMS: Journal of Common Market Studies*, Advanced Online Publication, 1–19. <https://doi.org/10.1111/jcms.13347>.

Wettstein, M. (2015). *angrist 1.2.1(en): Documentation and reference for the coder interface*.
https://github.com/Tarlanc/angrist/blob/master/ANGRIST_1-21-en.docx.



© Christian Strippel / Laura Laugwitz / Sünje Paasch-Colberg / Katharina Esau /
Annett Heft