



ARTIKEL

2.2. Algorithmenkritik

von [Althage, Melanie](#); [Deicke, Aline](#); [Hall, Mark](#); [Möbus, Dennis](#); [Petz, Cindarella](#); [Seltmann, Melanie](#);

Version 1.2 | Veröffentlicht 29. Juli 2024

1. Einleitung

Mit der Digitalisierung verändern sich nicht nur die Forschungsgegenstände klassisch hermeneutisch arbeitender Disziplinen, sondern mithin die analytischen und interpretativen Methoden. Die Verfügbarkeit von Quellen ist in den letzten Jahren durch eine umfassende Retrodigitalisierung und entsprechende Bereitstellung über webbasierte Datenbanken erheblich gestiegen. Durch *born-digital data* wird die Quellenbasis für Geistes- und Kulturwissenschaften in den nächsten Jahren weiter dramatisch anwachsen und verlangt nach Methoden, um diese zu durchmessen und zu analysieren. Die Digital Humanities leisten in dieser Hinsicht seit einigen Jahren Pionierarbeit, um den Methodenkanon der Geistes- und Kulturwissenschaften zu erweitern. Mit Einzug dieser computerbasierten Analyse- und Interpretationsverfahren wurde die geistes- und kulturwissenschaftliche Methodologie um Algorithmen erweitert. Im Unterschied zu den klassischen, in Aufsätzen oder Büchern ausformulierten Methoden der Geistes- und Kulturwissenschaften, muss der Algorithmus oder die diesen begleitenden Erklärungen (Dokumentation) nicht zwangsläufig gelesen werden, um ihn anzuwenden. In der Regel liegen Algorithmen als ausführbare Programme oder Codeimplementierungen in Form von Programmiersprachenerweiterungen (Paketen, Modulen, Bibliotheken) oder fertigen Pipelines vor, die zumeist auf Codeebene bedient oder über eine grafische Schnittstelle gesteuert werden müssen. Der Algorithmus selbst ist dabei nicht als literarischer Text zu verstehen, sondern als “an exact prescription, defining a computational process, leading from various initial data to the desired results” (Markov 1954) oder konziser: eine “effective procedure” (Minsky 1967).

Genügt zur Analyse und Kritik von Algorithmen also nur der Nachvollzug der Regeln? Diese Verkürzung würde der Analyse von Algorithmen nicht gerecht werden. Zudem schwingt in den Definitionen nicht selten ein Objektivitätspostulat mit, das unhaltbar ist, wie Phänomene wie *Biases* in *Machine-Learning*-Anwendungen zeigen (vgl. Mehrabi u. a. 2021; Noble 2018; Broussard 2018). Schon Markov hat 1954 auf die zentrale Verschränkung von Daten und Algorithmen – und damit von Welt und Algorithmen – hingewiesen (Markov 1954). Das macht Algorithmen zu einem wichtigen Gegenstand hermeneutischer Untersuchung, um den ihnen eingeschriebenen Blick auf die Daten und die Welt verstehen zu lernen. Eine derart erweiterte Methodenkritik sollte als Algorithmenkritik in den geistes- und kulturwissenschaftlichen Werkzeugkasten aufgenommen werden.

Der hier gewählte praktische Zugang beschränkt sich weder auf die rein logisch-mathematische noch auf die code-immanente Algorithmenkritik. Denn viele Algorithmen sind für Nicht-Informatiker:innen nicht oder nur schwer “lesbar”; das liegt zum einen daran, dass viele Algorithmen lediglich cursorisch dokumentiert sind, zum anderen, dass diese Dokumentationen durch ihren hohen Grad an Abstraktion und Formalisierung für Laien schwer nachvollziehbar sind und Spezialkenntnisse erfordern. Auch der bei der Implementierung von Algorithmen entstandene Programmcode ist zunächst eine Art Fremdsprache, die erlernt werden muss.^[1] Darüber hinaus kann Programmcode äußerst komplex und von beträchtlichem Umfang sein. Deshalb bedient sich der hier gewählte Zugang eines iterativen Vorgehens, um systematisch von den Ergebnissen aus in die Logik des Algorithmus einzusteigen. Nach einem kurzen Abriss über die Geschichte des Begriffs “Algorithmus” und damit einhergehenden Definitionen, werden drei Säulen der Algorithmenkritik herausgestellt: Forschungsdesign, Funktionalität und gesellschaftlich-lebensweltliche Auswirkungen. In drei folgenden Praxisbeispielen werden entlang dieser Säulen Suchmaschinen, Inferenzalgorithmen für das Topic Modeling und Algorithmen zur Berechnung von Zentralitätsmetriken aus der Netzwerktheorie untersucht. Auf diese Weise können auch Geistes- und Kulturwissenschaftler:innen im Sinne einer spezifischen *Algorithm Literacy* oder einer allgemeinen *Digital Literacy* die Methoden der Algorithmenkritik erlernen. Der hier gewählte Ansatz der Algorithmenkritik begegnet auch dem Desiderat, dass die qualitative Evaluation von algorithmisch generierten Ergebnissen – insbesondere im *Machine Learning* – häufig dramatisch zu kurz kommt (Dobson 2021).

2. Was ist ein Algorithmus?

Zunächst ist die Frage zu klären, was ein Algorithmus eigentlich ist. Diese Frage wird in der Literatur abhängig von den jeweiligen Forschungshintergründen und -perspektiven unterschiedlich beantwortet (Tsamados u. a. 2022, 216; Futschek 2006, 160; Kitchin 2017). Der Begriff "Algorithmus" kann etymologisch vom persischen Mathematiker Muhammad ibn Mūsā al-Khwārizmī abgeleitet werden, dessen latinisierter Name *Algoritmi* als Bezeichnung für das Dezimalsystem genutzt wurde (Pfeifer 1993). Mit Bezug auf die mittelalterliche Arithmetik ursprünglich als Rechenvorschrift verstanden^[2], entstanden ab dem 20. Jahrhundert moderne Definitionsversuche durch Wissenschaftler:innen wie Alonzo Church (Church 1936), Alan Turing (Turing 1937) und Konrad Zuse (Zuse 1948). Zuse suchte dabei nach einer formalen Sprache zur Maschinenprogrammierung, dem "Plankalkül" (Zuse 1948). Die aus diesen Anstrengungen hervorgegangenen ersten Programmiersprachen, insbesondere *Algol*, setzten Maßstäbe für die Entwicklung höherer Programmiersprachen, indem sie alle Sprachobjekte mit einem bestimmten Symbolsatz darstellten (Backus u. a. 1960). Dies ebnete den Weg für Computer, die über rein mathematische Verfahren hinausgingen (Anderson 1961).

2.1. Der Algorithmus – Eine Definition

Die klassischen Minimaldefinitionen dieser Ära bieten sich für den Einstieg in das Kapitel an – auch, um die Grenzen dieser Definitionen aufzuzeigen. Harold Stone definiert einen Algorithmus als "a set of rules that precisely defines a sequence of operations such that each rule is effective and definite and such that the sequence terminates in a finite time" (Stone 1972, 8). Etwas ausführlicher als Stone hat etwa zeitgleich sein Kollege Donald Knuth definiert, was einen Algorithmus ausmacht. Er beschreibt fünf grundlegende Eigenschaften (Knuth 1997, 4ff.): Finiteness ("An algorithm must always terminate after a finite number of steps"), Definiteness ("Each step of an algorithm must be precisely defined"), Input, Output, Effectiveness ("in the sense that its operations must all be sufficiently basic that they can in principle be done exactly and in a finite length of time by someone using pencil and paper").

Die klassischen Definitionen nehmen an, dass der Algorithmus direkt ausgeführt wird. In der Realität ist es aber so, dass der Algorithmus erst ausgeführt werden kann, wenn er in einer Programmiersprache implementiert wird. Im Laufe dieser Implementierung werden verschiedene Entscheidungen getroffen, die auf das Ergebnis des Algorithmus Einfluss haben können. Es ist daher notwendig, diese Ebenen klar zu unterscheiden. Yanofsky steuert eine derartige Unterscheidung auf drei Ebenen bei: "Functions", "Algorithms" und "Programs".^[3] Die oberste Ebene "Functions" definiert das abstrakte Ziel, das erreicht werden soll, zum Beispiel die Text-Mining-Methode Topic Modeling – dabei geht es um die inhaltliche Strukturierung umfangreicher Kollektionen von Textdaten. Darunter ist die Ebene, die klassischerweise als Algorithmus bezeichnet wird. Für ein abstraktes Ziel, wie hier Topic Modeling, kann es dann mehrere Algorithmen geben, die das Ziel umsetzen, wie zum Beispiel *Latent Dirichlet Allocation* (LDA) oder *probabilistic Latent Semantic Indexing* (pLSI). Auf der untersten Ebene findet sich dann die Implementierung der Algorithmen, wobei jeder Algorithmus auch mehrfach implementiert werden kann. Im Falle des LDA-Algorithmus sind verschiedene Implementierungen verfügbar, z. B. Mallet (McCallum 2002) und Gensim (Řehůřek und Sojka 2010). Wir werden später in diesem Kapitel auch sehen, dass verschiedene Implementierungen deutlich unterschiedliche Ergebnisse liefern können. Es ist daher unbedingt notwendig bei der Algorithmenkritik auf die konkrete Implementierung zu achten. Yanofskys Definition, wie auch die klassischen Algorithmusdefinitionen, sieht den Algorithmus als entkoppelt von der Welt. Der Algorithmus nimmt Inputs (Daten), macht etwas mit ihnen und produziert dann einen Output. Ob die Inputs für den Algorithmus passend sind oder nicht, hat meistens keinen Einfluss darauf, ob der Algorithmus erfolgreich ein Ergebnis liefert. In allen Fällen steht am Ende ein Output, der, aufgrund seiner Darstellung, absolute Korrektheit impliziert. Dass der Algorithmus, zum Beispiel, nur Sinn macht, wenn er auf Dokumente angewandt wird, die alle circa gleich groß sind, oder dass der Algorithmus mit Daten trainiert wurde, die verzerrt sind – nichts von dem ist im Output zu sehen, auch wenn dies die Ergebnisse verzerrt bzw. wertlos macht. Diese Verzahnungen mit der Welt werden zwar oft in den ursprünglichen Publikationen zu den Algorithmen mehr oder weniger deutlich erwähnt, in der Anwendung sind sie aber selten inkludiert.

Kitchin erweitert die Algorithmusdefinition um die Kontextualisierung und Verankerung in der Welt, indem er den Algorithmus als "ontogenetic, performative and contingent: that is, they are never fixed in nature, but are emergent and constantly unfolding" definiert (Kitchin 2017, 21). Für den Zweck dieses Kapitels ist sowohl die Definition von Yanofsky als auch die von Kitchin relevant. Die drei Ebenen in Yanofskys Definition erlauben eine klare Unterscheidung zwischen der Kritik auf der methodischen Ebene (bei Yanofsky die "Functions") und der praktischen Ebene (Yanofskys "Algorithms" und "Programs"). Auf der praktischen Ebene erlaubt der performative Aspekt der Definition von Kitchin eine Kritik des Algorithmus über seine Outputs. Dadurch können wir die Kritik auf der mathematisch-logischen und code-immanenten Ebene, für die in den Digital Humanities (DH) oft die notwendigen Skills fehlen, hintanstellen und uns auf die *Functions* von Algorithmen, gemäß Yanofsky, und damit einhergehend auf die praktischen Outputs konzentrieren. Der performative Aspekt macht es auch möglich, die Algorithmenkritik auf Methoden anzuwenden, die auf einem aus den Daten gelernten Modell basieren, wie zum Beispiel das Topic Modeling, da bei diesen erst die konkrete Verbindung mit dem Modell die für die Kritik notwendigen Ergebnisse liefert.

2.2. Algorithmisches Denken und Algorithm Literacy

Algorithmisches Denken ist auch für die digitalen Geistes- und Kulturwissenschaften eine zentrale Kompetenz, einerseits hinsichtlich der kreativen Konzeption von Algorithmen, andererseits hinsichtlich ihrer Kritik: Zunächst müssen wir dazu in der Lage sein, ein gegebenes Problem zu verstehen, zu analysieren und in seinen Dimensionen zu spezifizieren. Sodann sind die notwendigen Schritte zu identifizieren, die für die Lösung relevant sind. Schließlich müssen wir auch in der Lage sein, den Algorithmus auf Basis dieser Lösungsschritte korrekt zu konstruieren und dabei auch alle Sonderfälle zu berücksichtigen (Futschek 2006, 160; Jannidis 2017, 89–92). Was für die Konzeption relevant ist, hilft schlussendlich auch dabei, gegebene Algorithmen hinsichtlich ihrer Zielsetzung und Lösungsstrategie einzuordnen und zu bewerten. Das ist insofern entscheidend, da, wie Montfort u. a. (Montfort u. a. 2014), Kitchin (Kitchin 2017) und Bächle (Bächle 2015) zeigen konnten, Algorithmen nicht neutral sind, sondern einen je spezifischen Blick auf die Welt liefern.

Das algorithmische Denken erwächst als Konsequenz aus verschiedenen *digitalen*^[4] Kompetenzen, für die in den letzten Jahren verschiedene *Literacies* geprägt wurden. Genauer wollen wir uns die *Information, Data, Digital, Code* sowie *Algorithm Literacies* ansehen. Zuvor sei festgehalten, dass es für alle diese *Literacies* wie so oft keine einheitliche Definition gibt. Nichtsdestotrotz soll der Exkurs an dieser Stelle nicht allzu sehr ausufern und somit nur jeweils eine Definition präsentiert werden.

Unter *Information Literacy* versteht man die Kompetenz, Informationen zu finden, zu evaluieren, zu erstellen und zu kommunizieren (vgl. American Library Association 2006). *Data Literacy* verbindet zum einen Kompetenzen, Daten zu finden, zu managen, zu interpretieren, kritisch zu evaluieren und ethisch korrekt zu nutzen, aber auch Daten in Informationen sowie nutzbares Wissen zu transformieren sowie zum anderen statistische Daten zu verstehen und zu evaluieren.^[5] *Data Literacy* und *Information Literacy* sind folglich eng miteinander verzahnt. Betrachtet man die Definitionen dieser *Literacies* genauer, ist man schnell an die *DIKW-Hierarchie* (Hobohm 2010) erinnert, bei der aus Daten Informationen und aus Informationen Wissen wird. Beide *Literacies* kommen vor allem zu Beginn des Forschungsprozesses zur Anwendung, sie helfen dabei, das Forschungsdesign vorzunehmen, relevante Daten und Informationen zu finden und ihren Einfluss auf die Forschungsfrage auszumachen.

Die – nun explizit gemeinte – *Digital Literacy* befähigt Personen, Informationen zu finden, mit Anwendungen zu arbeiten, relevante Zielgruppen zu bilden, Webseiten zu bauen, sicher im Netz zu agieren, eigene Daten zu verstehen und zu kontrollieren sowie neue Technologien zu verstehen (vgl. Huang 2018). Damit greift auch sie in das Forschungsdesign ein, geht aber einen Schritt weiter, indem sie sich nicht allein auf die Daten bezieht, sondern bereits Anwendungen einbezieht.

Unter *Code Literacy* wiederum wird die Fähigkeit verstanden, Code zu lesen, zu schreiben und anzupassen sowie Software und Medienkanäle zu erstellen und zu modifizieren (vgl. Pegrum, Hockly und Dudeney 2022). *Code Literacy* kann als eine untergeordnete Kompetenz der *Algorithm Literacy* verstanden werden. Diese vereint die Fähigkeiten, Algorithmen in Plattformen zu verwenden, zu wissen, wie Algorithmen funktionieren, algorithmische Entscheidungen zu evaluieren, mit algorithmischen Operationen umzugehen sowie algorithmische Entscheidungen zu beeinflussen (vgl. Dogruel, Masur und Joeckel 2022; Dogruel 2021). *Code Literacy* wird folglich im Gegensatz zur *Algorithm Literacy* insbesondere dann benötigt, wenn ich einen Algorithmus nicht nur ausführen (und dafür verstehen) muss, sondern selbst in eigenen Code umsetze. Das heißt konkret für die in diesem Handbuchkapitel besonders relevanten Aspekte der einzelnen *Literacies*, dass im ersten Schritt ein *factual knowledge* über Algorithmen und ihre Funktionen und Anwendungsmöglichkeiten gebildet wird, zweitens durch die Ausbildung von *evaluation skills* verwendete Algorithmen und ihren Output kritisch hinterfragt werden können und die Ergebnisse entsprechend der verwendeten Algorithmen interpretiert werden können, drittens *coping strategies* entwickelt werden, wie mit den Limitierungen, die durch den Algorithmus entstehen, umgegangen werden kann, und schließlich viertens *programming skills* erlernt werden, um einen zur jeweiligen Forschungsfrage und dem vorliegenden Material passenden Algorithmus zu entwickeln oder anzupassen.

2.3. Die drei Säulen der Algorithmenkritik

Auf Grundlage dieser Feststellungen lassen sich drei Säulen der Algorithmenkritik ausmachen: Forschungsdesign, Funktionalität und gesellschaftlich-lebensweltliche Auswirkungen. Diese Säulen können nicht unabhängig voneinander betrachtet werden und sind gleichermaßen für eine erfolgreiche Algorithmenkritik relevant. Alle drei Säulen ruhen auf einer epistemologischen Prämisse: Sind Fragestellung und Algorithmus inkompatibel, ist die Kritik arbiträr. Im Folgenden werden die drei Säulen erläutert. In Abschnitt 3 werden drei Arten von Algorithmen (Suchmaschinen, Topic Modeling, Zentralitätsmetriken) entlang der Säulen einer praktischen Kritik unterzogen.

1. Säule: Forschungsdesign

Forschungsdesign bedeutet für Geisteswissenschaftler:innen in der digitalen Forschungspraxis, Erkenntnisinteresse, Problemaufriss, Fragestellung und die darauf aufsetzende Modellierung zu klären, bevor der erste Algorithmus überhaupt zum Einsatz kommt. Diese Reflexion kann als erste Säule bereits konstruktiv in die Algorithmenkritik eingeschlossen werden. Im Anschluss an die epistemologische Prämisse gilt es, methodologische Fragen zu klären, also welche möglichen Herangehensweisen für eine gegebene Fragestellung beziehungsweise den Forschungsgegenstand geeignet sind. Das kann in einem ersten Schritt anhand von geisteswissenschaftlicher Fachliteratur geschehen. Gilt es etwa, unstrukturierte Daten inhaltlich zu erfassen, gibt es diverse Text-Mining-Verfahren, wie etwa das Topic Modeling und dessen populärsten Algorithmus, der Latent Dirichlet Allocation (LDA). Dieser kann über technische Aufsätze oder Einführungswerke aus der Informatik studiert werden, beispielsweise anhand des grundlegenden Aufsatzes zu LDA von Blei, Ng und Jordan (Blei, Ng und Jordan 2003). Steht man vor der Frage, ob man selbst programmieren oder zumindest eine Pipeline aufsetzen muss, hilft der Blick in die Dokumentationen – also in diesem Fall die Beschreibung grundlegender Funktionen von Programmbibliotheken oder Programmen. In Gensim, der Standardbibliothek für Topic Modeling in Python, ergeben sich etwa verschiedene Optionen, das Training der Modelle zu beeinflussen.^[6] Liest man sich weiter ein, wird man erkennen, dass auch LDA wiederum aus verschiedenen Algorithmen zusammengesetzt ist und dass die in Java programmierte Topic-Modeling-Bibliothek Mallet LDA anders implementiert als Gensim. Es bietet sich also an zu testen, welchen Einfluss die unterschiedlichen Implementierungen auf die Ergebnisse des Topic Modeling mit historischen Daten haben. Diese Erkenntnisse sollten direkt in die Entwicklung des Forschungsdesigns einbezogen werden. Erst das Verständnis der grundsätzlichen Methode und der Vorannahmen eines Algorithmus kann überhaupt helfen, die Ergebnisse einzuordnen und zu überprüfen. Zusammengefasst geht es bei den Modi der ersten Säule um

- die “äußere” Algorithmenkritik, damit die grundsätzliche Funktionsweise des Algorithmus verstanden wird;
- einen Versuchsaufbau, der zuvor recherchierte Varianten und Modalitäten von Algorithmen vergleichbar macht;
- die Entwicklung einer Evaluationsroutine, die das systematische Beurteilen und Vergleichen der Ergebnisse ermöglicht.

2. Säule: Funktionalität

Der Modus der funktionalitätsseitigen Säule der Algorithmenkritik bildet gewissermaßen den praktischen Kern der Algorithmenkritik im Sinne einer konkreten Funktions- und Anwendungskritik. In der Informatik wird häufig die formale Semantik zur Analyse von formalen Sprachen oder der Logik von Algorithmen angewendet.^[7] Doch ist dieser text-immanente Modus für Geisteswissenschaftler:innen sinnvoll oder überhaupt praktikabel? Angesichts des erforderlichen Vorwissens auf der einen und des eindeutigen Anwendungsbezugs in den DH auf der anderen Seite kommt dieser Modus der Algorithmenkritik nicht in Frage.^[8] Auch die aus den *Science and Technology Studies* (STS) – und damit aus den Geistes- und Sozialwissenschaften – hervorgegangenen *Critical Code Studies* (CCS) bleiben im Modus der Text-Immanenz und verweisen eher auf Konventionen als auf die Funktionalität von Algorithmen, indem sie Code als literarische Quellen behandeln (Marino 2006; 2020). Für Geisteswissenschaftler:innen, die nicht Algorithmen als *Gegenstand* der Kritik, sondern deren *Ergebnis* kritisch reflektieren möchten, bietet sich ein ergebnisorientierter Modus der Kritik an. Als Kernkonzepte dieses Modus in der Informatik wären hier Evaluation und Testing zu nennen, in der Regel hochgradig spezialisierte Routinen der Informatik und Softwareentwicklung (vgl. Laboon 2016; Feuerriegel 2016). Doch das Testing fokussiert eher auf den Bereich der Entwicklung von Algorithmen und von auf diesen aufbauenden Programmen. Zur Evaluation werden häufig abstrakte mathematische, daher quantitative, Metriken genutzt, die für Fachfremde schwierig nachzuvollziehen und einzuordnen sind. So wurde in einem Review über Topic-Modeling-Evaluation festgestellt, dass die wenigsten Verfahren überhaupt eine qualitative Untersuchung der Ergebnisse vorsehen (Hoyle u. a. 2021, 1f.). Ein eng damit verbundenes Problem ist die Evaluation von Algorithmen auf Basis von Benchmark-Datasets. Da die Datengrundlage die Qualität von maschinellem Lernen signifikant beeinflusst, muss immer eine Bewertung auf Grundlage domänenspezifischer Daten erfolgen (etwa lebensgeschichtliche Interviews in der Oral History) (Koch u. a. 2021). Auch im Bereich der Suche ist eine kritische Beurteilung kompliziert. Dort basieren die Evaluationsmetriken primär darauf, wie viele “relevante” Dokumente in den Top-Ten-Suchergebnissen sind (Valcarce u. a. 2018), wobei sowohl die Zusammenstellung des Suchkorpus, wie auch die Bewertung der “Relevanz”, selten kritisch analysiert wird. Aus geisteswissenschaftlicher Sicht ist das nicht vorstellbar – die Abstraktion von der Quelle in die Vektorisierung ist bereits ein gravierender Eingriff in die traditionell hermeneutische Forschungspraxis. Dass die Qualität nur anhand quantitativer Metriken und ohne einen Blick in die jeweils unter veränderten Bedingungen produzierten Ergebnisse beurteilt wird, kann der inhaltlichen Komplexität und dem semantischen Gehalt sprachlicher sowie der symbolischen Beschaffenheit nicht-sprachlicher Quellen nicht Rechnung tragen (Dobson 2021).

Ein banales Hilfsmittel weist hier möglicherweise einen “sinnvolleren” Weg. Bei der Entwicklung von Software wird empfohlen, Zwischenschritte zu überprüfen, indem möglichst häufig Befehle zur Ausgabe von Zwischenergebnissen in den Code eingebaut werden (Feuerriegel 2016, 11). Diesen Ansatz, der zwischen Implementierung und Anwendung verortet ist, können wir für die Algorithmenkritik übernehmen. Zum Beispiel können auch im oben thematisierten Topic Modeling Ergebnisse in Form von Wortlisten ausgegeben oder die

Topics mit entsprechenden Tools in den Text zurückverfolgt werden, um die Aussagekraft der Ergebnisse und gleichzeitig der abstrakten Evaluationsmetriken zu überprüfen. Wir werden dieses Beispiel ausführlich in Kapitel 3.2 besprechen und für einen deskriptiv-komparativen Modus der Algorithmenkritik plädieren, der die Funktion des Algorithmus vom Ergebnis her aufwickelt.

3. Säule: gesellschaftlich-lebensweltliche Auswirkungen

Die Diskrepanz von quantitativer und qualitativer Evaluation führt uns zur dritten Säule der Algorithmenkritik. Diese umfasst die gesellschaftlich-lebensweltlichen Auswirkungen von Algorithmen, beispielsweise deren Suggestionskraft, deren Vermögen, Phänomene sowohl sichtbar zu machen als auch zu verschleiern, und, nicht zuletzt, die Wechselwirkungen zwischen Algorithmen und Forschenden. ^[9] Betrachtet man etwa den letztgenannten Aspekt, könnte man die Frage stellen, wie man Methodenpositivismus begegnen kann, wenn die Funktionsweise von Algorithmen, die Veränderbarkeit von Parametern und die Vergleichbarkeit von Ergebnissen für fachfremde Anwender:innen nicht erreichbar oder zumindest intransparent sind. Hier beginnt der Modus der Algorithmenkritik mit der Aufklärung über Funktionsumfang und Variablen eines Algorithmus, um überhaupt einen vergleichenden Versuchsaufbau zu ermöglichen. Werden Algorithmen positivistisch angewendet, ohne deren Funktionsweise zu reflektieren, hat das gravierende Auswirkungen auf das Forschungsverhalten und die Quellenkritik im digitalen Zeitalter. Dieses Szenario gilt stellvertretend für die gesellschaftliche Relevanz algorithmenkritischer Kompetenzen. Im nächsten Teil wird gezeigt, wie verschiedene Algorithmen Inhalte aus geisteswissenschaftlichen Quellen filtern und verzerren, während sie gleichzeitig Objektivität verheißen. Dieser spezifische Blick der Algorithmen auf die Daten und mithin die Welt ist ein weiterer Gegenstand der Algorithmenkritik, der sich in den letzten Jahren in den Kultur- und Sozialwissenschaften etabliert hat (Bächle 2015; Seyfert und Roberge 2017).

3. Praxis

Die Algorithmenkritik muss natürlich immer an den spezifischen Algorithmus angepasst werden. Offensichtlich kann das hier nicht für alle Algorithmen gezeigt werden, daher haben wir exemplarisch drei Methodenbereiche ausgewählt und demonstrieren an diesen Aspekte der Algorithmenkritik.

3.1. Suchen im Archiv

Suchmaschinen bieten aufgrund ihrer Alltäglichkeit ein anschauliches erstes Beispiel für die Interaktionen zwischen Nutzenden, Algorithmus und Daten. Die Interaktion mit dem Suchsystem ist dabei sehr einfach, unabhängig davon, ob die Interaktion manuell über eine Eingabemaske oder mittels Schnittstelle (API) stattfindet: Eine Suchanfrage wird an das Suchsystem gestellt und das Suchsystem antwortet mit einer Liste an Ergebnissen, die der Suchanfrage entsprechen.

In dieser Struktur ist die Suchmaschine eine "Black Box", also ein Konstrukt, in das wir nicht hineinblicken können. Das wird oft als problematisch gesehen, da es unmöglich ist, die Einflüsse des Suchalgorithmus auf das Ergebnis nachzuvollziehen. Bächle übernimmt etwa den Begriff der Gatekeeper und erklärt Suchmaschinen zu "Schleusenwärtern des Wissens, die Informationen entsprechend bestimmter Algorithmen nutzerspezifisch sortieren und präsentieren und somit den Zugriff auf Wissen steuern" (Bächle 2015, 22; vgl. auch Cardon 2010; Cardon 2017, 131; Lewandowski 2021). Die Frage, ob und inwieweit man in die Black Box hineinblicken kann, ist jedoch eine Ablenkung, denn wie sich gleich zeigen wird, ist ein vollständiges Verständnis für die exakten Abläufe in der Praxis nicht realistisch möglich.

Wichtig hierbei sind die Aspekte "vollständig" und "realistisch" (Tsamados u. a. 2022; vgl. dazu Passig 2017). Um die Abläufe vollständig zu verstehen, bräuchten wir eine exakte Kopie der Daten sowie vollen Zugriff auf den Code des Suchalgorithmus und die Hardware, auf der die Suche durchgeführt wurde, da dies alles die Ergebnisse beeinflussen kann. Damit könnten wir dann händisch die Berechnungen des Suchalgorithmus nachvollziehen. Sobald die Menge an Daten, die durchsucht wird, mehr als ein paar Tausend Dokumente beinhaltet, würden diese Berechnungen Wochen oder Monate dauern. Um die Wahrscheinlichkeit von Fehlern zu minimieren, müsste die Arbeit auch mindestens von zwei unabhängigen Personen durchgeführt und die Ergebnisse dann verglichen werden. Dieser Aufwand müsste für viele Suchanfragen gemacht werden, da nie ausgeschlossen werden kann, dass das Ergebnis nur für diese eine Suchanfrage korrekt bzw. sinnvoll war. Dies ist ein Arbeitsaufwand, der nicht realisierbar ist.

Dieser Aufwand ist für unsere Zwecke der Algorithmenkritik aber auch nicht notwendig. Für unsere Zwecke ist es ausreichend, den Einfluss des Suchsystems auf die konkreten Forschungsergebnisse beurteilen zu können.

3.1.1. Forschungsdesign

Der erste Schritt der Kritik ist es, zu betrachten, wie der Algorithmus die Welt repräsentiert bzw. welche Annahmen über die Welt in den Algorithmus eingeflossen sind. Diese Information beziehen wir, wie oben besprochen, aus der Literatur. Im Fall der Suchmaschine ist dies meistens das "Vector Space Model" (Salton, Wong und Yang 1975). In diesem Modell wird jedes Dokument auf eine Sammlung an Tokens reduziert, wobei jedes Token eine einzigartige Abfolge von Zeichen ist, die aufgrund irgendeiner Logik von den davor und danach existierenden Tokens abgetrennt ist. Zum Beispiel könnte der Satz "Wir kommen, denn es ist spät." in die Tokens "Wir", "kommen", ",", "denn", "es", "ist", "spät", "." entlang der Leer- und Satzzeichen aufgeteilt werden. Jedes Dokument wird als ein Vektor repräsentiert, wobei jeder Wert im Vektor angibt, wie oft das Token im Dokument vorkommt. In dem Ein-Satz-Dokument würde der Vektor dann wie in Tabelle A aussehen. Eine Suchanfrage wird dann ebenso in einen Vektor umgewandelt. Dieser Vektor wird schließlich mit den Vektoren aller Dokumente verglichen und aufgrund der Ähnlichkeit werden die Dokumente dann gereiht und angezeigt.

,	.	denn	es	ist	kommen	spät	wir
1	1	1	1	1	1	1	1

Tabelle A: Beispiel für einen Vektor für ein Dokument, das aus einem einzigen Satz "Wir kommen, denn es ist spät." besteht. Die erste Zeile listet die Tokens auf, aus denen der Satz besteht, und die zweite Zeile deren Häufigkeit im Dokument. Da alle Tokens nur einmal vorkommen, haben alle Häufigkeiten den Wert 1.

Da die Suche primär dazu dient, eine Sammlung an Daten aus einem digitalen Archiv zu extrahieren, ist der Einfluss auf das Forschungsdesign begrenzt. Primär muss man sich bewusst machen, dass es aufgrund der Ähnlichkeitssuche nicht garantiert ist, dass alle Suchergebnisse, die die Suche zurückliefert, auch alle Elemente der Suchanfrage beinhalten. Dies ist besonders der Fall, wenn die Suchanfrage länger ist, weil dann ein einzelnes Element der Suchanfrage nur einen begrenzten Einfluss auf das Ähnlichkeitsmaß hat. Ebenso ist es nicht garantiert, dass die Elemente der Suchanfrage auch in der Reihenfolge der Suche gefunden wurden. Für das Forschungsdesign bedeutet dies, dass nach der Erstellung des Datensatzes die Dokumente noch einmal genauer unter die Lupe genommen werden müssen, um sicherzustellen, dass möglichst wenig Dokumente darin sind, die nur teilweise der Suche entsprechen oder die Schlüsselwörter nicht in der gewünschten Zusammenstellung enthalten.

3.1.2. Funktionalität

Nun können die Konsequenzen der Grundannahmen des Algorithmus genauer analysiert werden, im ersten Schritt immer noch aufgrund der Definition in der Literatur. Wie schon beschrieben, basiert die Suche auf der Ähnlichkeit zwischen Vektoren von Tokens. Diese Tokens selbst sind im Suchmodell als Abfolge von Zeichen definiert. In diesem Modell sind die Wörter "Kind" und "Kinder" genauso unterschiedlich wie die Wörter "Kind" und "Kartoffel", aber auch genau so ähnlich wie die Wörter "Kind" und "Tochter". Die für uns klaren semantischen Unterschiede gehen in dieser Token-Repräsentation verloren. Die Konsequenz davon ist, dass das Dokument nicht gefunden wird, wenn der Suchtoken nicht genau dem Token im Dokument entspricht. Das kann dazu führen, dass signifikante Mengen an Dokumenten nicht gefunden werden, da die Suchanfrage und die Dokumente unterschiedliche Wörter oder Varianten der Wörter für den gleichen Inhalt nutzen (Markkula und Sormunen 2000). Damit Nutzende nicht immer alle Varianten explizit in der Suchanfrage angeben müssen, werden oft zwei Methoden angewandt, um das Variantenproblem zu reduzieren: Entweder werden die Tokens in den Dokumenten und der Suchanfrage vereinfacht oder die Suchanfrage wird automatisch um Wortvarianten erweitert.

Bei der Tokenvereinfachung werden durch Techniken wie *Stemming* (Singh und Gupta 2017) und Lemmatisierung die Tokens auf ihre Wortstämme zurückgeführt. Dadurch werden "Kind" und "Kinder" auf den gleichen Stamm "Kind" reduziert und daher findet die Suche nach dem einen auch das andere. Der Nachteil ist, dass weitere semantische Informationen verloren gehen und es unmöglich wird, die Suche auf spezifische Fälle zu begrenzen, wie zum Beispiel nur "Kind" oder nur „Kinder“.

Bei der automatischen Erweiterung der Suchanfrage (Azad und Deepak 2019) wird mittels eines Thesaurus (Nakade, Musaev und Atkison 2018) oder unter Zuhilfenahme von Wortmodellen wie *word2vec* (Kuzi, Shtok und Kurland 2016) die Suchanfrage mit ähnlichen oder verwandten Wörtern erweitert. Dadurch kann eine Suche nach "Kind" mit den Wörtern "Sohn" und "Tochter" erweitert werden. Dies ist weniger problematisch, wenn allgemeinere Begriffe um spezifische erweitert werden, aber bei der Erweiterung spezifischer Begriffe mit allgemeineren Begriffen, zum Beispiel indem die Suche nach "Tochter" mit "Kind" erweitert wird, können die Suchergebnisse signifikante Mengen an Dokumenten enthalten, die nicht der Suchintention entsprechen.

Für die Algorithmenkritik müssen wir also untersuchen, welche Sprache für die Dokumente genutzt wurde, wie diese mit den Suchwörtern überlappt, ob und wie die Tokens der Dokumente vorbearbeitet wurden, und ob die Suchwörter automatisch expandiert werden.

Idealerweise wären diese Informationen über die Suchmaschine als Dokumentation verfügbar, aber das ist oft nicht der Fall. Hier können

wir also den Ansatz der ergebnisorientierten Algorithmenkritik anwenden, indem verschiedene Suchanfragen abgesetzt werden, die darauf abzielen, ein Verständnis für die Daten und Verarbeitung der Suchmaschine zu entwickeln. Wir können das jetzt am Deutschen Textarchiv^[10] ausprobieren. Als erstes untersuchen wir, ob eine Tokenvereinfachung durchgeführt wurde. Dazu nutzen wir das “Kind” / “Kinder”-Beispiel und suchen nach beiden Begriffen. Die Suche nach “Kind” liefert 102.266 Ergebnisse, während die Suche nach “Kinder” 102.267 Ergebnisse liefert. Ein Unterschied von nur einem Dokument deutet stark darauf hin, dass eine Tokenvereinfachung durchgeführt wurde. Bestätigt wird das durch die Top-Ten-Suchergebnisse, die für beide Suchanfragen gleich sind. Zusätzlich zeigt das Deutsche Textarchiv einen Ausschnitt aus dem Text an und zeichnet den Suchbegriff visuell aus. Bei der Suche nach “Kind” erhalten wir Suchergebnisse, bei denen das Wort “Kinder” ausgezeichnet ist, was ein weiterer Hinweis auf die Vereinfachung ist.

Wir können jetzt eine weitere Suchanfrage nach “kind” absetzen. Die Top-Ten-Ergebnisse sind hier gleich, aber die Suche nach “kind” liefert mit 102.410 Dokumenten 143 Dokumente mehr. Die Unterschiede in den Dokumentzahlen sind relativ klein, was die reinen Zahlen aber verstecken ist, dass es sich hier sehr wohl um größere Unterschiede zwischen den Dokumentlisten handeln könnte und dass nur zufällig die Gesamtanzahlen sehr ähnlich sind. Hier müssten im nächsten Schritt die drei Dokumentlisten heruntergeladen werden und dann die Dokumente identifiziert werden, die nicht in allen drei Suchanfragen vorkommen. Eine Analyse dieser Dokumente könnte dann Aufschlüsse darüber geben, woher diese Unterschiede in den Ergebnissen der sehr ähnlichen Suchanfragen kommen.

Die automatische Suchanfragenerweiterung zu identifizieren ist schwieriger und das Ergebnis ist nicht so sicher wie bei der Identifikation der Tokenvereinfachung. Wir suchen zuerst nach dem Wort “Tochter” und wählen dann eines der Dokumente aus, z.B: “Abriß der neuesten Staatswissenschaft der vornehmsten Europäischen Reiche und Republicken”. Wir können dann den Dokumententitel zusammen mit dem Wort “Kind” als Suche absetzen. Wenn eine Suchanfragenerweiterung eingesetzt wird, dann wäre eine Erweiterung unseres Suchbegriffes “Kind” durch “Tochter” und “Sohn” sinnvoll und wir würden das Dokument, das mittels “Tochter” gefunden wurde, auch für diese Suche finden. In diesem Beispiel erhalten wir jedoch für die Kombination mit “Kind” kein Ergebnis. Es hat also keine Erweiterung der Suche nach “Kind” um den Begriff “Tochter” stattgefunden. Um eine generelle Suchanfragenerweiterung auszuschließen, müsste dieser Vorgang mehrfach wiederholt werden – damit könnte ausgeschlossen werden, dass wir nicht nur auf einen Ausnahmefall gestoßen sind.

Andere Aspekte der Sprache können auf ähnliche Weise untersucht werden. In einem historischen Textarchiv kann zum Beispiel eine Suche nach älteren Schreibweisen von Wörtern genutzt werden, um zu sehen, ob diese in irgendeiner Weise nachbearbeitet wurden. Zum Beispiel können wir nach “sein” und “seyn” suchen und würden sehen, dass wie beim “Kind” / “Kinder”-Beispiel keine Unterschiede im Ergebnis festzustellen sind. Der Grund dafür könnte sein, dass das Suchsystem eine Fuzzysuche anwendet, in der zwei Token nicht genau gleich sein müssen, um vom Suchsystem als äquivalent gewertet zu werden. Alternativ könnte auch eine Art phonetischer Suche laufen, da “ei” und “ey” phonetisch gleich sind. Wir können da ein bisschen weiter nachbohren, indem wir zwei weitere Suchen “syen” und “syne” absetzen. Die erste Suche “syen” liefert wieder die gleichen Ergebnisse. Die Suche nach “syne” liefert jedoch nur ein einziges Dokument, in dem das Suchwort “syne” mit dem Wort “seny” gleichgestellt wird. Diese Ergebnisse deuten darauf hin, dass das Suchsystem eine Art phonetische Ähnlichkeit nutzt.

In der Praxis müsste man hier dann noch weitere Suchanfragen absetzen, die für die Forschungsfrage relevant sind, damit man ein globaleres Verständnis dafür entwickelt, wie sich die Suche bei diesen Suchanfragen und bei der Forschungsfrage verhält.

3.1.3. Gesellschaftlich-lebensweltliche Auswirkungen

Wichtig ist jedoch, dass bei der Algorithmenkritik nicht nur in Betracht gezogen werden muss, was der Algorithmus macht, sondern auch was der Algorithmus *nicht* macht. Egal, wie gut man den Suchalgorithmus verstanden hat, es ist unmöglich zu sagen, was das Suchsystem fände, wenn die Suchanfrage leicht variierte. Selbst wenn die Suchanfrage “perfekt” ist, kann das Suchsystem nicht sagen, wie viele Dokumente nicht gefunden werden, weil die Dokumente Tipp- oder OCR-Fehler oder Synonyme enthalten oder weil die Dokumente überhaupt nicht digitalisiert wurden (Putnam 2016).

Dies kann einen signifikanten Einfluss auf den digitalen Forschungsprozess haben. Im klassischen Forschungsprozess im Archiv, im Museum, in der Galerie oder in der Bibliothek kann man meistens sehen, was man nicht eingesehen hat, in welche Box man nicht geschaut hat, welchen Korridor man nicht durchsucht hat. All dies ist im Digitalen nicht möglich und äquivalente Informationen sind nur selten verfügbar. Neben der Anzahl an gefundenen Objekten wird nur selten auch die Gesamtanzahl an Dokumenten angezeigt, nur um daran zu erinnern, dass der Großteil der Dokumente nicht durch die Suche gefunden wurde. Die Anzahl an nicht-digitalisierten Objekten ist generell überhaupt nirgendwo zu finden.

Das Erstellen eines Forschungskorpus mittels Suchanfrage an die Suchmaschine einer oder mehrerer GLAM^[11]-Institutionen stellt oft den ersten Schritt in einer Forschungsarbeit da. Wie wir gesehen haben, hat die genaue Konfiguration der Suchmaschine relativ unkontrollierbaren Einfluss auf die Ergebnisse, die angezeigt werden. Es ist daher wichtig, dass das aus den Suchmaschinen extrahierte Korpus in weiteren Schritten validiert und von falschen Suchergebnissen bereinigt wird, bevor es in den Analyseprozess übernommen wird. Der akribische Blick in die Ergebnisse auf Grundlage eines konsequenten Vergleichs schärft das Bewusstsein für die Eigenheiten und insbesondere die Varianzen algorithmischer Funktionalität.

3.2. Topic Modeling

Nicht nur die Geschichtswissenschaften stehen vor der Herausforderung, die stetig wachsende digitale Quellenbasis zu erfassen und zu erschließen. Insbesondere kleine Forschungsdatenarchive und -repositorien können meist nicht gewährleisten, die erhobenen Daten vollständig zu durchmessen oder gar zu durchleuchten. Damit bleiben wir zunächst im Bereich der Heuristik. Clustering-Algorithmen erscheinen für solche Anwendungsszenarien vielversprechend. Als Fallbeispiel für eine kritische Verwendung dieser Art von Algorithmen in den Geisteswissenschaften soll hier ein Korpus lebensgeschichtlicher Interviews aus der Oral History herangezogen werden. Das Beispielkorpus besteht aus etwa 130 Interviews und hat einen Umfang von 3,5 Mio. Wörtern. Es handelt sich um das Pionierprojekt der Oral History in Deutschland, *Lebensgeschichte und Sozialkultur im Ruhrgebiet* (Niethammer 1983). Der Untersuchungszeitraum – 1920 bis 1980 – lässt Nationalsozialismus, Krieg und Industrie als dominante Themen vermuten, was ein Blick in die aus dem Forschungsprojekt entstandenen Studien bestätigt. Dennoch bergen diese – und tausende weitere im Archiv „Deutsches Gedächtnis“ des Instituts für Geschichte und Biographie der FernUniversität Hagen vorliegende – Interviews ein kaum bezifferbares Spektrum größerer und kleinerer Themen, die nie vollständig erfasst und etwa verschlagwortet wurden.^[12] Genau dies soll mit Unterstützung von Computern geschehen. Gemäß der epistemologischen Prämisse wäre in diesem Fall ein exploratives, induktives Verfahren zu finden, um die teils unbekanntes Inhalte automatisiert mittels eines Algorithmus zu identifizieren. Darüber hinaus suchen wir ein deskriptives Verfahren und keines, das analytisch – und somit interpretierend – arbeitet. Eine erste Recherche liefert eine ganze Fülle von Methoden aus den Bereichen Korpuslinguistik, Natural Language Processing und Machine Learning. Einführungswerke in die Digital Humanities lenken den Blick auf Topic Modeling, eine Methode, die anhand von Worthäufigkeitsverteilungen so etwas wie „Themen“ aus Korpora erschließen kann. Das Verfahren erfüllt die epistemologische Prämisse: Es ist explorativ und die „Topics“ sind Listen von besonders häufig in Nachbarschaft (demselben Dokument bzw. der definierten Bezugsgrenze) vorkommenden Wörtern. Doch beim Blick in die Literatur zum Topic Modeling wird schnell klar: Das Topic Modeling gibt es nicht. Vielmehr verkörpert es eine Familie von Verfahren, die dasselbe Ziel verfolgen. Ihre Algorithmen greifen teils ineinander respektive bauen aufeinander auf, teils sind es komplett unabhängige Algorithmen. Damit bewegen wir uns auf der Ebene der ersten Säule der Algorithmenkritik: dem Forschungsdesign.

3.2.1. Forschungsdesign

Topic Modeling ist ein Verfahren zum Clustering umfangreicher Dokumentensammlungen. Ziel der Methode ist es, statistisch signifikante Sprachgebrauchsmuster, die sogenannten *Topics*, zu identifizieren und diejenigen Dokumente zu gruppieren, die diese Muster miteinander teilen. In den Digital Humanities wird das Verfahren über Explorations- und Klassifikationsaufgaben hinaus eingesetzt, um unter anderem die Entwicklung thematischer Trends sowie von Diskursen zu untersuchen.^[13] Solche Anwendungen sind von der Annahme motiviert, dass die computationell generierten Topics semantisch kohärent sind, d. h., es wird davon ausgegangen, dass die das Topic bildenden Begriffe etwas miteinander gemein haben (vgl. Schmidt 2012, 49). Forschungstheoretische Grundlage solcher Annahmen sind im Konzept der distributionellen Semantik begründet (Turney und Pantel 2010; Grundlage: Harris 1954; Firth 1957), das zahlreichen Methoden der Computerlinguistik zugrunde liegt. Hiernach ergibt sich die Bedeutung lexikalischer Einheiten (Wörter bzw. Token) aus ihrer gemeinsamen Vorkommenshäufigkeit mit anderen lexikalischen Einheiten in einem bestimmten Kontext (z. B. auf Ebene der Dokumente, Absätze oder einzelner Sätze). Diese Häufigkeitsbeziehungen werden in der Regel räumlich über Koordinaten im sogenannten *vector space* (siehe oben) repräsentiert. Dadurch soll es ermöglicht werden, die Textdaten mit dem „semantisch blinden“ Computer (Schwandt 2018, 108, 133) gewissermaßen auf ihrer Bedeutungsebene verarbeitbar zu machen, worauf auch Methoden wie Topic Modeling aufsetzen (vgl. Althage 2022, 256, 266f.).

Dabei dürfen Topics indes nicht leichtfertig mit Themen gleichgesetzt werden. Statistisch-mathematisch formuliert ist ein Topic zunächst lediglich eine Wahrscheinlichkeitsverteilung über das gesamte Korpus, die beschreibt, welche Wörter aus dem Gesamtvokabular des Korpus gehäuft gemeinsam und in Nachbarschaft vorkommen. Aus diesen gemeinsamen Häufigkeiten ergeben sich dann Wortcluster, die zu unterschiedlichen Anteilen in den einzelnen Dokumenten der Textsammlung vertreten sind (siehe die Beispiele in Tabelle C). Der Begriff „Topic“ kann damit zwar Assoziationen an den Forschungsbereich der Topik oder an Konzepte wie „Topoi“ wecken^[14], bildet jedoch reine Wahrscheinlichkeiten des gemeinsamen Vorkommens (Kookkurrenz) von Wörtern eines Textes ab – keine semantischen

Zusammenhänge.^[15] Das zeigt sich auch daran, dass das zugrundeliegende logisch-mathematische Modell nicht auf Textdaten beschränkt ist. Es kann auch eingesetzt werden, um Muster in Bilddaten oder Daten der Bioinformatik zu identifizieren (zum Beispiel Hooft u. a. 2016).

Bei LDA (*Latent Dirichlet Allocation*), das hier genauer in den Blick genommen werden soll, handelt es sich um ein generatives probabilistisches Modell, dem die Annahme zugrunde liegt, dass sich Dokumente aus einer *latenten* thematischen Struktur zusammensetzen, die sich aus der Verteilung gemeinsam vorkommender oder in Nachbarschaft befindlicher Terme ablesen lässt. Die Verteilungen werden über einen randomisierten Prozess ermittelt, der von einer Reihe von Parametern abhängig ist, die unmittelbare Auswirkungen auf das Modell haben, darunter die Anzahl zu generierender Topics und Iterationen (Durchläufe zum Trainieren des Modells) sowie die Hyperparameter (Einstellungen, um das Modell zu optimieren). Abhängig von der Implementierung des Algorithmus haben Nutzende unterschiedliche Konfigurationsmöglichkeiten. Gebrauchsfertige Werkzeuge wie der Dariah Topics Explorer (Simmler, Vitt und Pielström 2019) etwa ermöglichen nur die Konfiguration der Topic-Anzahl und Iterationen, Hyperparameter lassen sich dagegen über eigene codebasierte Lösungen in Programmiersprachen wie R oder Python anpassen. Hierbei gilt es, wie oben angedeutet, zu berücksichtigen, dass es nicht nur eine Implementierung für LDA gibt. Grund dafür ist, dass es sich bei der Berechnung der Topics um eine Optimierungsaufgabe handelt, die nur annäherungsweise durch die sogenannte Approximation zu erreichen ist. Für diese Annäherung gibt es ganz unterschiedliche Berechnungsmöglichkeiten. Gebräuchlich für Implementierungen von LDA sind vor allem zwei Ansätze: a) *Variational Bayes Inference* (Blei, Ng und Jordan 2003, 1003–1007) und b) *Gibbs Sampling* (Griffiths und Steyvers 2004, 5229–5230). Je nachdem, welcher dieser sogenannten Inferenzalgorithmen eingesetzt wird, können sich die Modellierungsergebnisse erheblich unterscheiden. Mit Mallet (McCallum 2002) und Gensim^[16] liegen in den digitalen Geisteswissenschaften etablierte und gut dokumentierte Werkzeuge vor, die auf entsprechend unterschiedliche Ableitungsstrategien setzen. Mallet wurde in der Programmiersprache Java geschrieben und federführend von David Mimno und Andrew McCallum an der University of Massachusetts entwickelt. Während Mallet auf das Gibbs Sampling als Inferenzalgorithmus setzt, nutzt das in der Programmiersprache Python geschriebene Gensim den Variational-Bayes-Algorithmus. Um die Tragweite dieser Unterschiede einschätzen zu können, ergibt eine Algorithmenkritik nach den Maßstäben formaler Semantik für die Geisteswissenschaften wenig Sinn. Zwar ist es unumgänglich, sich mit den konkreten formalen Eigenschaften des Algorithmus sowie der Beschaffenheit des Softwarecodes auseinanderzusetzen – auch, um die ursprünglichen Entwicklungsziele des Verfahrens angemessen zu berücksichtigen. Wie sich die Anlage des Algorithmus indes auf das konkrete Ergebnis auswirkt, ist aus den Formeln und Codebestandteilen nur bedingt ableitbar. Diese Unterschiede – etwa zwischen Inferenzalgorithmen verschiedener Topic-Modeling-Bibliotheken – sollten im Forschungsdesign berücksichtigt werden.

Um die vorliegenden Quellen – lebensgeschichtliche Interviews – verarbeiten zu können, müssen diese allerdings vorbereitet werden (Stichwort: *Preprocessing*). Die Vorverarbeitung der Daten stellt erhebliche Eingriffe dar, die dazu dienen, die Komplexität der Quellen zu reduzieren. Dieser Vorgang ist als Teil der Datafizierung in seinen Ausmaßen auf Anwendbarkeit und Angemessenheit zu prüfen und transparent zu dokumentieren. Zunächst ist es erforderlich, die Volltexte in Tokens umzuwandeln, das heißt, jede Zeichenkette, die als Wort definiert wird, wird als einzelne Identität in eine Datenstruktur, zum Beispiel eine Liste, eingefügt. Das Resultat hiervon ist eine Liste (Korpus) von Listen (Dokumente) mit den einzelnen Zeichenketten. Ein weiterer unabdingbarer Schritt auf dem Weg zur Datafizierung der Texte stellt die Entfernung der Satzzeichen dar, da der Algorithmus nur Zeichenketten “sieht” und nicht zwischen Buchstaben und nicht-Buchstaben unterscheiden kann, die Interpunktion für eine inhaltliche Klassifikation aber in der Regel keine bedeutungstragende Rolle spielt (*No Punctuation*). Obligatorisch ist in der Regel auch die Reduktion der Zeichenketten auf Kleinschreibung; das verhindert, dass am Satzanfang großgeschriebene Nicht-Nomen und deren kleingeschriebene Entsprechung in der Satzmitte als unterschiedliche Wörter eingeordnet werden (*Lower Casing*). Allerdings ergibt sich daraus auch ein sogenannter Loss: “ich kriege etwas” und “die schlimmen Kriege” werden allenfalls noch durch eine erfolgreiche Kontextualisierung sauber disambiguiert – ein Aspekt der im Rahmen des Forschungsdesigns stets individuell abzuwägen ist. Weitere Schritte bieten sich an, sind aber für die Durchführung des Topic Modeling nicht zwingend erforderlich, können allerdings die Qualität der Modellierungsergebnisse positiv beeinflussen. Je nach Forschungsinteresse und Quellenkorpus kann es etwa sinnvoll sein, sogenannte Stoppwörter (*Stopwords*) zu entfernen, das sind besonders häufige und nicht sinntragende Wörter (in erster Linie Artikel, Pronomen, Präpositionen etc.). Auch Verfahren wie *Lemmatisierung* und *Part-of-Speech-Tagging* (POS-Tagging) können zum Einsatz kommen. Bei der Lemmatisierung werden die Wörter auf ihre Grundformen reduziert, um Mehrfachnennungen gleicher Wörter etwa in unterschiedlicher Deklination zu vermeiden. Durch POS-Tagging wiederum ist es möglich, lediglich bestimmte Wortarten bei der Modellierung zu berücksichtigen (beispielsweise Nomen, Verben, Adjektive). Jeder einzelne dieser und anderer Vorverarbeitungsschritte basiert wiederum auf algorithmischen Lösungen, die interpretative Eingriffe in die Datenbasis darstellen; sie sollten daher stets kritisch reflektiert werden und nur insofern zum Einsatz kommen, als sie dem Forschungsziel dienen.^[17] Tabelle B zeigt, wie sich die Ergebnisse des Topic Modelings unter dem Einfluss verschiedener Schritte des Preprocessing verändern.

[,man', ,äh', ,die', ,wir', ,und', ,Das', ,von', ,zu', ,dort', ,oder', ,diese', ,sich', ,Leute', ,eine', ,ein', ,aus', ,den', ,wieder', ,eigentlich', ,nach', ,dem', ,Die', ,noch', ,wie', ,wurden', ,einem', ,im', ,bis', ,heute', ,nur', ,Wir', ,nicht', ,musste', ,konnte', ,usw.', ,einen', ,damals', ,Bergbau', ,mehr', ,mit', ,Mark', ,Es', ,hier', ,du', ,uns', ,Man', ,unter', ,alle', ,es', ,um']	Raw
[,wir', ,man', ,das', ,die', ,dann', ,ja', ,und', ,der', ,haben', ,jetzt', ,da', ,wenn', ,ist', ,du', ,hat', ,es', ,sind', ,mal', ,dass', ,den', ,auch', ,immer', ,uns', ,war', ,oder', ,mit', ,aber', ,damals', ,noch', ,von', ,wieder', ,wo', ,zeche', ,auf', ,gesagt', ,heute', ,kann', ,bergbau', ,im', ,wie', ,waren', ,steiger', ,ich', ,gemacht', ,dort', ,oben', ,gehabt', ,diese', ,kohle', ,eigentlich']	No Punctuation, Lower Casing
[,das', ,wir', ,man', ,nicht', ,ein', ,eine', ,mit', ,mal', ,den', ,ich', ,dort', ,als', ,bergbau', ,eigentlich', ,konnte', ,aus', ,die', ,der', ,später', ,wenn', ,hatte', ,steiger', ,zeche', ,diese', ,immer', ,einen', ,mark', ,einem', ,hast', ,wieder', ,dem', ,zur', ,mich', ,uns', ,kohle', ,damals', ,selbst', ,sach', ,wurde', ,von', ,habe', ,dieser', ,usw', ,unter', ,auf', ,gesacht', ,menschen', ,etwas', ,wurden', ,musste']	No Punctuation, Lower Casing, Mindestwortlänge drei Buchstaben
[,steiger', ,zeche', ,bergbau', ,kohle', ,son', ,de', ,dort', ,schicht', ,tage', ,naja', ,arbeiten', ,meter', ,bergleute', ,bergmann', ,betriebsrat', ,kumpel', ,usw', ,hauer', ,unter', ,kumpels', ,dabei', ,bekamen', ,ah', ,beispiel', ,manchmal', ,betriebsführer', ,franzosen', ,meinetwegen', ,bekam', ,hast', ,streb', ,nö', ,wagen', ,gewerkschaft', ,2', ,verdienen', ,gedinge', ,ihn', ,schacht', ,welche', ,kohlen', ,leistung', ,bruder', ,schachtanlage', ,robert', ,kommunisten', ,verdient', ,stempel', ,menschen', ,gearbeitet']	No Punctuation, Lower Casing, Stopword Removal by Threshold (<0,05%)
[,steiger', ,bergbau', ,zeche', ,kohle', ,schicht', ,kumpel', ,meter', ,bergleute', ,bekamen', ,hauer', ,bergmann', ,kumpels', ,betriebsrat', ,kohlen', ,menschen', ,walsum', ,betriebsführer', ,gedinge', ,geld', ,schachtanlage', ,wagen', ,lager', ,verdienen', ,robert', ,verdient', ,heimat', ,stempel', ,gefangenschaft', ,lohn', ,knappschaft', ,nu', ,schlesien', ,schacht', ,vertriebene', ,tonnen', ,zechen', ,leistung', ,ruhrgebiet', ,ruhrkohle', ,vertriebenen', ,nachtschicht', ,revier', ,pütt', ,bergschule', ,kameradschaftsgedinge', ,schichten', ,arbeiter', ,helmut', ,kollegen', ,ernst']	No Punctuation, Lower Casing, Stopword Removal by Curated Stoplist

Tabelle B: Kondensierung eines Topics zum Thema Bergbau in der Sammlung LUSIR während des Preprocessings, jeweils die ersten 50 Keywords. LDA, MALLET via Gensim-Wrapper, 20 Topics.

Bei der konkreten Anwendung von Topic Modeling kommt bei LDA als nächste Hürde die Wahl der optimalen Anzahl von Topics, also die Festlegung darauf, wie viele – thematisch möglichst kohärente – Wortlisten das Programm ausgeben soll. ^[18] Man kann Modelle mit zehn bis hunderten Topics trainieren. Multipliziert man die Ergebnisse mit den variablen Preprocessing-Schritten, steht man allerdings vor einer manuell kaum zu bewältigenden Anzahl von Wortlisten, deren Qualität beurteilt werden muss. Daher lohnt es sich, auf etablierte Evaluationsmetriken zurückzugreifen. Es gibt verschiedene Scores, die mathematisch berechnen, wann ein Modell "gut" ist – etwa, wenn eine besonders hohe Konsistenz der Topics (also Wortlisten) vorliegt oder diese besonders trennscharf differenziert werden können. Weit verbreitet und in die gängigen Bibliotheken und Pipelines integriert sind etwa die Scores Perplexity (Blei, Ng und Jordan 2003, 1008) und Coherence (Rosner u. a. 2014). Diese Scores sind rein mathematisch und immer wieder Gegenstand kritischer Diskussion (vgl. Chang u. a. 2009; Hoyle u. a. 2021). Deshalb bietet es sich an, verschiedene Scores berechnen zu lassen und zu vergleichen, ob die Tendenzen der Scores während der Schätzung des Modells gleichläufig sind und ob sich die besten Werte an einer Stelle überschneiden. Anschließend unterzieht man die Ergebnisse der auf diese Weise ermittelten Topic-Anzahl und deren Nachbarn einer qualitativen Analyse, sprich: Man nimmt drei Modelle und vergleicht diese (zum Beispiel, wenn man in Fünferschritten vorgeht: 45, 50, 55 Topics). Dieses Vorgehen kann nun iterativ mit verschiedenen Preprocessing-Konfigurationen durchlaufen werden. Damit befinden wir uns im Bereich der funktionsseitigen Säule und sollten nun die oben beschriebene systematische Ausgabe von Ergebnissen und deren Vergleich praktisch demonstrieren. Dazu kehren wir zu den angesprochenen Unterschieden zwischen den Inferenzalgorithmen zurück.

3.2.2. Funktionalität

Das vorliegende Sample lebensgeschichtlicher Interviews aus dem Oral-History-Pionierprojekt *Lebensgeschichte und Sozialkultur im Ruhrgebiet* wurde im Preprocessing folgenden Schritten unterzogen:

- Satzzeichen wurden entfernt,
- es wurde auf Kleinschreibung umgestellt,
- eine Stopwortbereinigung (*stopword removal*) wurde durchgeführt,

- es wurde lemmatisiert und alle Wortgruppen außer Nomen, Eigennamen, Adjektive, Verben und Adverbien herausgefiltert,
- außerdem wurden Zahlen und Wörter mit weniger als drei Buchstaben entfernt.

Da die Interviews zu umfangreich sind, wurden diese in kleinere Abschnitte (Chunks) aufgeteilt. Im letzten Schritt wurden Dataframes erstellt und die Chunks der Interviews als Listen von Strings (Tokenisierung) gespeichert. Es gibt zum Vergleich Korpora auf Grundlage von Chunks mit jeweils einem Satz, fünf Sätzen, zehn Sätzen, 25 und 50 Sätzen (ausführlich beschrieben in Hodel, Möbus und Serif 2022).

Das Thema der dezidierten Evaluation von Inferenzalgorithmen insbesondere im Hinblick auf geisteswissenschaftliche Forschungskontexte ist aktuell noch ein Desiderat. Der Vergleich der Topic-Modelle bietet daher zunächst exemplarisch erste Einsichten. Gehen wir davon aus, dass Mallet den Standard setzt und ziehen das am meisten konsistente Modell – also dasjenige, das am ehesten den Anschein semantischer Zusammenhänge erweckt – mit 50 Topics heran, das auf dem Korpus mit Chunks à 25 Sätzen trainiert wurde. Da die Algorithmen unterschiedlich arbeiten und die Topic-Nummern nach dem Zufallsprinzip vergeben werden, stimmt die thematische Reihenfolge der Topics in den beiden Modellen nicht überein. Mit ein paar Zeilen Code, können Topics verschiedener Topic-Modelle hinsichtlich Überschneidungen verglichen werden (Matcher), das ermöglicht eine thematische Gegenüberstellung. In Tabelle C wird deutlich, wie unterschiedlich Gibbs-Sampling und Variational Bayes trotz einer völlig identischen Ausgangslage arbeiten. Die über die Mallet-Referenz ausgemachten Themen “Krieg”, “Essen” (als gutes Beispiel einer Disambiguierung – hier handelt es sich um Nahrungsmittel und nicht die im Korpus häufig genannte Stadt Essen), “Bergbau” und “Judenverfolgung” finden sich im Gensim-Modell nicht in einer vergleichbar kohärenten Form wieder. Die Topics sind weniger thematisch konsistent und weisen eine höhere Zahl an irrelevanten oder abweichenden Termen auf (Noise), wodurch die Topics sich weniger trennscharf interpretieren lassen.

T-Nr	Mallet, 50 Topics, Chunks à 25 Sätze	T-Nr	Gensim, 50 Topics, Chunks à 25 Sätze
	Krieg		Krieg
26	krieg, gedenken, angst, gott, passieren, reden, wussten, schwer, menschen, mensch, schlecht, lieb, arbeit, ehrlich, froh	39	krieg, essen, schwer, kirche, hattingen, arbeit, partei, schlecht, zug, deutsch, juden, kinder, raus, gott, wählen
28	krieg, einziehen, arbeitsdienst, soldat, bruder, krieges, weltkrieg, freiwillig, mitmachen, gefangenschaft, entlassen, zurückkommen, urlaub, russland, kriege	3	essen, krieg, deutsch, kind, bruder, rauchen, arbeit, soldaten, schlecht, schwester, kontakt, morgens, zigaretten, schule, abends
39	krieg, erinnern, schlecht, zeiten, arbeitslos, 50er, erleben, menschen, ruhrgebiet, fünfziger, nachkriegszeit, arbeiter, verändern, normal, sachen	7	krieg, schwester, kinder, schlecht, familie, schule, angestellt, arbeiter, bauen, kind, schwer, erinnern, bruder, katholisch, kindern
	Essen: Disambiguierung		Essen: Disambiguierung
8	essen, brot, kartoffeln, garen, bauern, pfund, butter, land, hamstern, backen, kinder, lebensmittel, milch, hunger, bauer	25	essen, arbeit, schule, butter, menschen, mensch, lieb, schwer, erinnern, zigaretten, abteilung, urlaub, betrieb, milch, kollegen
	Bergbau		Bergbau
7	zechen, bergbau, steiger, kohle, arbeit, verdienen, kumpel, bergmann, schicht, bergleute, betriebsführer, hauer, schacht, schachtanlage, kumpels	49	essen, krieg, kinder, wohnen, krupp, erinnern, mädchen, schule, heiraten, arbeit, kind, lernen, wagen, deutsch, lager
	Judenverfolgung		Judenverfolgung
9	juden, berlin, jude, gewusst, jüdisch, jüdische, münchen, mitkriegen, deutsch, krieg, hitler, kristallnacht, amerika, wussten, politisch	39	krieg, essen, schwer, kirche, hattingen, arbeit, partei, schlecht, zug, deutsch, juden, kinder, raus, gott, wählen

Tabelle C: Gegenüberstellung exemplarischer Ergebnisse aus der Topic-Modellierung mit Mallet und Gensim unter gleichen Vorbedingungen.

In der Gensim-Evaluation war das Modell auf Grundlage des Korpus mit Chunks à 10 Sätzen an einigen Stellen graduell besser als dasjenige mit Chunks à 25 Sätzen. Doch auch die quere Gegenüberstellung der jeweils besten Modelle unter divergierenden Vorbedingungen relativiert den gravierenden Unterschied zwischen Gensim und Mallet nicht (siehe Tabelle D).

T-Nr	Mallet, Chunks à <u>25 Sätze</u>	T-Nr	Gensim, Chunks à <u>10 Sätze</u>
	Krieg		Krieg
26	krieg, gedenken, angst, gott, passieren, reden, wussten, schwer, menschen, mensch, schlecht, lieb, arbeit, ehrlich, froh	2	krieg, essen, bunker, gott, helfen, passieren, versuchen, mensch, lieb, schlecht, leuten, kontakt, arbeit, deutsch, chef
28	krieg, einziehen, arbeitsdienst, soldat, bruder, krieges, weltkrieg, freiwillig, mitmachen, gefangenschaft, entlassen, zurückkommen, urlaub, russland, kriege	6	krieg, schule, erinnern, kind, deutsch, gebären, erleben, kinder, einziehen, amerikaner, mitmachen, schwester, lernen, besuchen, russland
39	krieg, erinnern, schlecht, zeiten, arbeitslos, 50er, erleben, menschen, ruhrgebiet, fünfziger, nachkriegszeit, arbeiter, verändern, normal, sachen	39	krieg, erinnern, kinder, moment, kontakt, bezahlen, wohnung, mädchen, gedenken, gott, arbeit, heiraten, eltern, verdienen, wohnen
	Essen: Disambiguierung		Essen: Disambiguierung
8	essen, brot, kartoffeln, garen, bauern, pfund, butter, land, hamstern, backen, kinder, lebensmittel, milch, hunger, bauer	8	butter, essen, sachen, pfund, wohnen, fleisch, arbeit, schwer, mensch, meinung, abends, lernen, deutsch, leuten, fahrrad
	Bergbau		Bergbau
7	zechen, bergbau, steiger, kohle, arbeit, verdienen, kumpel, bergmann, schicht, bergleute, betriebsführer, hauer, schacht, schachanlage, kumpels	49	steiger, krieg, kohle, verdienen, kumpel, schicht, meter, bergbau, arbeit, erinnern, zechen, kohlen, arzt, kranken, morgens
	Judenverfolgung		Judenverfolgung
9	juden, berlin, jude, gewusst, jüdisch, jüdische, münchen, mitkriegen, deutsch, krieg, hitler, kristallnacht, amerika, wussten, politisch	30	essen, haushalt, mädchen, eltern, schwer, wohnen, wohnung, verdienen, juden, mensch, menschen, kind, zweit, schwiegermutter, gedenken

Tabelle D: Gegenüberstellung exemplarischer Ergebnisse aus der Topic-Modellierung mit Mallet und Gensim unter anderen Vorbedingungen.

3.2.3. Gesellschaftlich-lebensweltliche Auswirkungen

Die deutlichen Unterschiede der Modelle waren unerwartet; eine Lektüre der Dokumentationen sowie grundlegender Aufsätze aus der Informatik kann für solche Unterschiede allerdings sensibilisieren und über die Verfügbarkeit von Parametern zur Optimierung des jeweiligen Algorithmus aufklären. Es wäre interessant zu evaluieren, ob die Ergebnisse bei Daten, die sprachlich homogener und weniger komplex sind als Transkripte lebensgeschichtlicher Interviews, ähnlich deutlich ausfallen. Auch die Technik des Chunking, um die Interviews feingranularer untersuchen und anreichern zu können, kann weiter verfeinert und evaluiert werden. So könnten etwa die Schnittstellen, an denen die Interviews in Chunks geteilt werden, satzweise verschoben werden, wenn in ersten Modellen Topics über die Chunkgrenze hinaus dominant sind. Das könnte die Schnitte weniger arbiträr machen und in folgenden Modellen die semantische Kohärenz der Topics steigern. Der Hintergrund für die Implementierung von Variational Bayes im LDA-Algorithmus der Gensim-Bibliothek liegt darin, dass er deutlich schneller konvergiert und somit potenziell weniger Rechenzeit in Anspruch nimmt. Hier werden die Implikationen der Verwendung von Algorithmen auf der gesellschaftlich-lebensweltlichen Ebene deutlich, da größere Rechenressourcen in diesem konkreten Beispiel bessere Ergebnisse bedeuten. Das hier behandelte Sample ist mit knapp dreieinhalb Millionen Tokens für

Machine-Learning-Verhältnisse sehr klein. Doch auch in den Geisteswissenschaften wachsen Datenbestände rapide an und werden perspektivisch größere Investitionen in Recheninfrastrukturen erforderlich machen. Da, wo diese nicht verfügbar sind, müssen im Zweifel Einbußen bei der Qualität maschinell berechneter Ergebnisse in Kauf genommen werden. Auf Ebene der Daten befördert das Verzerrungen und möglicherweise Verluste bei deren Erschließung und Analyse. Auf wissenschaftspolitischer (und damit repräsentativ für die gesellschaftliche) Ebene werden auf diese Weise digitale Ungleichheitsregime etabliert.

3.2.4. Resümee und Ausblick

Nach der Entscheidung, ein exploratives Verfahren aus dem Bereich des Machine Learning zur inhaltlichen Erschließung eines unstrukturierten Korpus historischer Daten anzuwenden, hat ein Blick in die Fachliteratur zum Verständnis und einer Aufmerksamkeit für die Einflüsse vieler subtiler Faktoren auf die Ergebnisse von Topic Modeling geführt. Das Studieren von Dokumentationen und Einführungen in die Praxis des Topic Modeling hat über die Eigenheiten verschiedener Implementierungen (Python und Gensim, Java und Mallet) aufgeklärt. Daher sollte deren Lektüre obligatorischer Teil des Forschungsdesigns in einem Projekt sein, das mit Algorithmen arbeitet. Ein *Close Reading* des Codes hingegen ist nicht zwangsläufig erforderlich, da es für Fachfremde überfordernd und abschreckend sein oder sogar zu Fehlinterpretationen führen kann. Das hier bevorzugte und im Forschungsdesign skizzierte iterative Vorgehen, systematisch Ergebnisse unter veränderten Bedingungen zu produzieren, hat gravierende Unterschiede zwischen den Ergebnissen der Implementierungen in Python und Java hervortreten lassen, die letztlich auf unterschiedliche Inferenzalgorithmen zurückzuführen sind. Diese muss man selbst nicht formal verstehen, ein Bewusstsein für deren Existenz und Auswirkung ist jedoch zentral für eine seriöse Anwendung des Topic Modeling in den Geisteswissenschaften. Darüber hinaus verweist die Verwendung ressourcenschonenderer Algorithmen auf das Phänomen von Ungleichheitsregimen in der Digitalisierung, insbesondere, wenn man global denkt – denn nicht alle Forschungszusammenhänge können sich endlose Rechenressourcen leisten. Topic Modeling ist ein von vielfältigen Parametern abhängiges Verfahren, wobei diese einen spezifischen Einfluss auf die algorithmisch generierten Ergebnisse haben. Mit den Inferenzalgorithmen ist hier nur ein – wenngleich äußerst wichtiger – Faktor des Topic Modeling in den Blick genommen worden. Weitere einflussreiche Parameter neben den angesprochenen Schritten des Preprocessing sind die Iterationen, die das Training eines Modells benötigt. Hier gilt es zu bestimmen, wie viele Durchläufe für ein möglichst konsistentes Topic-Modell notwendig sind und ab wann unter Berücksichtigung von Ressourcen wie Zeit und Rechenleistung keine nennenswerte Qualitätssteigerung mehr zu erwarten ist. Auch hier bietet sich analog zur Bestimmung der optimalen Topic-Anzahl wieder die Methode des Vergleichs verschiedener Modelle an. Zur Transparenz und Reproduzierbarkeit im Sinne wissenschaftlichen Arbeitens können *Random Seeds* als “Startpunkte” für die an sich zufällig initialisierten Topic-Modeling-Algorithmen fixiert werden. Dadurch werden die Modelle bei identischer Datenvorbereitung und Parameterkonfiguration zwar reproduzierbar, jeder Seed bringt allerdings auch voneinander abweichende Modelle hervor. Und nicht zuletzt die Hyperparameter *Alpha* und *Beta* beeinflussen die konkrete Zusammensetzung der Wortlisten sowie die Gewichtung für die einzelnen Dokumente des Korpus. Gebrauchsfertige Werkzeuge und Standardeinstellungen verbergen diese Tragweite, die im Rahmen der Algorithmenkritik für den jeweiligen Anwendungsfall zu beurteilen ist.

3.3. Algorithmen in der Netzwerkforschung

3.3.1. Forschungsdesign

In der Netzwerkforschung nach der klassischen Definition von Freeman werden die (klassisch: sozialen) Beziehungen zwischen Akteuren auf Basis empirischer Daten als Netzwerk modelliert, gegebenenfalls graphisch repräsentiert und mithilfe mathematisch-formaler Modelle ausgewertet (Freeman 2004, 3). Ein Netzwerk ist dabei ein semantisch annotierter Graph, welcher aus einer festgelegten Anzahl von Verbindungen (Kanten) zwischen einer definierten Anzahl von Entitäten (Knoten) besteht (Abb. 1). Diese Verbindungen dienen zur Interpretation des (sozialen) Verhaltens der Entitäten bzw. Personen (Mitchell 1969, 2). Die Netzwerkanalyse ermöglicht dabei den Fokus auf die relationalen Eigenschaften von Entitäten und ihre Verbindungen: Zum einen erzeugen die Relationen und Interdependenzen zwischen den Entitäten überpersönliche Strukturen (und Muster) im Netzwerk (sog. emergente Netzwerkstrukturen), zum anderen gibt das hieraus resultierende Netzwerk einen gewissen Handlungsspielraum für die derart eingebundenen Entitäten bzw. schränkt diesen ein (Wasserman und Faust 1994, 4). Diese Strukturen lassen sich somit z. B. hinsichtlich sozio-ökonomischer und -politischer Gesichtspunkte, aber auch hinsichtlich anderer relationaler Fragestellungen analysieren (Wetherell 1998, 126).

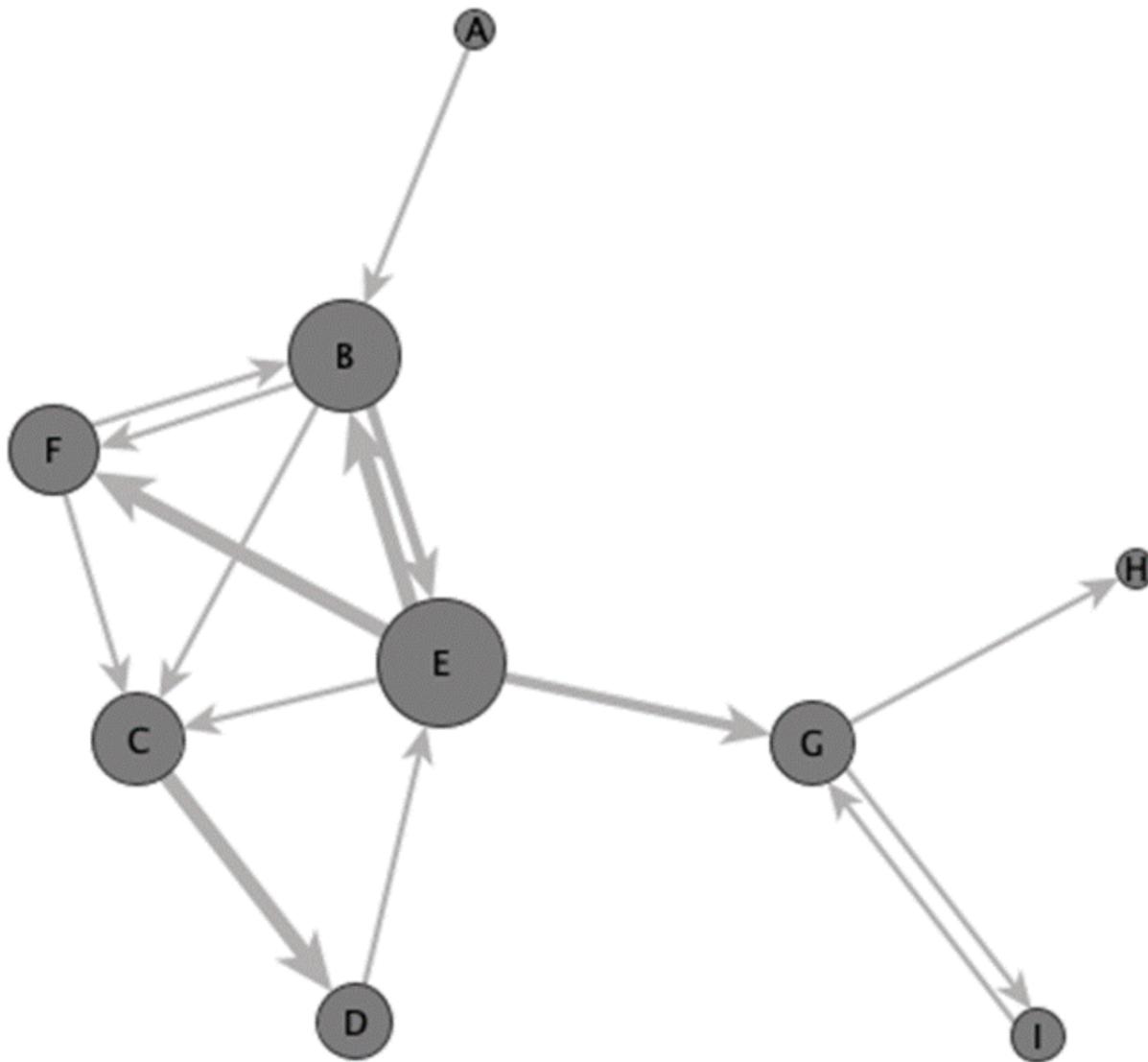


Abb. 1: Schematisches gerichtetes Beispielnetzwerk bestehend aus 9 Knoten. Dieses Netzwerk könnte z. B. ein Briefnetzwerk modellieren, bei dem die Knoten Personen darstellen, die Briefe senden oder empfangen. Das Gewicht der Kanten, hier visualisiert über die Dicke der Pfeile, richtet sich nach der Häufigkeit des Briefwechsels. Die Größe der Knoten repräsentiert die degree centrality, welche sich aus der Anzahl der empfangenen Briefe (in-degree) und gesendeten Briefe (out-degree) zusammensetzt. Damit stellt die degree centrality ein Maß für die Bedeutung der Person im Kommunikationsnetzwerk dar. Visualisierung erstellt in Visone mit Spring-embedded Layout.

Seit der Entwicklung der netzwerkanalytischen Methode(n) in der soziologischen Forschung fand diese zunehmend Eingang in anderen Forschungstraditionen: so wurden ab den 1990er Jahren ebenfalls historische Forschungsgegenstände als Netzwerke analysiert. Seit den späten 2000er Jahren fand die Netzwerkanalyse systematische Anwendung in den Geistes- und Kulturwissenschaften: So bildete sich eine Tradition der Netzwerkforschung in den historischen^[19] und archäologischen Wissenschaften^[20] heraus.

Die Forschungsgegenstände dieser Disziplinen erfordern dabei Anpassungen und neue theoretische wie methodische Ansätze, um den spezifischen Anforderungen des Quellenmaterials und der Fragen, die an dieses gestellt werden, gerecht zu werden. So beschäftigt sich die geisteswissenschaftliche Netzwerkforschung im weiteren Sinne nicht nur mit Personen, sondern auch mit Objekten, Konzepten oder Orten als Knoten sowie mit variablen und multimodalen Beziehungstypen (jenseits der "sozialen" Perspektive) als Kanten. "Zeit" als Grundbedingung und Analyseeinheit nimmt angesichts von Forschungsfragen, die gesellschaftliche Dynamiken und Entwicklungsprozesse ins Zentrum der Betrachtung stellen, eine prominente Rolle ein. Gleichzeitig werfen jedoch temporale Analysen angesichts ungenauer oder divergierender chronologischer Systeme sowie sogenannte *fuzzy dating* neue Herausforderungen auf (vgl. Lemerrier 2015). Nicht zuletzt verlangen komplexe historische Phänomene nach ebenso komplexen methodischen Herangehensweisen – so spielen beispielsweise bi- und multimodale Graphen und entsprechende Analyseverfahren eine bedeutende Rolle, wenn Netzwerke auf der Basis von Schriftquellen (z. B. Brief – Autor, Regest – Erwähnung) oder materiellen Assemblagen (Grab – Beigabe, Objekt – Dekorationselement) konstruiert werden. Netzwerke sind Modelle, die auf der Basis der Relationen und Verbindungen zwischen Entitäten operationalisiert werden. Als solche sind sie immer eine Annäherung an einen Forschungsgegenstand (vgl. dazu Drucker 2011; Petz 2022) und fokussieren auf ausgewählte Parameter, welche den Forschungsgegenstand angemessen abbilden sollen^[21] – deren Auswahl, folgt man der Logik der Modellbildung nach Stachowiak (Stachowiak 1973), subjektiv bleiben muss. Dieser Approximationscharakter von Netzwerken führt dazu, dass jedes Netzwerk nur eine Iteration einer unendlichen Anzahl an möglichen Netzwerken ist (Rehbein 2020, 265). Andere Fragen an bzw. andere Perspektiven auf das Material führen zur Konstruktion anderer Netzwerkmodelle. Deren Unterschiede können dabei sowohl in der Auswahl jener Entitäten liegen, die als Knoten konzeptualisiert

werden, als auch in der Art der Beziehung, durch welche Konnektivität zwischen diesen Knoten hergestellt wird, oder in den grundlegenden Eigenschaften des Graphen selbst (so etwa ob dessen Kanten gewichtet oder ungewichtet, gerichtet oder ungerichtet, uni- oder multimodal sind).

Auf der Basis dieser Netzwerkmodelle operieren Algorithmen zur Analyse ebenjener Netzwerke, welche wiederum eigene Funktions- und Aufbauweisen und entsprechend Modellcharakter haben. Das Angebot an verfügbaren Algorithmen zur Netzwerkanalyse muss nicht zwangsläufig zu den zu untersuchenden Netzwerken passen. Auch hier gilt: Gelten die Vorannahmen der Analyseverfahren (des Algorithmus) nicht für den zu untersuchenden Forschungsgegenstand bzw. die verwendete Operationalisierung des Netzwerks, so ist eine Kritik arbiträr.

Vor diesem Hintergrund ist eine sorgfältige Reflexion des gesamten netzwerkanalytischen und -theoretischen Forschungsprozesses unabdingbar, um *biases* des Datenbestandes zu erkennen und ihre Auswirkungen auf die quantitative Auswertung zu evaluieren und kontextualisieren zu können. Um die Abwägungen, die dabei nötig sind, zu verdeutlichen, konzentriert sich dieser Beitrag auf das Stadium der tatsächlichen Analyse und hierbei auf eine Gruppe von Algorithmen, die als Zentralitätsmetriken bekannt sind. Da sie standardmäßig in den meisten Netzwerkanalysesoftwares integriert sind, stellen sie die wohl bekanntesten Netzwerkmaße und oft einen ersten Faszinations- und Einstiegspunkt in die Netzwerkforschung dar.

3.3.2. Funktionalität

Anhand von Zentralitätsmetriken lässt sich der kritische Zugang zu Algorithmen in der Netzwerkforschung exemplarisch darstellen.

Die verschiedenen Zentralitätsmetriken operationalisieren in Algorithmen, wie zentral bzw. "wichtig" ein Knoten innerhalb eines Netzwerkes ist. Was die Wichtigkeit eines Knoten tatsächlich ausmacht, definieren diese Metriken durchaus unterschiedlich:

Welche Kriterien, Eigenschaften und Kontaktmöglichkeiten eines Knoten stehen im Vordergrund? So zählt die *degree centrality* die absolute Häufigkeit von Verbindungen eines Knoten, die *closeness centrality* misst jene Knoten hohe Werte zu, welche die kürzesten Distanzen zu allen anderen Knoten aufweisen, während die *betweenness centrality* aufzeigt, auf wie vielen Kommunikationspfaden zwischen zwei anderen Knoten der untersuchte Knoten liegt (nach Freeman 1978). Ein kurzes Beispiel soll diesen Sachverhalt illustrieren: Das schematische gerichtete Netzwerkbeispiel in Abb. 1 lässt sich als Briefnetzwerk zwischen Personen modellieren. Dort hat der Knoten E die höchste *degree centrality*: dies zeigt an, dass diese Person mit den meisten anderen Personen Briefe ausgetauscht hat; dieser Wert kann unterteilt werden in die Anzahl der Briefe, die sie geschrieben hat (*outdegree*), und die, die von ihr empfangen wurden (*indegree*). Dies spiegelt sich in der Visualisierung des gewählten Beispiels in der dargestellten Größe der Knoten wider. Person E hat ebenfalls die höchste *betweenness centrality*, da sie zwei Regionen des Netzwerks miteinander verbindet. Person E hat somit eine (prinzipielle) Kontrollmöglichkeit über die Informationsflüsse der beiden Netzwerkkomponenten und kann eine Brückenfunktion zwischen ihnen einnehmen. Außerdem weist den höchsten Wert der *closeness centrality* ebenfalls die Person E auf, da sie besonders effizient (also über besonders kurze Distanzen) Briefe zu allen weiteren Personen im Netzwerk schreiben kann und die geringste Anzahl an Brückenkontakten dazu benötigt. Somit lässt sich die Bedeutung eines Knoten unterschiedlich operationalisieren: Soll die bloße Menge an Kontakten den Ausschlag geben, oder die Kürze der Distanz eines Knoten zu den anderen Entitäten im Netzwerk?

Anders stellt sich das Beispielnetzwerk in Abb. 2 dar, wenn die Knoten im Netzwerk als Personen operationalisiert werden, die dann mit Kanten verbunden werden, wenn sie in demselben Brief erwähnt werden (also ein Kookkurrenznetzwerk). Analog zur vorherigen Operationalisierung hat E die höchste *degree centrality*, gefolgt von B, F und C. Die höchste *betweenness centrality* haben E, B und G und nehmen somit rein formal Brücken- oder *broker*-Positionen ein. Die höchste *closeness centrality* haben E, G, C und B. Diese letzten beiden Maße haben jedoch für ein Kookkurrenznetzwerk, das nach den Häufigkeiten und Mustern gemeinsamer Erwähnungen und den Gründen dafür fragt, wenig Aussagekraft. Stattdessen würde sich ein Ansatz anbieten, der Algorithmen zum Clustering von Knoten und damit zur *community detection* anwendet.

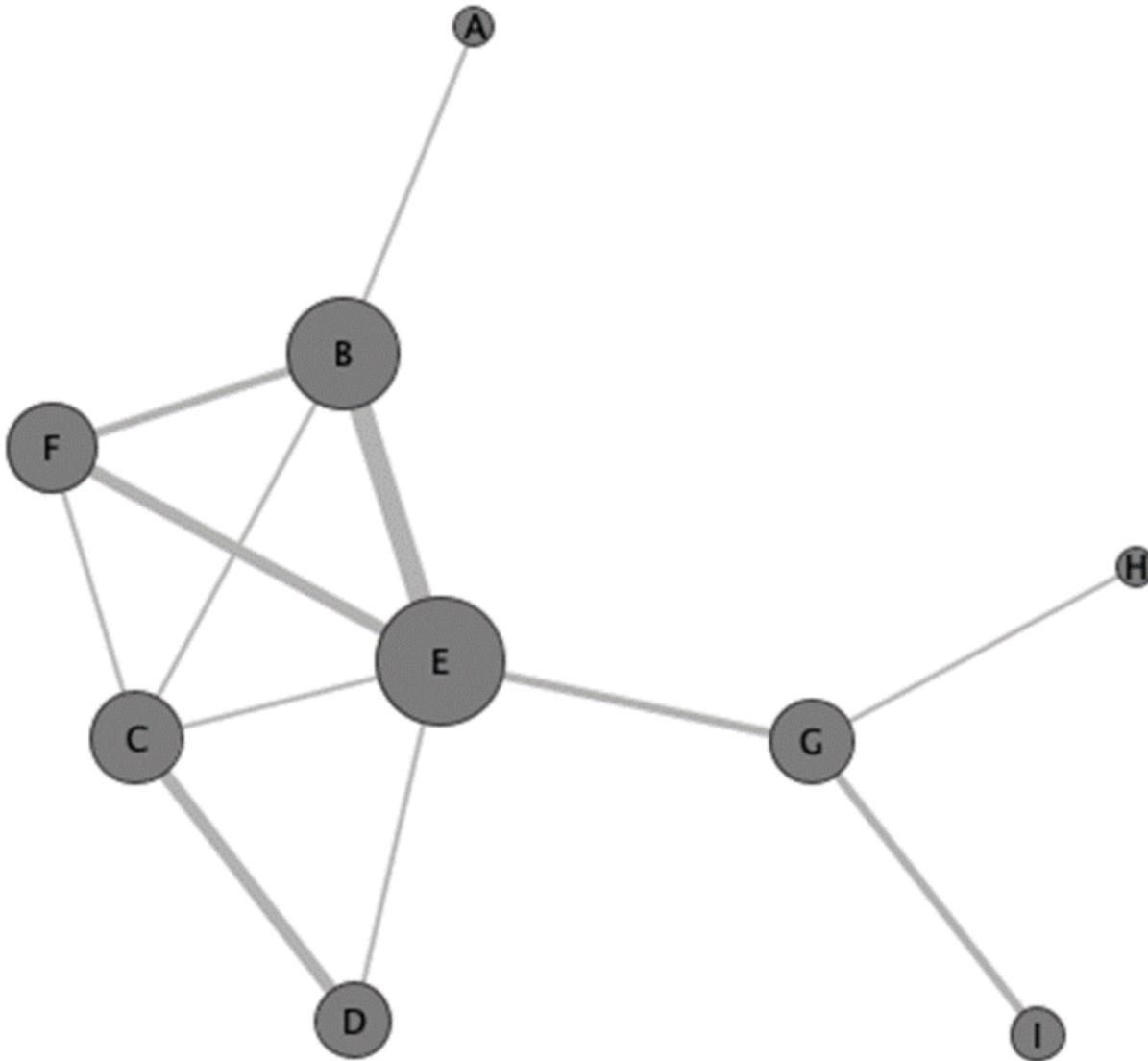


Abb. 2: Schematisches ungerichtetes Beispielnetzwerk bestehend aus 9 Knoten. Dieses Netzwerk stellt in diesem Fall ein Kookkurrenznetzwerk aus Personen dar, die in denselben Briefen gemeinsam erwähnt werden, d. h. zwischen denen eine semantische Beziehung anzunehmen ist. Das Gewicht der Kanten, hier visualisiert über deren Dicke, richtet sich nach der Häufigkeit der gemeinsamen Erwähnungen. Die Größe der Knoten repräsentiert die degree centrality, also die Gesamtanzahl der Erwähnungen mit anderen Personennamen. Damit stellt die degree centrality in diesem Netzwerk ein Maß dafür dar, wie oft über eine bestimmte Person in den untersuchten Briefen gesprochen wird. Visualisierung erstellt in Visone mit Spring-embedded Layout.

Darüber hinaus wurde eine Menge weiterer Zentralitätsmetriken entwickelt, welche etwa nicht nur die direkten Verbindungen eines Knotens, sondern auch die seiner Nachbarn messen (Eigenvektorzentralität) (Bonacich 1972), oder spezifische Forschungskontexte und gerichtete bzw. gewichtete Kanten berücksichtigen (zum Beispiel Freeman, Borgatti und White 1991; Lee, Yook und Kim 2009), deren Anzahl Koschützki u. a. als “daunting” bezeichnen (Koschützki, Lehmann, Tenfelde-Podehl, u. a. 2005, 83).

Die Verwendung von Zentralitätsmetriken hängt von der konkreten Forschungsfrage und den strukturellen Charakteristika des Datenmaterials ab, in die sich diese Frage abstrahieren lässt, sowie von der Zulässigkeit ihrer Anwendung in diesem spezifischen Forschungskontext.

Dabei basiert jedes dieser Modelle auf bestimmten Prämissen, die gerade für Nicht-Mathematiker:innen nicht immer problemlos zu erkennen sind – als solches heißt es, zu evaluieren, auf welchen spezifischen Vorannahmen die Algorithmen der Netzwerkforschung beruhen und ob diese für den spezifischen Untersuchungsgegenstand gelten können.

Unter den “klassischen” Zentralitätsmaßen nach Freeman (Freeman 1978) sticht die *betweenness centrality* heraus:

$$x_i = \sum_{st} n_{st}^i \mathbb{1}$$

Für den Knoten i ergibt sich dabei die *betweenness centrality* x_i aus der Summe der kürzesten Pfade zwischen allen übrigen Knoten eines ungerichteten Netzwerks (s und t), die über i laufen (Newman 2018, 173–174).

Eine normalisierte Variante des Algorithmus stellt die Summe der kürzesten Pfade, die über i laufen ($\sigma_{st}(i)$), der Summe aller kürzesten Pfade im Netzwerk (σ_{st}) gegenüber^[22]:

$$\delta_{st}(i) = \frac{\sigma_{st}(i)}{\sigma_{st}}$$

Dieses Zentralitätsmaß wird vor allem genutzt, um die Bedeutung des Knotens zu messen, welcher auf besonders vielen dieser kürzesten Verbindungen liegt, den sog. geodätischen Pfaden. Eine hohe *betweenness centrality* eines Knotens wird als Maß für seine Bedeutung beispielsweise im Kontext von Zugang zu oder Beschaffung von Informationen interpretiert und ist in der Regel mit sogenannten Brücken- oder *Broker*-Positionen verbunden (vgl. Freeman 1977; Burt 1999). Newman stellt dieses Maß dabei heraus, weil ein Knoten auch eine hohe *betweenness centrality* aufweisen kann, obwohl dieser sonst über vergleichsweise wenige Kanten mit dem Rest des Netzwerks verbunden ist – im Gegenteil würde sich ein Extremwert hier durch nur zwei Verbindungen ausdrücken, mit denen ein Knoten eine Brücke zwischen zwei sonst getrennten Subgraphen bilden könnte (Newman 2018, 175–176). Extremfälle kommen in *real world networks* allerdings selten vor, so dass auch das Modell, das der *betweenness centrality* zugrunde liegt, selten so deutlich den Anwendungsfall abbildet und so klar interpretiert werden kann.

Grundlegend für diesen Zentralitätsalgorithmus ist dabei das Konzept des kürzesten Pfades (Koschützki, Lehmann, Peeters, u. a. 2005, 28–29). Borgatti regt in diesem Kontext an, zu hinterfragen, was in einem Netzwerk tatsächlich gemessen wird: Was “fließt” im Netzwerk über die Verbindungen zwischen den Akteuren (= “flow”)? (Borgatti 2005) Und ist die Vorannahme dieser Metrik erfüllt, dass sich dieser “flow” auch tatsächlich effizient über die optimale Strecke (also auf dem kürzesten Pfad) zwischen zwei Akteuren bewegt?

Borgatti erstellt dazu eine Typologie von Prozessen, nach denen verschiedene Arten von “traffic”, z. B. Geld, Gerüchte, Infektionen oder auch materielle Objekte wie Pakete, durch ein Netzwerk fließen können. In der Zuordnung dieser Flussprozesse zu den vier “klassischen” Zentralitätsalgorithmen *degree*, *closeness* und *betweenness centrality* nach Freeman (Freeman 1978) sowie *Eigenvektor centrality* nach Bonacich (Bonacich 1972) kommt er zu dem Schluss, dass die meisten Prozesse nicht von diesen Algorithmen abgebildet werden können. Insbesondere die *betweenness centrality* sieht er lediglich für Transferprozesse wie z. B. Paketlieferungen als geeignet an (Borgatti 2005, 58-59 Tab. 1, 63 Tab. 2), da diese idealerweise tatsächlich optimiert, d. h. auf der kürzesten Strecke zwischen zwei Entitäten, ablaufen. Für die geisteswissenschaftliche Arbeit bedeutet dies daher zwangsläufig, dass stets sorgfältig evaluiert werden muss, inwiefern die zu messende Bedeutung eines Knotens und vor allem das dahinterstehende Konzept tatsächlich in Zusammenhang mit den Vorannahmen des Algorithmus gebracht werden können. Dies bezieht sich nicht nur auf die grundlegende Logik der Beziehung selbst, sondern auch auf Faktoren, die diese Logik beeinträchtigen und einschränken können. So ist zwar beispielsweise grundsätzlich anzunehmen, dass im Falle von Briefkommunikation Informationen bevorzugt über Personen “fließen”, die eine bedeutende Rolle im Korrespondenznetzwerk einnehmen, also auf kürzestem Wege bzw. am schnellsten Neuigkeiten erfahren; dennoch können Faktoren wie persönliche Animositäten oder auch einfach die logistischen Schwierigkeiten historischer Postbeförderung (vgl. Didczuneit 2020; Helbig 2020) Strukturen generieren, die dieser Annahme zuwiderlaufen. Gleichmaßen kann der Transport von Gütern, der tendenziell auf kürzesten Wegen vonstattengegangen sein sollte, von geopolitischen Konflikten oder Naturereignissen umgelenkt worden sein (zum Beispiel Carreras, De Soto und Muñoz 2019; Fulminante 2020). Natürlich nivellieren sich derartige Effekte mit der Zahl der Transportvorgänge und der Länge der Zeitspanne, in der sie betrachtet werden. Angesichts der oft begrenzten Quellenbasis historisch-archäologischer Forschungen ist dennoch Vorsicht geboten.

Wie bereits angedeutet, hat die Netzwerkforschung auf derartige Überlegungen mit der Entwicklung einer Vielzahl an Abwandlungen des ursprünglich von Freeman vorgeschlagenen Algorithmus geantwortet. Insbesondere existieren eine Reihe von Metriken wie *flow betweenness* oder *random-walk betweenness*, welche die direkte Abhängigkeit von geodätischen Pfaden zugunsten anderer Bewegungsmuster durch einen Graphen auflösen (Koschützki, Lehmann, Peeters, u. a. 2005, 31, 36–38, 43–45; Newman 2018, 176–177). Diese Varianten können dazu dienen, die oben beschriebenen konzeptionellen Herausforderungen aufzulösen, erfordern aber von den sie anwendenden Forschenden Kompetenzen z. B. in den Skriptsprachen R oder Python, um den Algorithmus korrekt zu berechnen, da sie in der Regel nicht oder nur vereinzelt in gängiger Netzwerkanalysesoftware wie Gephi^[23], nodegoat^[24] oder Visone^[25] integriert sind.

Darüber hinaus existieren auch Anwendungsfälle, in denen die grundsätzliche Logik der geodätischen Pfade nicht zutrifft. Dies gilt vor allem dann, wenn sich Bedeutung nicht primär aus der Kontrolle von *flow* über das gesamte Netzwerk, sondern aus der Zugehörigkeit zu Gruppen oder aus bestimmten Rollen und Positionen ergibt. Wird ein Netzwerk aus Korrespondenzdaten beispielsweise nicht aus der Sender-Empfänger-Beziehung konstruiert, sondern aus der gemeinsamen Nennung von Personen in den Brieftexten selbst, so stellt sich die Frage, was genau durch dieses Netzwerk fließt: Welche Bedeutung hat die Tatsache, dass Person A mit Person B erwähnt wird, die mit Person C erwähnt wird, die wiederum mit Person D erwähnt wird, für die Beziehung zwischen A und D (Abb. 1)? Ergibt sich ein sinnvoller Erkenntnisgewinn aus der Tatsache, dass B und C auf dem kürzesten Weg zwischen A und D liegen?

In der Regel werden in diesem Fall andere Auswertungsperspektiven, wie z. B. die Analyse von Clustern gemeinsamer Nennungen, im Zentrum der Betrachtung stehen, während die Messung von Zentralitäten und damit der Wichtigkeit einzelner Knoten nur nachrangige Bedeutung für die Erkenntnisziele der Untersuchung haben kann. Angesichts des z. T. unsicheren, lückenhaften und selektiv überlieferten bzw. zugänglichen Quellenmaterials, mit dem Historiker:innen wie Archäolog:innen in der Regel arbeiten, besteht ein nicht unerheblicher Bestandteil der kritischen Reflexion entsprechender Algorithmen in Überlegungen, inwiefern die Einschränkungen der Quellen mit der Funktionsweise der jeweiligen Metrik in Wechselwirkungen treten. Inwiefern können auf einer solchen Datenbasis überhaupt signifikante Ergebnisse entsprechender Algorithmen erzielt werden?

Sogenannte Robustheitsanalysen helfen dabei, genau jene Abweichungen der Metriken zu evaluieren, die durch beispielsweise fehlende, fehlerhaft zugeordnete oder falsche Daten entstehen könnten. Tendenziell deuten Robustheitsanalysen auf die relative Stabilität der Zentralitätsmetriken selbst bei *fuzzy* historischen Daten hin und setzen neue Standards in der Evaluation quantitativer Analysen auf diesem Gebiet. [\[26\]](#)

Um den multidimensionalen und multikausalen Faktoren Rechnung zu tragen, die auf Zeugnisse der Vergangenheit einwirken, muss jede formale Analyse von einer ausführlichen Dokumentation der *biases* des Quellenbestands wie auch der ausgeführten Analyseschritte und einer sorgfältigen Kontextualisierung der Daten begleitet werden.

Diese Quellenreflexion und -kontextualisierung ermöglicht erst die Einschätzung, ob die zu verwendenden Methoden & Analysealgorithmen die Datengrundlage angemessen greifen und somit sinnvolle Ergebnisse produzieren können, welche wiederum nuanciert interpretiert werden sollten. Erst dann kann das Potential der Netzwerkanalyse voll ausgeschöpft werden.

3.3.3. Gesellschaftlich-lebensweltliche Auswirkungen

Gesellschaftlich-lebensweltliche Auswirkungen speziell von Zentralitätsalgorithmen sind begrenzt. Zentralitätsmetriken wie die *betweenness centrality* produzieren als Ergebnis einen Wert, welcher nur relativ zu den anderen Entitäten im Netzwerk sowie zu dessen Struktur [\[27\]](#) interpretiert werden kann. Ein sehr hoher Wert der *betweenness centrality* bedeutet, dass ein Knoten auf besonders vielen (bzw. den meisten) Kommunikationspfaden im Netzwerk liegt. Die absoluten Kennzahlen sollten dabei nicht überbewertet, sondern müssen relativ zu den Werten der anderen Knoten im Netzwerk gesehen werden. So lassen sich etwa die Top 10% der zentralsten Knoten im Netzwerk identifizieren, die einen signifikanten Unterschied in der Höhe ihrer Zentralitätswerte haben. Eine absolute Grenze für vermeintlich hohe Werte in Zentralitätsmaßen gibt es jedoch unserer Einschätzung nach nicht sinnvollerweise. Die Ergebnisse der Analyse müssen somit wieder durch qualitative und kontextuelle Einordnung rückgebunden werden, um das untersuchte Phänomen entsprechend fassen zu können. Dies ermöglicht eine relationale Einordnung der mithilfe von Zentralitätsalgorithmen als zentral identifizierten Entitäten und eine Reflexion, auf welche Art und Weise diese "wichtig" für den Forschungsgegenstand sind.

Zentralitätsmetriken lassen sich dabei auch visuell darstellen, hier gilt es eine gewisse *visual literacy* zu etablieren: In Abb. 1 sind die Knoten von unterschiedlicher Größe abhängig zum Wert des gewählten Zentralitätsmaßes dargestellt, welches unreflektiert genutzt die Bedeutung eines Knoten verzerren kann. Darüber hinaus gibt die räumliche Verortung eines Knoten im Netzwerk keine Auskunft über die Bedeutung des Knotens, sondern ist abhängig vom verwendeten Layout-Algorithmus, welcher die Verteilung der Knoten im Raum nicht nach ihren Werten bzw. Attributen, sondern nach bestimmten physikalischen Mechanismen berechnet. So nutzt z. B. der in diesem Artikel verwendete *spring embedder*-Layoutalgorithmus "force-directed methods in which a network is likened to a physical system of repelling objects (the nodes) and springs of a given length (the links) binding adjacent nodes together" [\[28\]](#).

Um eine Transparenz der Methodenverwendung zu gewährleisten, ist es auch hier essentiell genaue Angaben zur verwendeten Skriptsprache bzw. Software sowie zu den Packages (in welchen wiederum der Algorithmus implementiert wurde) zu machen.

4. Schluss und Ausblick

Die angeführten Beispiele mahnen zu einer Art *algorithmic awareness* bei der Benutzung ubiquitärer digitaler Angebote. Die Herausforderungen beginnen schon bei der Auswahl der richtigen Methode. Mittlerweile gibt es zahlreiche Kanäle und Publikationen, um sich einen Überblick zu verschaffen – angefangen bei Wikipedia, über Blogs [\[29\]](#) oder DH-Periodika. [\[30\]](#) Wichtig ist, sich vor der Wahl einer digitalen Methode über Fragestellung und Erkenntnisinteresse im Klaren zu sein. Überlegungen wie: "Ich möchte mal etwas mit Graphdatenbanken machen." oder "Ich würde gerne einmal Topic Modeling ausprobieren." sind zwar nachvollziehbar, aber problematisch. Die Ergebnisse müssen sich letztlich an einer Fragestellung messen lassen, um sie beurteilen zu können. Zwar kann auch explorativ und iterativ mit digitalen Methoden umgegangen werden, jedoch müssen dann viele Sackgassen und Nebenstrecken einkalkuliert werden. Auch befördert dieses Vorgehen einen gewissen Positivismus, weil das Erkenntnisinteresse dann gerne der Methode angepasst wird. Wir schlagen die Orientierung an drei Säulen der Algorithmenkritik – 1. Forschungsdesign, 2. Funktionalität und 3. gesellschaftlich-

lebensweltliche Auswirkungen – vor, um in den Geisteswissenschaften methodisch und reflektiert algorithmenbasiert zu arbeiten. Wichtig ist, sich vor dem Aufbau des Forschungsprojekts über Gegenstand, Fragestellung und Methode im Klaren zu sein – denn nur, wenn Algorithmus und Erkenntnisinteresse kompatibel sind, macht eine Evaluation und Kritik der Algorithmen Sinn. Diese epistemologische Prämisse geht direkt ein in das Forschungsdesign, der ersten Säule nach dem Fixieren von Forschungsgegenstand und Fragestellung. Hier kann auch im Bereich der digitalen Methoden ein Forschungsüberblick helfen, den richtigen Ansatz zu finden. Anhand bereits durchgeführter Studien kann die Spur zur richtigen Methode gefunden werden. Wenn die Entscheidung getroffen wurde, geht es einen Schritt weiter hinein in die Methode: Welche Algorithmen sind am Werk und wie kann ich deren Auswirkungen und Unterschiede verstehen? Diese Frage kann praktisch entlang der Modi der zweiten Säule der Algorithmenkritik verfolgt werden. Für Geisteswissenschaftler:innen bietet sich eine ergebnisorientierte, iterative Herangehensweise an, um die Funktionsweise von Algorithmen und die Auswirkungen von Data-Preprocessing und Parametrisierungen zu überprüfen und verstehen. Die Reflexion und Dokumentation dieser methodischen Herausforderungen hilft kommenden Generationen, mindestens eine *algorithmic awareness* – angesiedelt im Bereich des *factual knowledge innerhalb der Algorithm Literacy* –, besser noch eine generelle *Algorithm Literacy* auszubilden. Das ist Teil der dritten Säule der Algorithmenkritik, den gesellschaftlich-lebensweltlichen Auswirkungen.

//

Literatur

Althage, Melanie. 2022. Potenziale und Grenzen der Topic-Modellierung mit Latent Dirichlet Allocation für die Digital History. In: *Digital History. Konzepte, Methoden und Kritiken Digitaler Geschichtswissenschaft*, hg. von Karoline Dominika Döring, Stefan Haas, Mareike König und Jörg Wettlaufer, 255–277. Studies in Digital History and Hermeneutics 6. Berlin, Boston: De Gruyter Oldenbourg.
<https://doi.org/10.1515/9783110757101-014>.

American Library Association. 2006. Presidential Committee on Information Literacy: Final Report. Text.
<https://www.ala.org/acrl/publications/whitepapers/presidential> (zugegriffen: 15. April 2024).

- Anderson, James P. 1961. A computer for direct execution of algorithmic languages. In: *Proceedings of the December 12-14, 1961, eastern joint computer conference: computers – key to total systems control*, 184–193. AFIPS '61 (Eastern). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1460764.1460777>.
- Azad, Hiteshwar Kumar und Akshay Deepak. 2019. Query expansion techniques for information retrieval: A survey. *Information Processing & Management* 56, Nr. 5: 1698–1735. <https://doi.org/10.1016/j.ipm.2019.05.009>.
- Bächle, Thomas Christian. 2015. *Mythos Algorithmus: Die Fabrikation des computerisierbaren Menschen*. Wiesbaden: Springer VS. <https://doi.org/10.1007/978-3-658-07627-6>.
- Backus, J. W., F. L. Bauer, J. Green, C. Katz, J. McCarthy, A. J. Perlis, H. Rutishauser, K. Samelson, B. Vauquois, J. H. Wegstein, A. van Wijngaarden und M. Woodger. 1960. Report on the algorithmic language ALGOL 60. *Communications of the ACM* 3, Nr. 5: 299–314. <https://doi.org/10.1145/367236.367262>.
- Bauer, Friedrich L. 2007. *Kurze Geschichte der Informatik*. Paderborn, München: Fink.
- Blei, David M., Andrew Y. Ng und Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3: 993–1022. <https://dl.acm.org/doi/10.5555/944919.944937> (zugegriffen: 26. Juli 2024).
- Bonacich, Phillip. 1972. Factoring and weighting approaches to status scores and clique identification. *The Journal of Mathematical Sociology* 2, Nr. 1: 113–120. <https://doi.org/10.1080/0022250X.1972.9989806>.
- Borgatti, Stephen P. 2005. Centrality and network flow. *Social Networks* 27, Nr. 1: 55–71. <https://doi.org/10.1016/j.socnet.2004.11.008>.
- Borgatti, Stephen P., Kathleen M. Carley und David Krackhardt. 2006. On the robustness of centrality measures under conditions of imperfect data. *Social Networks* 28, Nr. 2: 124–136. <https://doi.org/10.1016/j.socnet.2005.05.001>.
- Brandes, Ulrik. 2001. A faster algorithm for betweenness centrality. *The Journal of Mathematical Sociology* 25, Nr. 2: 163–177. <https://doi.org/10.1080/0022250X.2001.9990249>.
- Broussard, Meredith. 2018. *Artificial Unintelligence: How Computers Misunderstand the World*. Cambridge, Massachusetts: The MIT Press. <https://mitpress.mit.edu/9780262537018/artificial-unintelligence/> (zugegriffen: 15. April 2024).
- Brughmans, Tom, Anna Collar und Fiona Coward, Hrsg. 2016. *The Connected Past: Challenges to Network Studies in Archaeology and History*. Oxford, New York: Oxford University Press.
- Burt, Ronald S. 1999. The Social Capital of Opinion Leaders. *The Annals of the American Academy of Political and Social Science* 566: 37–54. <https://www.jstor.org/stable/1048841>.
- Cardon, Dominique. 2010. *La démocratie Internet: promesses et limites*. Paris: Seuil.
- Cardon, Dominique. 2017. Den Algorithmus dekonstruieren: Vier Typen digitaler Informationsberechnung. In: *Algorithmenkulturen: Über die rechnerische Konstruktion der Wirklichkeit*, hg. von Robert Seyfert und Jonathan Roberge, 131–150. Kulturen der Gesellschaft Band 26. Bielefeld: Transcript.
- Carreras, Cèsar, Pau De Soto und Aina Muñoz. 2019. Land transport in mountainous regions in the Roman Empire: Network analysis in the case of the Alps and Pyrenees. *Journal of Archaeological Science: Reports* 25: 280–293. <https://doi.org/10.1016/j.jasrep.2019.04.011>.
- Chang, Jonathan, Jordan Boyd-Graber, Sean Gerrish, Chong Wang und David Blei. 2009. Reading Tea Leaves: How Humans Interpret Topic Models. In: *Advances in Neural Information Processing Systems*, hg. von Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, und A. Culotta, 22. Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2009/file/f92586a25bb3145facd64ab20fd554ff-Paper.pdf> (zugegriffen: 15. April 2024).
- Church, Alonzo. 1936. An Unsolvable Problem of Elementary Number Theory. *American Journal of Mathematics* 58, Nr. 2: 345–363. <https://doi.org/10.2307/2371045>.
- Claes, Arnaud und Thibault Philippette. 2020. Defining a critical data literacy for recommender systems: A media-grounded approach. *Journal of Media Literacy Education* 12, Nr. 3: 17–29. <https://doi.org/10.23860/jmle-2020-12-3-3>.

- Collar, Anna, Hrsg. 2022. *Networks and the Spread of Ideas in the Past: Strong Ties, Innovation and Knowledge Exchange*. Digital Research in the Arts and Humanities. London: Routledge. <https://doi.org/10.4324/9780429429217>.
- Dantzig, Tobias. 2007. *Number: the language of science*. Republication of the 4th ed. The Masterpiece Science Edition. New York: Plume.
- Didczuneit, Veit. 2020. Postgeschichte. In: *Handbuch Brief: Von der Frühen Neuzeit bis zur Gegenwart*, hg. von Marie Isabel Matthews-Schlinzig, Jörg Schuster, Gesa Steinbrink und Jochen Strobel, 163–186. Berlin, Boston: De Gruyter. <https://doi.org/10.1515/9783110376531-010>.
- Dobson, James. 2021. Interpretable Outputs: Criteria for Machine Learning in the Humanities. *Digital Humanities Quarterly* 15, Nr. 2. <http://www.digitalhumanities.org/dhq/vol/15/2/000555/000555.html> (zugegriffen: 15. April 2024).
- Dogrueel, Leyla. 2021. What is Algorithm Literacy? A Conceptualization and Challenges Regarding its Empirical Measurement. In: *Algorithms and Communication*, hg. von Monika Taddicken und Christina Schumann, 67–93. Digital Communication Research 9. Berlin: Freie Universität Berlin. <https://www.ssoar.info/ssoar/handle/document/75897> (zugegriffen: 15. April 2024).
- Dogrueel, Leyla, Philipp Masur und Sven Joeckel. 2022. Development and Validation of an Algorithm Literacy Scale for Internet Users. *Communication Methods and Measures* 16, Nr. 2: 115–133. <https://doi.org/10.1080/19312458.2021.1968361>.
- Drucker, Johanna. 2011. Humanities Approaches to Graphical Display. *Digital Humanities Quarterly* 5, Nr. 1. <http://digitalhumanities.org/dhq/vol/5/1/000091/000091.html> (zugegriffen: 15. April 2024).
- Düring, Marten. 2016. How Reliable are Centrality Measures for Data Collected from Fragmentary and Heterogeneous Historical Sources? A Case Study. In: *The Connected Past: Challenges to Network Studies in Archaeology and History*, hg. von Tom Brughmans, Anna Collar und Fiona Coward, 85–101. Oxford, New York: Oxford University Press.
- Düring, Marten, Ulrich Eumann, Martin Stark und Linda von Keyserlingk-Rehbein, Hrsg. 2016. *Handbuch Historische Netzwerkforschung. Grundlagen und Anwendungen*. Berlin: LIT.
- Feuerriegel, Stefan. 2016. Software Testing. Universität Freiburg. https://www.is.uni-freiburg.de/ressourcen/algorithm-design-and-software-engineering-oeffentlicher-zugriff/11_softwaretesting.pdf (zugegriffen: 15. April 2024).
- Firth, J. R. 1957. A synopsis of linguistic theory, 1930-1955. In: *Studies in Linguistic Analysis*: 1–32. Special Volume of the Philological Society. Nachdruck v. 1962. Oxford: Blackwell.
- Freeman, Linton C. 1977. A Set of Measures of Centrality Based on Betweenness. *Sociometry* 40, Nr. 1: 35–41. <https://doi.org/10.2307/3033543>.
- . 1978. Centrality in social networks conceptual clarification. *Social Networks* 1, Nr. 3: 215–239. [https://doi.org/10.1016/0378-8733\(78\)90021-7](https://doi.org/10.1016/0378-8733(78)90021-7).
- . 2004. *The development of social network analysis: a study in the sociology of science*. Vancouver, BC: Empirical Press.
- Freeman, Linton C., Stephen P. Borgatti und Douglas R. White. 1991. Centrality in valued graphs: A measure of betweenness based on network flow. *Social Networks* 13, Nr. 2: 141–154. [https://doi.org/10.1016/0378-8733\(91\)90017-N](https://doi.org/10.1016/0378-8733(91)90017-N).
- Frege, Gottlob. 1879. *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*. Halle: Verlag von Louis Nebert. <https://gdz.sub.uni-goettingen.de/id/PPN538957069> (zugegriffen: 26. Juli 2024).
- Fulminante, Francesca. 2020. Terrestrial communication networks and political agency in Early Iron Age Central Italy (950–500 BCE): A bottom-up approach. In: *Archaeological Networks and Social Interaction*, hg. von Lieve Donnellan, 196–213. London: Routledge.
- Futschek, Gerald. 2006. Algorithmic Thinking: The Key for Understanding Computer Science. In: *Informatics Education – The Bridge between Using and Understanding Computers*, hg. von Roland T. Mittermeir, 159–168. Lecture Notes in Computer Science 4226. Berlin, Heidelberg: Springer. https://doi.org/10.1007/11915355_15.
- Gamper, Markus, Linda Reschke und Marten Düring, Hrsg. 2015. *Knoten und Kanten III: Soziale Netzwerkanalyse in Geschichts- und Politikforschung*. Bielefeld: Transcript Verlag. <https://doi.org/10.1515/9783839427422>.

- Gillespie, Tarleton. 2016. Algorithm. In: *Digital Keywords | A Vocabulary of Information Society and Culture*, hg. von Benjamin Peters, 18–30. Princeton, Oxford: Princeton University Press. <https://culturedigitally.org/wp-content/uploads/2016/07/Gillespie-2016-Algorithm-Digital-Keywords-Peters-ed.pdf> (zugegriffen: 15. April 2024).
- Griffiths, Thomas L. und Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences* 101, Nr. 1: 5228–5235. <https://doi.org/10.1073/pnas.0307752101>.
- Grootendorst, Maarten. BERTopic. <https://maartengr.github.io/BERTopic/index.html> (zugegriffen: 15. April 2024).
- Harris, Zellig S. 1954. Distributional Structure. *WORD* 10, Nr. 2–3: 146–162. <https://doi.org/10.1080/00437956.1954.11659520>.
- Hartwig, Rolf. 1996. Formale Semantik. Universität Leipzig. <http://web.archive.org/web/20091229120932/https://www.informatik.uni-leipzig.de/~rhartwig/download/FormaleSemantik.pdf> (zugegriffen: 26. Juli 2024).
- Helbig, Joachim. 2020. Historische Kommunikationslogistik: 600 Jahre Briefpostbeförderung. In: *Handbuch Brief: Von der Frühen Neuzeit bis zur Gegenwart*, hg. von Marie Isabel Matthews-Schlinzig, Jörg Schuster, Gesa Steinbrink und Jochen Strobel, 377–386. Berlin, Boston: De Gruyter. <https://doi.org/10.1515/9783110376531-024>.
- Hobohm, Hans-Christoph. 2010. DIKW Hierarchie. In: *Lexikon der Bibliotheks- und Informationswissenschaft (LBI)*, hg. von Stefan Gradmann und Konrad Umlauf, 222–223. Stuttgart.
- Hodel, Tobias, Dennis Möbus und Ina Serif. 2022. Von Inferenzen und Differenzen. Ein Vergleich von Topic-Modeling-Engines auf Grundlage historischer Korpora. In: *Von Menschen und Maschinen. Mensch-Maschine-Interaktionen in digitalen Kulturen*, hg. von Selin Gerlek, Sarah Kissler, Thorben Mämecke, und Dennis Möbus, 185–209. Hagen: Hagen University Press. https://ub-deposit.fernuni-hagen.de/receive/mir_mods_00001849 (zugegriffen: 15. April 2024).
- Hooft, Justin Johan Jozias van der, Joe Wandy, Michael P. Barrett, Karl E. V. Burgess und Simon Rogers. 2016. Topic modeling for untargeted substructure exploration in metabolomics. *Proceedings of the National Academy of Sciences* 113, Nr. 48: 13738–13743. <https://doi.org/10.1073/pnas.1608041113>.
- Horstmann, Jan. 2018. Topic Modeling. *forText. Literatur digital erforschen*. 15. Januar. <https://fortext.net/routinen/methoden/topic-modeling> (zugegriffen: 15. April 2024).
- Hoyle, Alexander, Pranav Goel, Denis Peskov, Andrew Hian-Cheong, Jordan Boyd-Graber und Philip Resnik. 2021. Is Automated Topic Model Evaluation Broken? The Incoherence of Coherence. arXiv. <https://doi.org/10.48550/arXiv.2107.02173>.
- Huang, Roger. 2018. What is Digital Literacy? A Comprehensive Guide. *code(love)*. <https://code-love.com/2018/12/12/digital-literacy/> (zugegriffen: 15. April 2024).
- Jannidis, Fotis. 2017. Grundbegriffe des Programmierens. In: *Digital Humanities. Eine Einführung*, hg. von Fotis Jannidis, Hubertus Kohle und Malte Rehbein, 68–95. Stuttgart: J. B. Metzler. https://doi.org/10.1007/978-3-476-05446-3_6.
- Kerschbaumer, Florian, Linda von Keyserlingk-Rehbein, Martin Stark und Marten Düring, Hrsg. 2020. *The Power of Networks: Prospects of Historical Network Research*. Digital Research in the Arts and Humanities. London: Routledge. <https://doi.org/10.4324/9781315189062>.
- Khuwārizmī, Muḥammad ibn Mūsá, Menso Folkerts und Paul Kunitzsch. 1997. *Die älteste lateinische Schrift über das indische Rechnen nach al-Ḥwārizmī*. Abhandlungen / Bayerische Akademie der Wissenschaften. Philosophisch-Historische Klasse n. F., Heft 113. München: Verlag der Bayerischen Akademie der Wissenschaften.
- Kitchin, Rob. 2017. Thinking critically about and researching algorithms. *Information, Communication & Society* 20, Nr. 1: 14–29. <https://doi.org/10.1080/1369118X.2016.1154087>.
- Kleene, Stephen Cole. 1935a. A Theory of Positive Integers in Formal Logic. Part I. *American Journal of Mathematics* 57, Nr. 1: 153–173. <https://doi.org/10.2307/2372027>.
- . 1935b. A Theory of Positive Integers in Formal Logic. Part II. *American Journal of Mathematics* 57, Nr. 2: 219–244. <https://doi.org/10.2307/2371199>.
- Knappett, Carl, Hrsg. 2013. *Network Analysis in Archaeology. New Approaches to Regional Interaction*. Oxford: Oxford University Press.

- Knuth, Donald. 1997. *The Art of Computer Programming. Volume 1: Fundamental Algorithms*. Boston: Addison-Wesley-Longman.
- Koch, Bernard, Emily Denton, Alex Hanna und Jacob Gates Foster. 2021. Reduced, Reused and Recycled: The Life of a Dataset in Machine Learning Research. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2). <https://openreview.net/forum?id=zNQBIBKJRkd> (zugegriffen: 15. April 2024).
- Koschützki, Dirk, Katharina Anna Lehmann, Leon Peeters, Stefan Richter, Dagmar Tenfelde-Podehl und Oliver Zlotowski. 2005. Centrality Indices. In: *Network Analysis: Methodological Foundations*, hg. von Ulrik Brandes und Thomas Erlebach, 16–61. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-31955-9_3.
- Koschützki, Dirk, Katharina Anna Lehmann, Dagmar Tenfelde-Podehl und Oliver Zlotowski. 2005. Advanced Centrality Concepts. In: *Network Analysis: Methodological Foundations*, hg. von Ulrik Brandes und Thomas Erlebach, 83–111. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-31955-9_5.
- Kunilovskaya, Maria und Alistair Plum. 2021. Text Preprocessing and its Implications in a Digital Humanities Project. In: *Proceedings of the Student Research Workshop Associated with RANLP 2021*, 85–93. INCOMA Ltd. <https://aclanthology.org/2021.ranlp-srw.13> (zugegriffen: 15. April 2024).
- Kuzi, Saar, Anna Shtok und Oren Kurland. 2016. Query Expansion Using Word Embeddings. In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 1929–1932. CIKM '16. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2983323.2983876>.
- Laboon, Bill. 2016. *A Friendly Introduction to Software Testing*. North Charleston, SC, USA: CreateSpace Independent Publishing Platform.
- Lee, S., S.-H. Yook und Y. Kim. 2009. Centrality measure of complex networks using biased random walks. *The European Physical Journal B* 68, Nr. 2: 277–281. <https://doi.org/10.1140/epjb/e2009-00095-5>.
- Lemercier, Claire. 2015. Taking time seriously: How do we deal with change in historical networks? In: *Knoten und Kanten III: Soziale Netzwerkanalyse in Geschichts- und Politikforschung*, hg. von Markus Gamper, Linda Reschke und Marten Düring, 183–212. Bielefeld: Transcript Verlag. <https://doi.org/10.1515/9783839427422-006>.
- Lewandowski, Dirk. 2021. *Suchmaschinen verstehen*. 3. Aufl. Berlin, Heidelberg: Springer. <https://doi.org/10.1007/978-3-662-63191-1>.
- Luhmann, Jan und Manuel Burghardt. 2021. Digital humanities—A discipline in its own right? An analysis of the role and position of digital humanities in the academic landscape. *Journal of the Association for Information Science and Technology* 73, Nr. 2: 148–171. <https://doi.org/10.1002/asi.24533>.
- Marino, Mark C. 2006. Critical Code Studies. *Electronic Book Review*. <https://electronicbookreview.com/essay/critical-code-studies/> (zugegriffen: 26. Juli 2024).
- . 2020. *Critical Code Studies*. Software Studies. Cambridge, Massachusetts; London: The MIT Press.
- Markkula, Marjo und Eero Sormunen. 2000. End-User Searching Challenges Indexing Practices in the Digital Newspaper Photo Archive. *Information Retrieval* 1, Nr. 4: 259–285. <https://doi.org/10.1023/A:1009995816485>.
- Markov, Andrey. 1954. *Theory of Algorithms*. Moskau.
- McCallum, Andrew Kachites. 2002. MALLET: A Machine Learning for Language Toolkit. <http://mallet.cs.umass.edu> (zugegriffen: 15. April 2024).
- Mehrabi, Ninareh, Fred Morstatter, Nripsuta Saxena, Kristina Lerman und Aram Galstyan. 2021. A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys* 54, Nr. 6: 1–35. <https://doi.org/10.1145/3457607>.
- Minsky, Marvin. 1967. *Computation: finite and infinite machines*. Englewood Cliffs, New Jersey: Prentice-Hall. <https://archive.org/details/computationfinit0000mins> (zugegriffen: 15. April 2024).

- Mitchell, J. Clyde. 1969. The Concept and Use of Social Networks. In: *Social Networks in Urban Situations: Analyses of Personal Relationships in Central African Towns*, hg. von J. Clyde Mitchell, 1–50. Oxford: Published for the Institute for Social Research University of Zambia by Manchester University Press.
- Miyazaki, Shintaro. 2012. Algorhythmics: Understanding Micro-Temporality in Computational Cultures. *Computational Culture* 2. <http://computationalculture.net/algorhythmics-understanding-micro-temporality-in-computational-cultures/> (zugegriffen: 15. April 2024).
- Montfort, Nick, Patsy Baudoin, John Bell, Ian Bogost, Jeremy Douglass, Mark C. Marino, Michael Mateas, Casey Reas, Mark Sample und Noah Vawter. 2014. *10 PRINT CHR\$(205.5+RND(1)); : GOTO 10*. Cambridge, London: The MIT Press. <https://mitpress.mit.edu/9780262526746/10-print-chr205-5rnd1-goto-10/> (zugegriffen: 15. April 2024).
- Nakade, Vidya, Aibek Musaev und Travis Atkison. 2018. Preliminary research on thesaurus-based query expansion for Twitter data extraction. In: *Proceedings of the ACMSE 2018 Conference*, 1–4. ACMSE '18. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3190645.3190694>.
- Newman, Mark. 2018. *Networks*. 2. Aufl. Oxford: Oxford University Press. <https://doi.org/10.1093/oso/9780198805090.001.0001>.
- Niethammer, Lutz, Hrsg. 1983. „Die Jahre weiss man nicht, wo man die heute hinsetzen soll“: *Faschismuserfahrungen im Ruhrgebiet: Lebensgeschichte und Sozialkultur im Ruhrgebiet 1930 bis 1960*. Berlin: Dietz.
- Noble, Safiya Umoja. 2018. *Algorithms of Oppression: How Search Engines Reinforce Racism*. New York: New York University Press. <https://doi.org/10.2307/j.ctt1pwt9w5>.
- O A. Deutsches Textarchiv. <https://www.deutschestextarchiv.de/> (zugegriffen: 15. April 2024).
- O A. FernUniversität in Hagen: Archiv „Deutsches Gedächtnis“. <https://www.fernuni-hagen.de/geschichteundbiographie/deutschesgedaechtnis/index.shtml> (zugegriffen: 15. April 2024).
- O A. Gensim: topic modelling for humans. https://radimrehurek.com/gensim/auto_examples/index.html (zugegriffen: 19. März 2024).
- O A. Gensim: Latent Dirichlet Allocation. <https://radimrehurek.com/gensim/models/ldamodel.html> (zugegriffen: 19. März 2024).
- O A. Gephi – The Open Graph Viz Platform. <https://gephi.org/> (zugegriffen: 19. März 2024).
- O A. Network Robustness Tool. https://networkingarchives.shinyapps.io/network_robustness_tool/ (zugegriffen: 15. April 2024).
- O A. nodegoat. nodegoat. <https://nodegoat.net/> (zugegriffen: 19. März 2024).
- O A. nodegoat. Documentation. Create Graph. <https://nodegoat.net/documentation.s/86/create-graph> (zugegriffen: 15. April 2024).
- O A. Programming Historian. <http://programminghistorian.org/> (zugegriffen: 19. März 2024).
- O A. visone. Visual Social Networks. <https://visone.ethz.ch/> (zugegriffen: 19. März 2024).
- O A. visone manual. Betweenness. <https://visone.ethz.ch/wiki/index.php/Betweenness> (zugegriffen: 15. April 2024).
- O A. visone manual. Visualization tab: spring embedder. https://visone.ethz.ch/wiki/index.php/Visualization_tab#spring_embedder (zugegriffen: 15. April 2024).
- O A. ZfdG – Zeitschrift für digitale Geisteswissenschaften. <https://zfdg.de/> (zugegriffen: 19. März 2024).
- Passig, Kathrin. 2017. Fünfzig Jahre Black Box. *Merkur* 823: 16–30.
- Pegrum, Mark, Nicky Hockly und Gavin Dudeney. 2022. *Digital Literacies*. 2. Aufl. New York: Routledge. <https://doi.org/10.4324/9781003262541>.
- Petz, Cindarella. 2022. On Combining Network Research and Computational Methods on Historical Research Questions and its Implications for the Digital Humanities. Diss. Technische Universität München. <http://mediatum.ub.tum.de/node?id=1624881> (zugegriffen: 15. April 2024).

- Pfeifer, Wolfgang. 1993. Algorithmus. In: *DWDS*. Etymologisches Wörterbuch des Deutschen (1993), digitalisierte und von Wolfgang Pfeifer überarbeitete Version im Digitalen Wörterbuch der deutschen Sprache. <https://www.dwds.de/wb/etymwb/Algorithmus> (zugegriffen: 15. April 2024).
- Piper, Andrew. 2018. *Enumerations: Data and Literary Study*. Chicago, London: The University of Chicago Press.
- Putnam, Lara. 2016. The Transnational and the Text-Searchable: Digitized Sources and the Shadows They Cast. *The American Historical Review* 121, Nr. 2: 377–402. <https://doi.org/10.1093/ahr/121.2.377>.
- Rawson, Katie und Trevor Muñoz. 2019. Against Cleaning. In: *Debates in the Digital Humanities 2019*, hg. von Matthew K. Gold und Lauren F. Klein, 279–292. Minneapolis, London: University of Minnesota Press. <https://doi.org/10.5749/j.ctvg251hk.26>.
- Rehbein, Malte. 2020. Historical Network Research, Digital History, and Digital Humanities. In: *The Power of Networks: Prospects of Historical Network Research*, hg. von Florian Kerschbaumer, Linda von Keyserlingk-Rehbein, Martin Stark und Marten Düring, 253–279. Digital Research in the Arts and Humanities. London: Routledge. <https://doi.org/10.4324/9781315189062>.
- Řehůřek, Radim und Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 46–50. Valletta, Malta: ELRA.
- Rosner, Frank, Alexander Hinneburg, Michael Röder, Martin Nettle und Andreas Both. 2014. Evaluating topic coherence measures. arXiv. <https://doi.org/10.48550/arXiv.1403.6397>.
- Ryan, Yann C. und Sebastian E. Ahnert. 2021. The Measure of the Archive: The Robustness of Network Analysis in Early Modern Correspondence. *Journal of Cultural Analytics* 6, Nr. 3: 57–88. <https://doi.org/10.22148/001c.25943>.
- Salton, G., A. Wong und C. S. Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM* 18, Nr. 11: 613–620. <https://doi.org/10.1145/361219.361220>.
- Schmidt, Benjamin M. 2012. Words Alone: Dismantling Topic Models in the Humanities. *Journal of Digital Humanities* 2, Nr. 1: 49–65. Online: <http://journalofdigitalhumanities.org/2-1/words-alone-by-benjamin-m-schmidt/> (zugegriffen: 15. April 2024).
- . 2016. Do Digital Humanists Need to Understand Algorithms? In: *Debates in the Digital Humanities 2016*, hg. von Matthew K. Gold und Lauren F. Klein, 546–555. Minneapolis, London: University of Minnesota Press. <https://doi.org/10.5749/j.ctt1cn6thb.51>.
- Schwandt, Silke. 2018. Digitale Methoden für die Historische Semantik: Auf den Spuren von Begriffen in digitalen Korpora. *Geschichte und Gesellschaft* 44, Nr. 1: 107–134. <https://doi.org/10.13109/gege.2018.44.1.107>.
- Seyfert, Robert und Jonathan Roberge, Hrsg. 2017. *Algorithmenkulturen: Über die rechnerische Konstruktion der Wirklichkeit*. Kulturen der Gesellschaft Band 26. Bielefeld: Transcript.
- Shadrova, Anna. 2021. Topic models do not model topics: epistemological remarks and steps towards best practices. *Journal of Data Mining & Digital Humanities* 2021. <https://doi.org/10.46298/jdmdh.7595>.
- Simmler, Severin, Thorsten Vitt und Steffen Pielström. 2019. Topic Modeling with Interactive Visualizations in a GUI Tool. In: *Proceedings of the Digital Humanities Conference*. Utrecht. <https://dariah-de.github.io/TopicsExplorer/> (zugegriffen: 15. April 2024).
- Singh, Jasmeet und Vishal Gupta. 2017. A systematic review of text stemming techniques. *Artificial Intelligence Review* 48, Nr. 2: 157–217. <https://doi.org/10.1007/s10462-016-9498-2>.
- Stachowiak, Herbert. 1973. *Allgemeine Modelltheorie*. Wien, New York: Springer Verlag. <https://archive.org/details/Stachowiak1973AllgemeineModelltheorie> (zugegriffen: 15. April 2024).
- Stone, Harold S. 1972. *Introduction to computer organization and data structures*. McGraw-Hill Computer Science Series. New York: McGraw-Hill.
- Tsamados, Andreas, Nikita Aggarwal, Josh Cowls, Jessica Morley, Huw Roberts, Mariarosaria Taddeo und Luciano Floridi. 2022. The ethics of algorithms: key problems and solutions. *AI & SOCIETY* 37, Nr. 1: 215–230. <https://doi.org/10.1007/s00146-021-01154-8>.
- Turing, Alan M. 1937. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, Nr. 42: 230–265. <https://doi.org/10.1112/plms/s2-42.1.230>.

Turney, Peter D. und Patrick Pantel. 2010. From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research* 37, Nr. 1: 141–188.

Valcarce, Daniel, Alejandro Bellogín, Javier Parapar und Pablo Castells. 2018. On the robustness and discriminative power of information retrieval metrics for top-N recommendation. In: *Proceedings of the 12th ACM Conference on Recommender Systems*, 260–268. RecSys '18. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3240323.3240347>.

Valeriola, Sébastien de. 2021. Can historians trust centrality? Historical network analysis and centrality metrics robustness. *Journal of Historical Network Research* 6, Nr. 1: 85–125. <https://doi.org/10.25517/jhnr.v6i1.105>.

Wasserman, Stanley und Katherine Faust. 1994. *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences 8. New York: Cambridge University Press.

Wetherell, Charles. 1998. Historical Social Network Analysis. *International Review of Social History* 43: 125–144. <https://doi.org/10.1017/S0020859000115123>.

Yanofsky, Noson S. 2011. Towards a Definition of an Algorithm. *Journal of Logic and Computation* 21, Nr. 2: 253–286. <https://doi.org/10.1093/logcom/exq016>.

Zuse, Konrad. 1948. Über den Allgemeinen Plankalkül als Mittel zur Formulierung schematisch-kombinativer Aufgaben. *Archiv der Mathematik* 1, Nr. 6: 441–449. <https://doi.org/10.1007/BF02038459>.

Endnoten

- ¹ Vgl. zur Unterscheidung von theoretischen Algorithmen und praktischer Umsetzung im Code Tsamados u. a. 2022; Kitchin 2017, 17.
- ² Bauer 2007, 39, 73f.; Zusammenfassung bei Miyazaki 2012; auf Grundlage von Dantzig 2007; Khuwārizmī, Folkerts und Kunitzsch 1997.
- ³ Die drei von Yanofsky genutzten Begriffe sind mit verschiedenen Definitionen belastet. Zentral an der Arbeit ist jedoch die Unterscheidung der drei Ebenen und die Inklusion des abstrakten Ziels und der Implementierung in die Algorithmusdefinition. Siehe: Yanofsky 2011.
- ⁴ An dieser Stelle sei "digital" als Oberbegriff gemeint und nicht als explizite Digital Literacy, die im Folgenden beschrieben wird.
- ⁵ Letzteres wird teilweise auch als Statistical Literacy bezeichnet, vgl. Claes und Philippette 2020.
- ⁶ Gensim: topic modelling for humans. https://radimrehurek.com/gensim/auto_examples/index.html (zugegriffen: 19. März 2024).
- ⁷ Vorarbeiten dazu finden sich bei Frege 1879; hinsichtlich konkreter computationeller Anwendung dann bei Kleene 1935a; Kleene 1935b; Church 1936; Turing 1937; vgl. zur Einführung das öffentlich zugängliche Vorlesungsskript von Hartwig 1996.
- ⁸ Vgl. Schmidt 2016, 546: „Past a certain point, humanists certainly do *not* need to understand the algorithms that produce results they use; given the complexity of modern software, it is unlikely that they could. [...] What an algorithm does is distinct from, and more important to understand, than how it does it.”
- ⁹ Guter einführender Überblick in Digital Keywords bei Gillespie 2016.
- ¹⁰ Deutsches Textarchiv. <https://www.deutschestextarchiv.de/> (zugegriffen: 15. April 2024).
- ¹¹ Galleries, Libraries, Archives, Museums – Galerien, Bibliotheken, Archive, Museen.
- ¹² FernUniversität in Hagen: Archiv „Deutsches Gedächtnis“. <https://www.fernuni-hagen.de/geschichteundbiographie/deutschesgedaechtnis/index.shtml> (zugegriffen: 15. April 2024).
- ¹³ Siehe exemplarisch für die möglichen Anwendungsszenarien in den Digital Humanities: Luhmann und Burghardt 2021.
- ¹⁴ Siehe dazu etwa Piper 2018, 66, 70–75; Horstmann 2018, Abs. 4-7.
- ¹⁵ Blei, Ng und Jordan 2003, Anm. 1 auf 996; Shadrova 2021; vgl. auch Althage 2022, 266f.
- ¹⁶ Gensim: Latent Dirichlet Allocation. <https://radimrehurek.com/gensim/models/ldamodel.html> (zugegriffen: 19. März 2024).
- ¹⁷ Vgl. zur kritischen Diskussion Rawson und Muñoz 2019; Kunilovskaya und Plum 2021.
- ¹⁸ Es gibt Implementierungen, die selbst die Anzahl der Cluster schätzen. Vgl. etwa BERTopic, ein Ansatz, Topic Modeling mit Embeddings, TF-IDF und UMAP umzusetzen (Grootendorst o. J.).
- ¹⁹ Düring u. a. 2016; Gamper, Reschke und Düring 2015; Kerschbaumer u. a. 2020; Journal of Historical Network Research: <https://jhnr.uni.lu/index.php/jhnr> (zugegriffen: 28. Januar 2024) bzw. seit Ende 2023: <https://jhnr.net/> (zugegriffen: 26. Juli 2024).

- ²⁰ Brughmans, Collar und Coward 2016; Collar 2022; Knappett 2013.
- ²¹ So etwa: auf welche Art Beziehung zwischen welchen Akteuren fokussiert sich das Forschungsinteresse, um das Netzwerk zu modellieren bzw. den Untersuchungsgegenstand abzubilden? Auf welchen sozio-kulturellen / -politischen Vorannahmen beruht der Forschungsgegenstand, z. B. wie funktioniert die zu untersuchende Gesellschaft eigentlich?
- ²² Nach Koschützki, Lehmann, Peeters, u. a. 2005, 29, eq. 3.14.
- ²³ Gephi – The Open Graph Viz Platform. <https://gephi.org/> (zugegriffen: 19. März 2024); Gephi nutzt für die Berechnung von *betweenness centrality* den Algorithmus nach Brandes 2001.
- ²⁴ Nodegoat, <https://nodegoat.net/> (zugegriffen: 19. März 2024); Nodegoat, Documentation. Create Graph. <https://nodegoat.net/documentation.s/86/create-graph> (zugegriffen: 15. April 2024).
- ²⁵ Visone. Visual Social Networks. <https://visone.ethz.ch/> (zugegriffen: 19. März 2024). In Visone ist es möglich, *betweenness*, siehe visone manual. Betweenness. <https://visone.ethz.ch/wiki/index.php/Betweenness> (zugegriffen: 15. April 2024), und *current flow betweenness* (Algorithmus im Wiki zum gegenwärtigen Zeitpunkt nicht spezifiziert) zu berechnen.
- ²⁶ Vgl. Borgatti, Carley und Krackhardt 2006, die allerdings die Aussagekraft ihrer Studie einschränken, die auf “random error on random networks” basiert und damit nicht die unbewusst oder bewusst intentionalen Prozesse einbezieht, die real world networks, insbesondere in ihrer historisch motivierten Ausprägung, zugrunde liegen, siehe dazu ebd., S. 135. Vgl. auch Düring 2016; Valeriola 2021; siehe auch das „Network Robustness Tool“ des „Networking Archives“-Projekts https://networkingarchives.shinyapps.io/network_robustness_tool/ (zugegriffen: 15. April 2024); sowie Ryan und Ahnert 2021.
- ²⁷ Dabei spielt die Dichte (Vernetztheit) eines Netzwerks eine Rolle: beispielsweise weisen unimodale Projektionen bimodaler Netzwerke in der Regel eine höhere Dichte auf (sind also stärker vernetzt) als von vorneherein unimodal konstruierte Graphen, so dass auch die auf diesem Netzwerk berechneten Zentralitätsmaße im Durchschnitt höhere Werte aufweisen werden.
- ²⁸ Visone manual. Visualization tab: spring embedder. https://visone.ethz.ch/wiki/index.php/Visualization_tab#spring_embedder (zugegriffen: 15. April 2024).
- ²⁹ Zum Beispiel Programming Historian. <https://programminghistorian.org/> (zugegriffen: 19. März 2024).
- ³⁰ Zum Beispiel ZfdG – Zeitschrift für digitale Geisteswissenschaften. <https://zfdg.de/> (zugegriffen: 19. März 2024).

Zitierweise

Althage, Melanie / Deicke, Aline / Hall, Mark / Möbus, Dennis / Petz, Cindarella / Seltmann, Melanie (2024): Algorithmenkritik. In: Living Handbook "Digitale Quellenkritik". Version 1.2. hrsg. v. Deicke, Aline; Geiger, Jonathan D.; Lemaire, Marina; Schmunk, Stefan. <https://doi.org/10.5281/zenodo.12648832>

Metadaten

Autor:innen	Althage, Melanie ; Deicke, Aline ; Hall, Mark ; Möbus, Dennis ; Petz, Cindarella ; Seltmann, Melanie ;
Sprache	Deutsch
DOI	10.5281/zenodo.12648832
Creative Commons Lizenztyp	Attribution CC BY (4.0)
Verwandte Artikel	1. Einleitung: Was ist Digitale Quellenkritik? ;