

I. Visual Linguistics

Noah Bubenhofer

Visual Linguistics: Plädoyer für ein neues Forschungsfeld

Abstract Diagramme spielen auch in der Linguistik eine große Rolle. Ob der Verständlichkeit, mit der Diagramme erstellt und verwendet werden, geht die Reflexion über die diagrammatische Praxis manchmal verloren. Der folgende Beitrag ist ein Plädoyer, diese Praxis aus drei unterschiedlichen Perspektiven zu befragen: Aus diagrammatischer, algorithmischer und wissenschaftsgeschichtlicher Perspektive. Dieses Programm einer „Visual Linguistics“ stellt Fragen nach dem Charakter von Diagrammen, dem Status von Diagrammen in Forschungsprozessen und insbesondere dazu, welchen Einfluss Digitalität auf die Visualisierung sprachlicher Phänomene ausübt. Schließlich kann mit Ludwik Fleck die diagrammatische Praxis in Beziehung zu wissenschaftlichen Denkstilen gesetzt werden. Vor dem Hintergrund dieser Überlegungen ergeben sich fünf diagrammatische Grundformen, die bei der Visualisierung von sprachlichen Daten eine wichtige Rolle spielen: Liste, Karte, Partitur, Vektoren, Graph/Netz. Listen und Partituren werden im vorliegenden Beitrag ausführlich diskutiert und es wird gezeigt, welche Rolle sie bei der Gegenstandskonstitution in der Linguistik haben.

1. Einleitung

Dialektkarten, Syntaxbäume, Kollokationsgraphen, Kommunikationsmodelle, Gesprächstranskripte, geclusterte Netzwerke: Die Linguistik ist geprägt von Diagrammen verschiedenster Art, um theoretische Modelle zu visualisieren, Daten zu explorieren oder Ergebnisse zu veranschaulichen. Damit nimmt die Disziplin keine Sonderstellung ein gegenüber anderen Disziplinen. Wissenschaftliche Visualisierungen sind überall ein wichtiges Mittel mit vielfältigen Funktionen.

Der Umgang mit Visualisierungen in der Linguistik ist aber, um es positiv zu formulieren, unbeschwert und vorwiegend kanonisch. Unbeschwert, weil weitgehend eine Reflexion darüber fehlt, welche semiotischen und hermeneutischen Eigenschaften und Effekte und welche wissenschaftsgeschichtlichen Implikationen Visualisierungen haben. Kanonisch, weil Visualisierungen in der Linguistik

mehrheitlich als Mittel zum Zweck angesehen werden und deren Gebrauch deswegen bewährten Praktiken folgt, das Experiment mit alternativen Visualisierungen aber gescheut wird.

Im Folgenden möchte ich für ein Programm plädieren, das zwar der Visualisierungspraxis in der Linguistik die Unbeschwertheit nimmt, dafür aber Erkenntnisse bietet, die für das Selbstverständnis des Fachs von Bedeutung sind, die Gründe für die Wirkmächtigkeit von Visualisierungskanons untersucht und zum Experiment anstiftet. Dieses Programm nenne ich „Visual Linguistics“.

Die Eckpunkte des Programms ergeben sich aus drei Perspektiven, die mir fruchtbar für eine Analyse von Visualisierungspraktiken erscheinen und die ich im Folgenden näher beschreiben möchte:

- Die diagrammatische Perspektive: Was macht ein Diagramm zum Diagramm und welchen Status hat es in Forschungsprozessen?
- Die algorithmische Perspektive: Was ändert sich bei der Visualisierung von sprachlichen Phänomenen, wenn die Daten digital vorliegen und Visualisierungen computergeneriert sind?
- Die wissenschaftsgeschichtliche Perspektive: Warum ist das Diagramm als Drittes zwischen Sprache und wissenschaftlichem Arbeitsinstrument so wirkmächtig, dass es nicht nur Denkstile repräsentiert, sondern diese auch prägt?

Um es deutlich zu sagen: Für die jeweiligen Perspektiven gibt es einige Vorarbeiten aus der Philosophie und Semiotik (Diagrammatik), den Digital Humanities (Software Studies, Critical Code Studies), Medienwissenschaften (Computer as Medium) und der Wissenssoziologie und Wissenschaftsgeschichte. Die Verbindung dieser Erkenntnisse und die Anwendung auf die Linguistik ist jedoch nach wie vor ein Desiderat. Zudem ergaben sich innerhalb weniger Jahre und ergeben sich auch weiterhin durch die Masse digitaler Daten für die Geistes- und Kulturwissenschaften entscheidende Veränderungen, die mitbedacht werden müssen.

2. Die diagrammatische Perspektive

Wissenschaftliche Visualisierungen sind Diagramme, die eine dritte Position zwischen Bild und Text einnehmen, einer „Schriftbildlichkeit“ (Krämer 2012a) angehören. Sie sind ein Ensemble von grafischen Ausdrucksmitteln, die im Verhältnis einer „entworfenen Ähnlichkeit“ (Bauer und Ernst 2010: 18) zum Gemeinten stehen.

„Diagramme sind, so könnte man vielleicht sagen, graphische Abkürzungsverfahren für komplexe Schematisierungen. Sie bewahren ein Minimum ästhetischer Anschauung, das wir benötigen, um zu verstehen, wovon die Rede ist, vor allen Dingen, um uns von abstrakten Sachverhalten in buchstäblichem Sinn ein Bild machen zu können.“ (Stetter 2005: 125)

Entscheidend ist bei einem Diagramm die Typenbildung (vgl. Abb. 1). Durch sie unterscheidet sich das Diagramm vom Bild (Stetter 2005: 125).

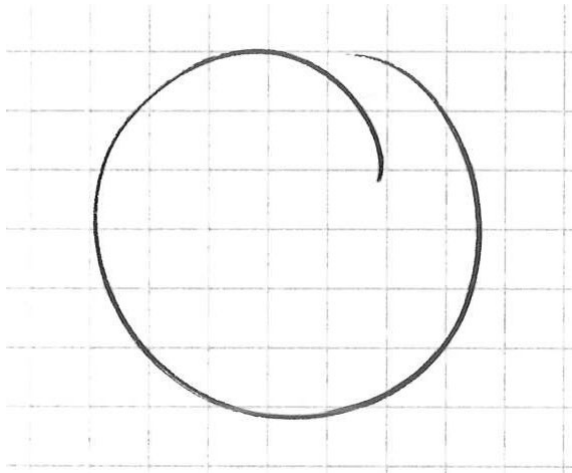


Abb 1: Ein Kreis als Diagramm.

Wenn diese Form als Realisierung eines abstrakten Typs, nämlich eines Kreises, wahrgenommen wird, ist die Form ein Diagramm. Wir abstrahieren von der tatsächlichen Form und sehen darin die idealisierte Form des Kreises. Die grafische Form steht in einem ikonischen Verhältnis zum Denotat: „Die ikonischen Zeichen geben einige Bedingungen der Wahrnehmung des Gegenstandes wieder“ (Eco 2002: 205); die Visualisierung, das Diagramm beschränkt sich also auf bestimmte Aspekte des Denotats. Diese Beschränkung ist allerdings willkürlich und deswegen konventionalisiert; die Wiedergabe funktioniert also „erst, nachdem diese [Bedingungen] auf Grund von Erkennungs-codes selektioniert und auf Grund von graphischen Konventionen erläutert worden sind“ (Eco 2002: 205).

Diagramme werden mit Peirce (1994) gesprochen als „Sinzeichen“ wahrgenommen, als „Verwirklichung eines abstrakten Modells“ (Eco 1977: 58). Ihre grafischen Variationsmöglichkeiten wie Strichdicke, Färbung, leichte

Formabweichungen sind nicht bedeutungstragend,¹ im Unterschied beispielsweise zu Bildern, bei denen solche „Qualizeichenmerkmale“ bedeutungstragend sind (Peirce 1994: 2.244; Eco 1977: 58–59). Sie entwerfen eine „proportionale Homologie“, bei der sie „eine Logik darstellen, die vom selben Gesetz beherrscht wird wie die Diagramme“ (Eco 1977: 143). Das wird deutlich, wenn beispielsweise ein Balkendiagramm betrachtet wird: Die Länge der Balken steht keineswegs in einem Ähnlichkeitsverhältnis zum Denotat, also einer Menge gezählter Entitäten, sondern die Balkenlängen stellen untereinander Proportionen dar, die homolog sind zu den denotierten abstrakten Zahlenverhältnissen. Der Zauber des Diagramms liegt darin, dass mit diesem grafischen Modus, mit dem die Balken gezeichnet werden können, beliebige Zahlenproportionen abgelesen werden können. Deswegen steht das Diagramm in einem Ähnlichkeitsverhältnis zum abstrakten Denotat der Zahlenverhältnisse.

Damit wird ein weiterer Aspekt deutlich, nämlich dass mit Diagrammen operiert werden kann (Krämer 2009): Besonders anschaulich dafür wird dies bei Karten. Auf ihnen wird mittels grafischer Elemente ein Schema eines bestimmten Ausschnittes des Realraums als Virtualraum dargestellt. Mit diesem Diagramm kann nun operiert werden, sobald die eigene Position und Ausrichtung darauf lokalisierbar ist. Die Karte macht eine Voraussage darüber, wo man einen Weg, ein Hindernis usw. antreffen wird, wenn man sich in eine bestimmte Richtung bewegt. Durch das Operieren mit der Karte ist es möglich, einen Weg zu finden (Krämer 2012b). Genauso ist das Sehen von Zahlenrelationen im Balkendiagramm eine solche Operation.

Besonders wichtig sind aber Operationen in Diagrammen bei Visualisierungen explorativer Datenanalysen (Chen u. a. 2008; Keim u. a. 2010): In solchen Visualisierungen wird bereits verfügbares Wissen diagrammatisch dargestellt, allerdings in einer für Operationen mit dem Diagramm optimierten Form. Durch die Arbeit mit dem Diagramm werden bestimmte Zusammenhänge erst sichtbar und dadurch neues Wissen aus dem Diagramm gezogen.

Obwohl solche explorativen Visualisierungen in neuerer Zeit etwa im Paradigma der Visual Analytics für Data-Mining-Aufgaben besonders wichtig sind, muss betont werden, dass dieses operative Element grundsätzlich in allen Diagrammen vorhanden ist (allerdings mit mehr oder weniger starken Ausprägung)

1 Selbstverständlich können in Diagrammen Strichdicken, Färbungen etc. sehr wohl bedeutungstragend sein. Es gibt aber immer ein Maß an Abweichung davon, das nicht mehr bedeutungstragend ist: Bei einem handgezeichnetem Diagramm, bei dem die Farben Rot und Blau zur Markierung von Klassen o. Ä. eingesetzt werden, werden alle ähnlichen Farbtönungen, die beispielsweise durch unterschiedlichen Druck des Stifts beim Zeichnen entstanden sind, unter der gleichen Farbe subsumiert.

und vor allem kein neues Phänomen ist. Ein Beispiel dafür ist eine Passage aus Platons *Menon*, wo die Erkenntnisfunktion des Diagramms deutlich wird (Krämer 2009): Die Aufgabe herauszufinden, welche Operationen nötig sind, um ein neues Quadrat mit der doppelten Fläche des ursprünglichen Quadrates zu erhalten, wird allein diagrammatisch durch Zeichnen gelöst. Durch die erste, fehlerhafte, Operation, nämlich das Verdoppeln der Seitenlängen, erwächst aus der so erweiterten Zeichnung die Erkenntnis, dass damit nicht die doppelte, sondern vierfache Fläche erzeugt worden ist. Der entscheidende Schritt ist nun zu sehen, dass die Hälfte dieser vervierfachen Fläche reichen würde und dass dies durch das Halbieren der vier entstandenen Quadrate mit einer Diagonale erreicht werden kann. Nur im Diagramm und nur durch die Operation damit kann diese Tatsache gesehen werden:

„Wir sehen *in* der diagrammatischen Figur ein mathematisches Konzept; und wir sehen in den Transformationen der Figur den *verallgemeinerbaren Lösungsweg* eines generellen Problems, d. h. also eine mathematische Einsicht.“ (Krämer 2009)

Darin liegt die Hoffnung jeglicher Visual Analytics, also der explorativen, visuellen Datenanalysen, verborgen: Durch die geschickte Überführung von Wissen in ein Diagramm damit operieren und eine verallgemeinerbare Erkenntnis daraus ziehen zu können, weil das Diagramm eben in einem schematisierten, ikonischen Verhältnis zu den Daten steht. Dies bedeutet nicht, dass die diagrammatische Operation immer ausreicht, um Erkenntnisse zu generieren – oft werden aus solchen Operationen Hypothesen generiert, die danach wiederum mit anderen Methoden getestet werden müssen.

In den Beiträgen in diesem Buch finden sich einige Beispiele für visuelle Analysemethoden. Ein Beispiel ist der „Topic Explorer“ von Hinneburg und Oberländer (S. 269), bei dem eine komplexe Darstellung von Topic Models gerechnet auf einem beliebigen Textkorpus die Interaktion mit ebendiesem Modell erlaubt. Deutlich wird bei diesem Beispiel aber auch, dass sich das Operieren mit solchen visuellen Analyseinstrumenten selten auf rein diagrammatische Aspekte beschränkt, sondern damit auch die statistische Modellierung davor beeinflusst wird. Hinneburg und Oberländer erhoffen sich somit nicht nur, durch das Operieren neue Erkenntnisse aus den Daten ziehen zu können, sondern damit auch eine „Plausibilitätsstruktur zur Interpretation der Themen“ (S. 270) erschließen – also das statistische Modell des (in diesem Fall) Topic Models verstehen zu können „ohne detaillierte Kenntnisse der zugrunde liegenden mathematischen Theorie“ (S. 270) besitzen zu müssen.

Ich werde weiter unten deshalb auch argumentieren, dass ein algorithmisch erstelltes Diagramm, um Operationalität zu ermöglichen, mehr sein muss als ein interaktives Diagramm.

Die meisten Diagramme erlauben zumindest eine minimale Form des Operierens. Besonders ausgeprägt ist dies bei den oben bereits erwähnten explorativen Visualisierungen, wie sie in der datenintensiven Geistes- und Sozialwissenschaften zur Anwendung kommen. Neben diesen explorativen Visualisierungen gibt es aber auch solche, die primär der Präsentation von bereits gewonnenen Erkenntnissen dienen. Solche Präsentationsgrafiken (Chen u. a. 2008: 5) sollen eine Erkenntnis klar und deutlich visualisieren. In der Varietätenlinguistik dienen Karten mit eingezeichneten Isoglossen diesem Zweck: Die zugrunde liegenden Daten, die zur Verortung der Isoglossen geführt haben, sind ggf. nicht mehr sichtbar. Eine Varietätenkarte aber, auf der viele erhobene Merkmalsausprägungen visualisiert werden, dient eher der Exploration; im besten Fall können durch die Arbeit mit der Karte Dialekträume bestimmt werden.

Schließlich dienen Diagramme aber oft auch dazu, mehr oder weniger komplexe Modelle zu skizzieren. Marc Richard Lauersdorf verweist in diesem Band auf eine Reihe von Diagrammen, die in der Linguistik dazu dienten, Theorien zu verdeutlichen und die zu „ikonischen“ Visualisierungen geworden sind (S. 91). Der Beitrag von Jana Pflaeging in diesem Band (S. 123) zeigt Beispiele dafür, wie Bildmetaphern verwendet werden können, um theoretische Konzepte zu verdeutlichen – die, so Pflaeging, zu einem „lange verweilenden Blick“ und „wilder Semiose“ führen können. Die grundsätzliche Offenheit gegenüber unterschiedlichen Interpretationen von Diagrammen betont auch Lauersdorf und konzipiert Diagramme deswegen als *objets d'art* (S. 93). In der Linguistik haben Visualisierungen von Theorien eine lange Tradition, denkt man etwa an die verschiedenen Darstellungen des Zeichenkonzeptes bei Saussure („Ei“), Bühler, Morris etc., die als Abbreviationen der jeweiligen Konzepte zum Fachwissen gehören.

Fast alle Visualisierungen in diesem Band sind algorithmisch erstellt und dienen der Exploration von Daten. In der datenintensiven Linguistik und den Digital Humanities ist dies eine Visualisierungsform, die in den letzten Jahren besonders viel Aufmerksamkeit gewinnen konnte. Ich möchte mich deshalb im Folgenden auf diesen Typus fokussieren und als Nächstes diese algorithmische Perspektive einnehmen und ausführlicher diskutieren. Meine Argumentation, insbesondere anschließend bei der wissenschaftsgeschichtlichen Perspektive, beschränkt sich jedoch nicht grundsätzlich auf diesen Typus.

3. Die algorithmische Perspektive

Die Visualisierungen, wie sie in datenintensiven Geisteswissenschaften verwendet werden, sind Kinder einer „Welt der Maschinen“ (Moles 1959; zit. nach Hörl 2008: 94), und das in zweifacher Weise: Einerseits modellieren sie auf algorithmische Weise ein Phänomen, simulieren dies also computertechnisch und lassen den Computer „zu einem wesentlichen epistemischen Tool“ (Hörl 2008: 97) werden. Andererseits ist bereits das Phänomen selber digital. Auch wenn die digitalen Daten auf nichtdigitale Gegenstände der Welt referieren (Texte, Bilder, Personen), werden sie algorithmisch nur greifbar in der übersetzten, digitalen Form, sie werden in den „Bestand des Technischen“ integriert (Rheinberger 1994: 409).

3.1 Computer als Metamedium

Die oben genannten methodischen Paradigmen – datengeleitete Analysen, Visual Analytics – sind möglich dank bestimmter technischer Voraussetzungen, nämlich Mitteln der Informatik. Zum einen benötigt man Rechner, also Hardware, um Daten digital verarbeiten zu können, zum anderen Software. Selbst wenn eine Visualisierung händisch skizziert wird, wird sie in fast allen Fällen für die Publikation digital weiterverarbeitet, nachdem sie gescannt worden ist. In den meisten Fällen entstehen Visualisierungen jedoch bereits im digitalen Raum – und meist sogar algorithmisch.

Für die Erstellung von wissenschaftlichen Visualisierungen hat sich eine Praxis entwickelt, in der bestimmte Programme dafür verwendet werden. Besonders weit verbreitet beispielsweise sind sog. Office-Programme wie Microsoft Excel oder OpenOffice Calc.² Neben Software, die leicht über eine grafische Benutzeroberfläche bedient kann, werden jedoch auch Programmiersprachen wie „R“ oder „JavaScript“ (und viele weitere) verwendet, insbesondere um algorithmische Visualisierungen zu erzeugen (siehe dazu den Beitrag von Wolfer/Hansen-Morath, S. 227). Eine Programmiersprache ist dabei ein Mittel, um sehr flexibel und genau Anweisungen an einen Rechner zu formulieren, bestimmte Operationen durchzuführen (vgl. dazu die gut lesbare Einführung von Ford 2015). Programmiersprachen gibt es auf unterschiedlichen Abstraktionsebenen; eine sog. höhere Programmiersprache erlaubt beispielsweise die Formulierung von

2 Interessanterweise entstehen Visualisierungen auch mit Software, die nicht dafür gedacht ist, etwa mit Microsoft PowerPoint, das für die Präsentation von Folien, nicht aber die Erstellung von druckfähigen Grafiken entwickelt wurde. Daran zeigt sich die Wirkmächtigkeit von Praktiken, die weit über die intendierten Verwendungsweisen eines Werkzeugs hinausgehen können.

Anweisungen, die sich z. B. so paraphrasieren lassen: *Mache für jedes Wertepaar der Spalten 3 und 4 im Datensatz einen Punkt, wobei der Wert der Spalte 3 der Position auf der x- und der Spalte 4 der y-Achse entspricht.* Dabei kann die Art des Punktes von den Daten abhängig gemacht werden oder Interaktionen definiert werden, etwa: *Wenn der User mit der Maus über einen Punkt fährt, zeige den Wert aus Spalte 1 der entsprechenden Zeile des Datensatzes an derselben Position an.*

Bei der Ausführung des Codes wird die hochsprachliche Anweisung letztlich in sog. Maschinencode überführt. Dies sind einfache Anweisungen an den Mikroprozessor, bestimmte Speicherbereiche mit bestimmten Werten zu belegen.

Nun lohnt es sich allerdings, genauer über die Funktionsweise eines Computers nachzudenken, und zwar aus linguistischer Perspektive. Erhellend ist dabei die Geschichte, die Heilmann (2012) erzählt: Die Geschichte des Computers als Schreibmaschine. Es gibt drei Typen des Schreibens im Zusammenhang mit Computern: „das Schreiben für Computer (aber nicht an ihnen), das Schreiben für und an Computern, und das Schreiben an Computern (aber nicht für sie)“ (Heilmann 2012: 8). Diese drei Typen stehen für drei unterschiedliche Epochen der Computergeschichte und drei unterschiedliche Verhältnisse zwischen Menschen und Computern. Zunächst, in den 1940er-Jahren, dienten Computer als Automaten. Ihnen wurde ein Set von Anweisungen eingeschrieben, das sie für beliebige Eingabewerte abarbeiteten, z. B. um Flugbahnen von Geschossen zu berechnen. Das Regelset wurde dabei handschriftlich auf Formblättern formuliert und dann in Form von Lochkarten eingegeben, in denen die gewünschten Speicherbereiche und Operationen codiert waren. Um diesen beschwerlichen Vorgang zu vereinfachen, fasste man häufig verwendete Anweisungen zu „Subroutinen“ zusammen, daraus entwickelten sich dann – verkürzt dargestellt – Programmiersprachen. In den 1960er-Jahren kam es dann zu einem entscheidenden Wandel: Der Anschluss von Schreibmaschinen oder Fernschreibern erlaubte „Interactive Computing“: Die Anweisungen konnten sozusagen live während der Ausführung eines Programms abgesetzt und so das Programm beeinflussen. „Diese neue Art des Programmierens erforderte selbst passende Programme (sogenannte Editoren) und es stellte sich schnell heraus, dass Computer nicht nur zum Schreiben von Code taugten, sondern auch dem Verfassen von technischen Berichten, Artikeln und Dokumentationen dienlich sein konnten“ (Heilmann 2012: 8).

Damit wurde Text offensichtlicher Gegenstand des Computers – textuell funktionierte er jedoch schon von Anfang an: *Codieren* bedeutet nämlich „anschreiben“ von Werten in Form eines Systems diskreter Zeichen als Verweise auf Speicherorte. Und *kalkulieren* bedeutet schriftliches Operieren mit diesen Zeichen, genau so, wie wir beispielsweise durch schriftliche Operationen auf Papier eine Multiplikation durch geschicktes, regelhaftes Schreiben durchführen können (Heilmann 2012: 42 in Rückführung auf Kittler 1989; Turing 1936). Der Computer diente also bereits von Anbeginn der Verarbeitung von Text

(Programmcode, Quelltext), operierte schriftlich und gab Text aus. Die schriftliche Angabe der Zahl 452 in Form einer Lochkartenanweisung oder des in einer höheren Programmiersprache definierten Ausdrucks „ $x = 452$ “ ist für den Computer keine Zahl, sondern sind ein bestimmtes Set von Schaltzuständen von Transistoren. Der Ausdruck „ $x = 452$ “ wird semantisch entleert und rein *syntaktisch* übersetzt in Schaltzustände (Heilmann 2012: 65) und unterscheidet sich nicht grundsätzlich von der Anweisung „ $v = \text{Haus}$ “ – beides sind Schaltzustände, mit denen, ausgelöst durch Text, operiert wird.

Auffallend ist dabei die Ähnlichkeit zum Operieren mit einem Diagramm. Die manuelle Methode des „schriftlichen Rechnens“ zeigt bereits die Bedeutung des Diagrammatischen: Um mehrere Zahlen zu summieren, ordnet man sie als Liste untereinander stellenweise parallel rechtsbündig an, um dann Stelle für Stelle von rechts nach links summieren zu können. Die Positionierung der Zahlen im Raum ist dabei entscheidend und funktioniert, weil die Verschriftlichung der mathematisch abstrakten Zahlen in einer Systematik geschieht, die genau dies erlaubt. Ähnlich geschieht dies nun beim Computer, bei dem die abstrakten Zahlen nicht direkt verschriftlicht, sondern durch Schaltzustände als binäre Zahlen repräsentiert werden, die es aber dank ihrer Anordnung erlauben, genau so Operationen durchzuführen, wie wir das beim schriftlichen Rechnen im Dezimalsystem gewohnt sind. Insofern argumentiere ich, dass auch das „Rechnen“ des Computers einen diagrammatischen Charakter hat, da dieses Rechnen mit Schaltzuständen im Binärsystem eine proportionale Homologie entwirft: Das Setting der Schaltzustände (dessen Materialität bei den ersten Rechnern noch viel deutlicher hervortrat als heute) ist eine Art „Diagramm“, mit dem operiert wird, um im abstrakten mathematischen Raum eine Aufgabe zu lösen. In Anbetracht der Bedeutung des (schriftlichen) Codes, der diese Operation ermöglicht und der unzweifelhaft in seiner schriftlichen Form diagrammatische Mittel der Textformation nutzt (Listen, Gliederungen, Hierarchien, Einrückungen etc., vgl. Steinseifer 2013), könnte der Computer als eine Art „diagrammatische Maschine“ aufgefasst werden.

Hinzu kommt nun jedoch ein weiterer Aspekt: Mit der Wende hin zum Interactive Computing und damit den neuen Eingabe- und Ausgabemedien (Maus, Bildschirm) wird der Computer als eine Art Plattform wahrgenommen, um völlig unterschiedliche Dinge tun zu können. Kay und Goldberg (1977) charakterisieren ihn als „metamedium“:

„Although digital computers were originally designed to do arithmetic computation, the ability to simulate the details of any descriptive model means that the computer, viewed as a medium itself, can be *all other media* if the embedding and viewing methods are sufficiently well provided.“ (Kay und Goldberg 1977)“

Damit können mit diesem Metamedium auch neue Medien erfunden werden, die es bisher gar nicht gab. Manovich ist der Auffassung, dass dies weitreichende Konsequenzen hatte, wie heute Software als Medium funktioniert: „Once computers and programming were democratized enough, some of the most creative people of our time started to focus on creating these new structures and techniques rather than using the existing ones to make ‘content’“ (Manovich 2014: 81).

Entscheidend dafür ist, dass der Computer als „diagrammatische Maschine“ mittels dreier Grundfunktionen – Codierung, Algorithmisierung und Formatierung (Heilmann 2012: 195) – Daten speichert, überträgt und verarbeitet. Die Codierung übersetzt Eingaben in Transistorzustände (ins Digitale), die Algorithmen operieren damit und die Formatierung macht die abstrakten Transistorzustände wieder sicht- und deutbar. Es gibt jedoch unendlich viele verschiedene Formatierungen, keine ist die „eigentliche“ Repräsentation der Transistorzustände. Und wieder ist es Text, Programmcode, der die Formatierung der Daten bestimmt – im Falle von Textdaten „ist die Formatierung von Texten eine Sache der Beschreibung von Text *durch* Text. Zum ersten Mal in der Geschichte des Schreibens bestimmen damit Texte über die ‘Materialität’ der Schrift und nicht umgekehrt die Materialität der Schrift über die Gestalt von Texten“ (Heilmann 2012: 239).

Welchen Unterschied macht es, wenn Text über die Materialität der Schrift bestimmt und nicht die Materialität der Schrift (auf Papier gedruckt, in Holz gekerbt, mit Pinsel und Tusche gemalt) über die Gestalt von Texten?

Zunächst ergibt sich ein ungeheures Potenzial an Transformationen von Text in unterschiedliche Formen (vgl. dazu Jägers „transkribierende Verfahrenslogik“, Jäger 2007). Bedeutender ist jedoch, dass diese Transformationsprozesse live und rekursiv ablaufen können. Solange der Text den Computer nicht verlässt (z. B. ausgedruckt wird), so lange kann mit Text auf ihn Einfluss ausgeübt werden. Und der Text kann *sich selber* rekursiv beeinflussen.

Die „Visualisierung“ von Text (z. B. einer Frequenztafel) ist also eine textuell definierte Formatierung dieses Textes (in Form von Instruktionen), eine von vielen möglichen Transformationen von Text in eine bestimmte diagrammatische Form. Das Operieren mit diesem Diagramm beschränkt sich deshalb nicht nur darauf, mit dem formatierten Diagramm zu interagieren (gemeinhin *Interaktivität* genannt), sondern umfasst alle Transformationsprozesse, die im Computer möglich sind. Dies im Unterschied etwa zur Skizze auf Papier: Dort kann durch Zeichnen beispielsweise ein geometrischer Beweis geführt werden (vgl. oben Platons Menon-Beispiel), die materiellen Vorbedingungen, Papier, Stift, können jedoch nicht verändert werden. Bei der algorithmischen Visualisierungen jedoch wird selbst die Materialität operationabel – das Diagramm kann verschiedene Zustände der Materialität einnehmen: angezeigt auf einem Bildschirm, zweidimensional, [simuliert] dreidimensional etc. –, denn alles, was mit den digitalen

Daten im Computer passiert, sind diagrammatische Operationen. Der Computer ist eine diagrammatische Maschine.

Wenn aus diagrammatischer Perspektive also deutlich gemacht wird, dass mit Diagrammen operiert werden kann, dann wird aus algorithmischer Perspektive klar, dass diese Operationalität umfassend ist und das Diagramm nicht bloß eines der „Bilder“ ist, die durch Transformationen entstanden sind, sondern alle Prozesse davor bereits diagrammatisch sind. Eine interaktive Visualisierung ist dann umfassend interaktiv, also operationabel im diagrammatischen Sinn, wenn sie alle Interaktionen mit den Daten zulässt, also wenn die digitale Aufbereitung und algorithmische Beeinflussung der Daten genauso dazugehören wie die Beeinflussung der visuellen Darstellung. Es leuchtet natürlich ein, dass die Operationalität sinnvollerweise nicht bis auf die Ebene des Maschinencodes reichen soll: Wir möchten uns nicht damit herumschlagen, wie die Zeichen im Computer digital repräsentiert und gegenseitig verrechnet werden. Die Operationen von Interesse finden auf einer viel höheren Ebene statt – es geht um die Frage, welcher Art die Indizes sind, mit denen die Daten gespeichert werden, welche Aspekte der Daten überhaupt gespeichert, welche Transformationen in der Folge möglich sind und in welchen Formen die Daten materialisiert werden. Wenn mit Daten diagrammatisch operiert werden soll, dann sollten diese Aspekte potenziell kontrolliert werden können.

Diese umfassende Kontrolle über das Digitale ist natürlich genau der Witz des Interactive Computings, also des Computers, wie wir ihn heute kennen. Dieser Grundgedanke, mit dem Computer ein Metamedium bedienen zu können, wird allerdings durch Betriebssysteme, die möglichst viele potenzielle Interaktionsmöglichkeiten hinter einer schönen Metapher verstecken, oder durch Software, die besonders benutzerfreundlich gestrickt ist und Interaktionen nur auf der Oberfläche zulässt, sabotiert.

Muss man sich demnach als Geisteswissenschaftlerin oder -wissenschaftler bei der Nutzung von visuellen Analysemethoden mit Algorithmen auseinandersetzen? Unbedingt – und man sollte sich sogar mit Programmierkulturen auseinandersetzen, wie ich im Folgenden zeigen möchte.

3.2 Topoi in der Programmierpraxis

Es ist das Verdienst der Software Studies, auf die kulturelle Verfasstheit von Software aufmerksam gemacht zu haben (Fuller 2003; Mackenzie 2006). Und es ist einsichtig, dass die Gestaltung von Software eine kulturelle Praxis widerspiegelt und gleichzeitig reproduziert: Die grafischen Benutzeroberflächen, die „eine ganze Maschine ihren Benutzern entziehen“ (Kittler 1993: 233), ermöglichen (und verunmöglichen) gewisse Praktiken, die Art, wie Wissen als verarbeitbare

Daten repräsentiert wird (Datenbanktypen), ist nicht unabhängig von kulturellen Übereinkünften zu Wissenrepräsentationen (Manovich 2002; Dourish 2014).

Auch auf der Ebene des Programmiercodes begegnen einem Hinweise über die Einbettung in kulturelle Praktiken auf Schritt und Tritt. Die Wahl einer Programmiersprache „is the most important signaling behavior that a technology company can engage in“ (Ford 2015) und Ford nennt in seinem Text die überspitzten Klischees, die in der Gemeinde der Programmierer/innen den unterschiedlichen Programmiersprachen zugeschrieben werden:

„Tell me that you program in Java, and I believe you to be either serious or boring. In Ruby, and you are interested in building things quickly. In Clojure, and I think you are smart but wonder if you ship. In Python, and I trust you implicitly. In PHP, and we sigh together. In C++ or C, and I nod humbly. In C#, and I smile and assume we have nothing in common. In Fortran, and I ask to see your security clearance. These languages *contain entire civilizations*.“ (Ford 2015, Hervorhebungen NB)

Larry Wall, bezeichnenderweise Linguist, entwickelte eine der wichtigsten Programmiersprachen: Perl. 1999 hielt er an der Konferenz „LinuxWorld“ eine Rede mit dem Titel „Perl, the first postmodern computer language“ (Wall 1999). Auf sehr unterhaltsame Weise argumentiert Wall darin, dass Perl – und bis damals nur Perl – eine postmoderne Programmiersprache sei, im Gegensatz zu allen anderen, die in der Moderne stecken geblieben seien. Sie richte sich gegen vier die Informatik beherrschende Kulte: „spareness“, „originality“, „seriousness“ und „objectivity“. Perl ist eine quelloffene Sprache, die auf vielen Bruchstücken der Betriebssysteme Unix und Linux beruht und versucht, Lösungsdogmen zu vermeiden und dafür Diversität zuzulassen: „Perl programming is unabashedly genre programming. It has conventions. It has culture. Perl was the first computer language whose culture was designed for diversity right along with the language“ (Wall 1999).³ Walls Vortrag endet mit der These, dass es gerade die

3 Um keine falschen Vorstellungen zu wecken: Trotz der Diversität und dem Perl-Mantra „There’s More Than One Way To Do It“ handelt es sich um eine rigide Programmiersprache, die auf eine korrekte Verwendung ihrer Syntax angewiesen ist, um zu funktionieren. Im Vergleich zu anderen Programmiersprachen ist sie jedoch tatsächlich in vielen Aspekten wenig rigid; zu nennen ist beispielsweise, dass in Perl bei der Definition von Variablen nicht festgelegt werden muss, welche Inhaltstypen (Ganzzahlen, Kommazahlen, Buchstaben) darin vorkommen dürfen, sondern dies im entsprechenden Kontext automatisch geschieht.

Open-Source-Bewegung sei, die die oben genannten Kulte umstoße und so für einen kulturellen Wandel stehe.

Walls „Kulte“, von Fuller (2003: 15) als „idealist tendencies in computing“ bezeichnet, drehen sich um einen starken Topos der Informatik: Die Gleichsetzung von Purismus von Zahlen und Schönheit. Die ganze Welt kann in Verhältnisse von Zahlen aufgelöst werden und wird dann zu „purer Mathematik“. Je stärker sich diese Repräsentation durch Zahlen der puristischen Form annähern,

„the more beautiful they become. There is an endpoint to this passage to beauty which is absolute beauty. Access to and understanding of this beauty is allowed only to those souls that are themselves beautiful.“ (Fuller 2003: 15)

Fuller kritisiert in der Folge Tendenzen, „ästhetisches Programmieren“ zu einem Dogma zu erklären, und warnt, dass solche Ästhetiken zu sozialer Kontrolle führen. Die „märchenhaften“ [„fabulatory“] Programmieransätze seien weit interessanter, denn:

„Numbers do not provide big answers, but rather opportunities to explore further manifold and synthetic possibilities—that is to say, they provide access to more figures.“ (Fuller 2003: 16)

Eng verwandt mit dem Purismus-Topos ist der Utilitarismus-Topos: „Computer programming seems to carry to an extreme an understanding of technology as a utilitarian tool predicated on a reframing of the world as a set of calculable quantities“ (Goffey 2014: 21). Dieser Topos ist nicht nur für Computertechnik wirkmächtig, sondern für Technologie generell, wie Böhme (2006) zeigt. Er nennt Karl Marx als prägend für diesen Topos, der wohl als einer der ersten eine „theory of technology“ entwickelte, „to have a conception of technology as a social enterprise“ (Böhme 2006: 55). Die Auffassung, dass Technologie dem „Reich der Notwendigkeit, nicht der Freiheit“ angehört und dazu dient, die Lebensbedingungen zu verbessern und somit letztlich der Reproduktion der Menschheit dient, verankerte sich in der Folge fest in unserem Verständnis (Böhme 2006: 55–56). Allerdings entspringt diese Auffassung von Technologie einer vorherrschenden Theorie von Technologie, nicht ihrer tatsächlichen Geschichte (Böhme 2006: 56).

Die Geschichte nämlich brachte immer wieder Beispiele hervor für Technologien, die der Unterhaltung, dem Genuss, der Zerstreuung dienten, unabhängig jeglicher Nützlichkeitsziele. Böhme zeigt dies anhand der Technologien an den königlichen Höfen, Goffey nennt – neben den naheliegenden Computerspielen

– Beispiele aus der Computergeschichte: „The fascination with automata evinced within Department of Defense-funded research in artificial intelligence (AI) could perhaps be adduced as evidence that technologies of enjoyment were alive“ (Goffey 2014: 27).

Entscheidend in unserem Kontext ist aber, dass durch das Metamedium Software eine Möglichkeit entstanden ist, mit Programmcode nicht nur Modelle der *Welt*, also „Aneignungen von Natur“ (Böhme 2006: 55) als zweckrationale Tools zu entwickeln, sondern Modelle *virtueller* Welten, die selber Produkte des Computers sind: „Strictly speaking, many of the things that software models are not real-world processes at all, they are things that are brought into being by computational technologies themselves“ (Goffey 2014: 34). Software weist also eine „demiurgische“ Qualität auf, sie kann etwas erzeugen, was vorher noch gar nicht existierte (Goffey 2014: 35). Dabei geht eine Praxis der Programmierung in ein Stück Software auf, der die Praxis der Codeerzeugung nicht mehr anzusehen ist:

„clever tricks with assembly language, manipulating the side effects of an algorithm, devising workarounds, the smoke and mirrors of user interface design, and so on. The virtuoso smarts of programming practice that get software working can of course eventually be explained logically if the code compiles, the app works and users accept it. But in doing so, the process disappears into the product and the experimenting, the ‘bricolage’ gets forgotten.“ (Goffey 2014: 35)

Die semiotische Qualität von Programmiersprache wird damit deutlich: Programmiersprachen sind Sets von Codes, die im Rahmen einer Praxis des Programmierens in einem kulturellen Kontext neue Codes erzeugen, die wiederum als Zeichen gelesen werden. Software wird zu einer Äußerung: „software is a semiotic artefact, a set of operations in and on codes that implies the ongoing, repeated fact of enunciation“ (Goffey 2014: 36). Äußerungen sind aber nicht formallogisch vollständig erfassbar; „Enunciation brings the messiness of the world back and forces us to connect coding or programming with practice in more rigorous ways“ (Goffey 2014: 37).

3.3 Coding Cultures

Ein anschauliches Beispiel für die Einbettung von Programmierpraktiken in Kulturen bieten Hacker-Szenen. *Hacking* wird in diesem Zusammenhang verstanden als Tätigkeit, mit der nicht nur ein bestimmtes konstruktives Ziel erfüllt

wird, sondern einen Selbstzweck dient – im Fall des Computers beispielsweise das Programmieren des Programmierens willen; aus purer Freude (Levy 2010: 10). Dabei spielen Ideale wie freier Zugang zu Wissen, Redefreiheit, Transparenz, Chancengleichheit, Öffentlichkeit und ein meritokratisches Verständnis eine wichtige Rolle. In der Geschichte des Computerhackings sind diese Werte stark mit einem Engagement für liberale Softwarelizenzen verbunden: Erzeugter Code gehört allen, jede und jeder darf ihn modifizieren (Coleman 2012: 3). Diese Ansicht stellt ein deutliches Gegengewicht zur kommerziellen Softwareindustrie dar; ein Kampf gegen die oben bereits von Wall erwähnten „Kulte“.

Im Selbstverständnis der Hacker-Szenen spielen diese Werte, zunächst implizit, manchmal auch explizit ausformuliert in einer „Hacker-Ethik“, eine wichtige Rolle:

„It was a philosophy of sharing, openness, decentralization, and getting your hands on machines at any cost to improve the machines and to improve the world. This Hacker Ethic is their gift to us: something with value even to those of us with no interest at all in computers.“ (Levy 2010: IX)

Diskussionen über diese ethischen Prinzipien und Formulierungen von Manifesten begleiten die Geschichte des Hackens bis heute und werden teilweise heftiger diskutiert als technische Fragen (Coleman 2012: 18). Im Zeitalter der Start-ups des Silicon Valleys spricht Scott (2015) etwa von der gehackten, nämlich „gentrifizierten“ Hacker-Kultur: Der kommerziellen Vereinnahmung dieser Kultur, die damit des rebellischen, kritischen Moments beraubt wird.

Es liegt nahe zu untersuchen, welchen Einfluss Hacker-ethische Überlegungen auf die tatsächliche Praxis des Hackens – angefangen beim Auseinandernehmen und Modifizieren von Hardware über das Verändern bestehender Programme und dem Programmieren neuer Programme bis zum Hacken von Sicherheitssystemen – haben. Levy und Coleman (Coleman 2012; Levy 2010) zeigen dies über unterschiedliche Ansätze; Coleman argumentiert beispielsweise, wie sich in unterschiedlichen Varianten, die gleiche Funktion in der gleichen Programmiersprache zu programmieren, Witz und Können gleichzeitig manifestiert, was natürlich nur für die Eingeweihten überhaupt ersichtlich ist (Coleman 2012: 93ff.). Anders als bei proprietärer Software bleibt bei quelloffenen, geteilten Programmen die gefundene Lösung, die „Bricolage“ (vgl. oben; Goffey 2014: 35), sichtbar.

Die Computerwissenschaften waren quelloffenen und freien Software-Prinzipien grundsätzlich immer aufgeschlossen, zumal die ersten Hacker-Bewegungen an Universitäten entstanden sind (Levy 2010). In den Geistes- und

Sozialwissenschaften spielte und spielt jedoch proprietäre Software eine wichtige Rolle, seien es Office-Programme zur Textproduktion, Datenverwaltung und Präsentation oder Statistikprogramme wie SPSS.⁴ In jüngerer Zeit und vor allem im Bereich der Digital Humanities, der Korpus- und Computerlinguistik und der elektronischen Datenverarbeitung werden mit Programmiersprachen wie R, Perl, Python, Javascript etc. Techniken angewandt, die zu quelloffener und freier Software führen. Diese Programmiersprachen tragen ihre kulturelle Genese und Bedeutung in sich, auch wenn sie uninformiert und naiv eingesetzt werden – ihre Verwendung enthält eine Botschaft.

Es finden sich im Netz viele Texte, die sich an programmiertechnische Laien richten und argumentieren, warum eine bestimmte Programmiersprache besser ist als eine andere. Mit R lassen sich beispielsweise statistische Berechnungen und Visualisierungen erstellen, was teilweise natürlich auch mit dem viel bekannteren Office-Paket „Excel“ von Microsoft geht.⁵ Paradigmatisch für die Nennung der Vorteile von R gegenüber Excel ist beispielsweise der Text von Isaac Petersen mit dem Titel „Why R is Better Than Excel for Fantasy Football (and most other) Data Analysis“.⁶ Neben einigen technischen Argumenten werden auch Aspekte genannt, die eher ideologischer Natur sind, beispielsweise Quelloffenheit, Lauffähigkeit auf vielen Plattformen und die Möglichkeit, selber als Teil einer großen Community sich aktiv an der Weiterentwicklung zu beteiligen. Ähnlich argumentiert ein Blogbeitrag von Michael Milton, der zudem ein schönes Beispiel für den oben genannten Purismus-Topos darstellt:

„The visualizations you can create in R are much more sophisticated and much more nuanced. And, philosophically, you can tell that the visualization tools in R were created by people more interested in good thinking about data than about beautiful presentation. (The result, ironically, is a much more beautiful presentation, IMHO.)“⁷

4 Wobei es immer auch Gegenströmungen gab wie für die Textproduktion zum Beispiel die Verwendung des quelloffenen Satzsystemes LaTeX.

5 Die Möglichkeiten im Bereich statistischer Analyse und Visualisierung, aber auch der Datenaufbereitung und insbesondere der Einbettung in andere Programmier Routinen sind bei R deutlich vielfältiger als bei einer Software wie Microsoft Excel. Trotzdem begnügen sich viele Wissenschaftler/innen mit Excel, ggf. auch in Arbeitsroutinen, die R-Nutzer/innen als sehr unelegant bezeichnen würden.

6 <http://fantasyfootballanalytics.net/2014/01/why-r-is-better-than-excel.html> (25. August 2015).

7 <http://www.michaelmilton.net/2010/01/26/when-to-use-excel-when-to-use-r/> (24. August 2015).

Wenn man sich als Novize oder Novizin auf R einlässt, wird man automatisch seine Praxis ändern, sobald das erste Problem auftaucht und man nach einer Lösung sucht: Die Lösung findet sich nicht in einem singulären Handbuch, sondern in zig Diskussionsschnipseln im Netz. Dabei wird es nicht nur eine Lösung geben, sondern viele verschiedene. Bald wird auch klar werden, wodurch sich die Lösungen unterscheiden: Unterschiedliche Codevarianten mit gleichem Ausgang oder Abhängigkeit von bestimmten Zusatzpaketen mit Variationen der Ausgabe, wobei Argumente wie „Eleganz“, „Ästhetik“ oder „Sauberkeit“ für die eine oder andere Lösung auftauchen werden.

Obwohl der Umgang mit Computern von einem starken Utilitarismus-Topos durchdrungen ist, zeigen die Ausführungen oben, dass Praktiken der Programmierung und Softwarenutzung zutiefst kulturelle Praktiken sind. Die Verwendung von bestimmten Programmiersprachen und Programmen kommt einer kulturell bedeutsamen Botschaft gleich. Es macht einen Unterschied, ob die Visualisierung mit Excel oder R erstellt worden ist; die Entscheidung über das Programm oder die Sprache beeinflusst den Erkenntnisprozess genauso wie die Wahl der Daten, die Formulierung der Forschungsfrage oder die theoretische Grundierung der Analyse.

4. Die wissenschaftsgeschichtliche Perspektive

Eine „visuelle Linguistik“ muss sich zwingend mit wissenschaftsgeschichtlichen Theorien auseinandersetzen, um die Rolle der Visualisierungen in der Disziplin synchron als auch diachron zu verstehen. Neben Thomas S. Kuhns Buch *Struktur wissenschaftlicher Revolutionen* von 1962 (Kuhn 1996), das zweifellos eine bedeutende Stellung in der Wissenschaftsgeschichte einnimmt, sind die Arbeiten von Ludwik Fleck besonders gut anschlussfähig an die Desiderate einer visuellen Linguistik. Fleck hat in mehreren Arbeiten, ausgehend vom 1935 erschienenen Text *Entstehung und Entwicklung einer wissenschaftlichen Tatsache* (Fleck 1980), seine Konzepte Denkstil und Denkkollektiv entwickelt und plausibilisiert (Fleck 1983, 2011): Die Angehörigen einer wissenschaftlichen (Teil-)Disziplin eint ein gemeinsamer „Denkstil“, der sie zu einem „Denkkollektiv“ macht (Fleck 2011: 87). Das erschwert nicht nur das gegenseitige Verständnis über Denkkollektive hinweg, sondern es entsteht auch „eine spezifische Bereitschaft, dem Stil entsprechende Gestalten wahrzunehmen“ und es „verschwindet dagegen parallel das Vermögen, nicht stilgemäße Phänomene wahrzunehmen“ (Fleck 1983: 107). Ein Denkkollektiv ist also blind gegenüber Evidenzen, die nicht zum eigenen Denkstil passen.

Auf die Kommunikation innerhalb eines Denkkollektivs richtet Fleck ein besonderes Augenmerk. Erkenntnisse und Wissen werden innerhalb des

Kollektivs gemäß dem herrschenden Denkstil kommuniziert und verändern sich dadurch laufend, festigen aber gleichzeitig den Denkstil innerhalb des Kollektivs.

Flecks Beobachtungen zur Wirkmächtigkeit von Denkstilen in der Wissenschaft sind auch in der Linguistik – mit einigen Jahrzehnten Verzögerung – auf fruchtbaren Boden gefallen. Dabei stößt der Stilbegriff auf reges Interesse: Denn „das, was wir mit Denkstil meinen und am Denkstil beobachten, [muss sich] ja immer sprachlich materialisiert haben [...], um wahrnehmbar, beobachtbar zu sein“ (Fix 2011: 1; Möller 2007). Fleck nennt Mittel, die dazu dienen, einen Denkstil zu pflegen: „Ich möchte nur noch zwei Mittel erwähnen, über die der wissenschaftliche Denkstil verfügt, um seinen Produkten den Charakter einer Sache zu verleihen. Eines von ihnen sind *technische Termini* [...]. Das zweite Mittel ist *das wissenschaftliche Gerät* [...]. Wer es versteht, in ein Fernrohr zu schauen und an den Saturn zu denken, benutzt damit allein bereits einen bestimmten abgegrenzten Denkstil“ (Fleck 1983: 121f.). Ulla Fix hat plausibel gezeigt, wie anschlussfähig Flecks Überlegungen zur Rolle des Sprachgebrauchs zur Ausbildung von Denkstilen an Konzepte der Linguistik sind (Fix 2011). Sie nennt Kategorien wie Stil, Text, Wort, Varietät und Metapher als sprachliche Elemente des Wahrnehmbaren eines Denkstils.

Doch gehören Diagramme nicht ebenfalls zu den Mitteln, die wissenschaftliche Denkstile prägen? Und gehören sie zu den „technischen Termini“ oder zum „wissenschaftlichen Gerät“? Ich möchte argumentieren, dass Diagramme eine eigenartige Doppelfunktion zwischen Sprache und Gerät einnehmen – und gerade deshalb besonders wirkmächtig sind.

4.1 Visualisierungen als Zeichen

Zweifellos sind Visualisierungen komplexe Zeichen, die in einem ikonischen Verhältnis zum Referenten stehen. Sie weisen damit semiotische Qualitäten auf und funktionieren in der wissenschaftlichen Praxis genauso als kommunikatives Mittel wie sprachliche Zeichen, können also, z. B. Bühler folgend, darstellen, ausdrücken und appellieren (Bühler 1934). Mit Fleck gedacht können Visualisierungen dank dieser Funktionen innerhalb eines Denkkollektivs „den Produkten den Charakter einer Sache“ verleihen. Dies wird sofort evident, wenn man sich vergegenwärtigt, in welchen wissenschaftlichen Disziplinen gewisse Typen von Visualisierungen üblich, zwingend notwendig oder geradezu verpönt sind. Jedes Denkkollektiv pflegt eine Praxis des Visualisierens oder Nicht-Visualisierens, angefangen bei leichten Formen der Visualisierung wie textstrukturierenden Merkmalen (Spiegelpunkte, Listen, Tabellen) über übliche und kanonisierte

Formen (Balkendiagramme, Streudiagramme, Karten etc.) bis hin zu komplexen Formen (interaktive Visualisierungen).⁸

In der Linguistik entwickelten sich in bestimmten Teildisziplinen mehr oder weniger stabile Formen, etwa in der Dialektologie und Varietätenlinguistik, wo Karten eine wichtige Rolle einnehmen und sich bestimmte Praktiken durchgesetzt haben oder zumindest das Ziel der Standardisierung definiert wird. So formuliert etwa Naumann (1982) in der Dialektkartografie das folgende Desiderat:

„Die *Methodik* wird [...] immer deutlicher einen Rahmen für große Teile der Arbeitsschritte geben und damit dem Erfindungsreichtum Grenzen setzen, soweit das die Kommunizierbarkeit von Kartierung und Karte verbessert.“ (Naumann 1982: 687)

Es werde zwar weiterhin ein „sozusagen künstlerischer Anteil“ notwendig sein, wobei Naumann darunter alle Entscheidungen versteht, die nicht einer Systematik folgen (Naumann 1982: 668). Auf das „Künstlerische“, offensichtlich ein eigentlich in der Wissenschaft unerwünschter Aspekt, müsste noch zurückgekommen werden (vgl. aber die Beiträge von Pflaeging und Lauersdorf in diesem Band); der Wunsch einer Systematisierung von Visualisierungspraktiken, der in vielen wissenschaftlichen Disziplinen zu beobachten ist, zeugt natürlich davon, dass Visualisierung dort nicht als Ausschmückung, sondern als Arbeitsinstrument angesehen wird. Gleichzeitig wird daran sichtbar, wie wichtig definierte Visualisierungspraktiken für die Durchsetzung von Denkstilen in einem Denkkollektiv sind: Wer den kanonisierten Praktiken nicht folgt, muss eine überzeugende Innovation anbieten oder aber riskiert seine disziplinäre Glaubwürdigkeit.

Bemerkenswert beim Stellenwert von Visualisierungen in unterschiedlichen Disziplinen (und auch in historischer Perspektive) sind die Differenzen der Einschätzung darüber, wie akkurat Visualisierungen einen Gegenstand wiedergeben können und welchen Zwecken sie dienen. In stark empirisch ausgerichteten Disziplinen, z. B. der Korpuslinguistik, spielen Visualisierungen eine wichtige Rolle, die die Eigenschaften von Datenmengen wiedergeben sollen. Eine der einfachsten Formen ist dabei das Balkendiagramm, eine professionellere Praxis verwendet aber eine breite Palette von Streudiagrammen mit zusätzlichen

8 Karin Knorr Cetina (2001) etwa zeigt am Beispiel der Hochenergiephysik die Bedeutung von Visualisierungen für die Disziplin, da sie die Grundlage der gesamten wissenschaftlichen Kommunikation überhaupt bilden. Sie prägte dafür den Begriff der „Viskurse“.

Indikatoren, die die Verteilung von Datenpunkten möglichst exakt beschreiben sollen. Das Vertrauen in die Visualisierung ist dabei so groß, dass sie oft als erstes Mittel der Datenanalyse empfohlen wird und sich daran dann statistische Tests anschließen, um den über die Visualisierung gewonnenen Eindruck zu prüfen (Gries 2008). Dies ist akzeptiert, da die Visualisierung auf wohldefinierten und in der Statistik akzeptierten Mess- und Testverfahren beruht und der Transfer der daraus entstandenen Werte in grafische Elemente (sog. „Mapping“) ebenfalls systematisiert und standardisiert ist.

Ebenso stark von Denkstilen durchsetzt sind die Rollen, die in den unterschiedlichen Disziplinen Visualisierungen zugedacht werden. Verfahren des Data Minings, die visuelle Analysemethoden nutzen, sind beispielsweise stark von einem Utilitarismus-Topos durchdrungen, bei dem davon ausgegangen wird, dass die Daten eine Wahrheit (eine „ground truth“, einen „Schatz“) enthalten, der gefunden werden kann (Bubenhofer 2016; Bubenhofer u. a. 2018; vgl. auch den Beitrag von Barbaresi in diesem Band, S. 167). Die Rolle des Tools ist damit klar definiert: diesen Schatz zu heben. Damit kann ein visuelles Analyseinstrument anhand eines „Goldstandards“ evaluiert und seine Güte ausgedrückt werden. In geisteswissenschaftlichen Kontexten herrscht allerdings oft Skepsis gegenüber einer solchen Sicht, da eher von einem Gewebe von Theorie, Methodologie und deren Operationalisierung und Implementierung, bei dem die/der Forscher/in bedeutender Bestandteil ist, ausgegangen wird. Aus dieser Sicht sind deshalb visuelle Analysemethoden eher nicht effektive Analysetools, die den „Informationsüberfluss“ zähmen sollen, sondern eher eine Möglichkeit, reiche Nahrung für Deutungen zu bieten, etwa im Sinne einer „dichten Beschreibung“ (Geertz 1987).

4.2 Visualisierungen als wissenschaftliches Gerät

Genau so, wie Visualisierungen als Zeichen rhetorische Funktionen einnehmen, dienen sie in der Forschung oft als Instrument. Dies gilt insbesondere für explorative Visualisierungen: Mit den Daten kann erst gearbeitet werden, wenn diejenigen Aspekte davon, die von Interesse sind, in grafische Formen überführt wurden. Die Operationen im visuellen Ensemble der grafischen Formen führen dann im besten Fall zu neuen Erkenntnissen, analog der Arbeit mit dem Mikroskop.

Im Fall von algorithmisch erstellten Visualisierungen sind nach den Ausführungen oben zu den diagrammatischen und algorithmischen Perspektiven zwei Aspekte wichtig: 1) Zum wissenschaftlichen Gerät gehört nicht nur die Visualisierung selber, sondern der Computer insgesamt als diagrammatische Maschine. 2) Programmcode, Algorithmen und Software als Bestandteile des wissenschaftlichen Geräts sind kulturell geprägt und machen das Gerät dadurch

zu einem besonders wirkmächtigen Mittel wissenschaftlicher Denkstile. Denn das Metamedium Computer ist fluider und unbestimmter als etwa ein Mikroskop und damit weit stärker kulturellen Praktiken unterworfen, die allerdings häufig den Anwenderinnen und Anwendern von Visualisierungssoftware nicht bewusst sind.

4.3 Kanons und Kulturen

Visualisierungen sind also Ausdruck von wissenschaftlichen Denkstilen. Die Visualisierungspraxis eines Denkkollektivs zeigt sich daran, ob Visualisierungen in der jeweiligen Disziplin akzeptiert sind, welche Typen und welche Ausprägungen üblich sind und zu welchen Zwecken sie verwendet werden. Für jede Disziplin wird in bestimmten Zeiträumen ein bestimmter Bildstil erkennbar sein: „Aus dem Abstand von hundert Jahren wird man die Darstellungen der Nanoforschung auf einen Blick auf zwei oder drei Jahre genau datieren können; in Bezug auf die fraktale Mathematik gilt das gleiche.“ (Bredenkamp u. a. 2008: 41f.) Ebenso existieren in jeder Disziplin „ikonisierte Diagramme“ (vgl. Lauersdorf in diesem Band S. 91), die alternative Visualisierungen behindern.

Insbesondere bei algorithmischen Visualisierungen müssen dabei auch die programmiertechnischen und algorithmischen Grundlagen mitbedacht werden. Visualisierungen, die mit der Javascript-Bibliothek D3 (Bostock u. a. 2011) erstellt wurden, ähneln sich vom Typus her, obwohl diese Bibliothek eine enorme Vielfalt an neuen Visualisierungsformen von Daten auslöste. Die Designprinzipien der Bibliothek sind in eine bestimmte Softwarekultur eingebettet (Open Source, browserbasiert, leicht nutzbar, modular etc.), die u. a. vom Erfinder selber auch explizit verbalisiert wird, etwa im Rahmen der neuen Version 4 der Bibliothek: „Programming interfaces are user interfaces. Or, to put it another way: Programmers are people, too“ (Bostock 2016).

Um aber Innovation im Bereich wissenschaftlicher Visualisierung zu ermöglichen, muss dieser Kanon disziplinärer Visualisierungspraktiken immer wieder hintergangen werden. Die *andere, nicht-kanonisierte* Visualisierung ist notwendig, um Innovation zu ermöglichen. Pflaeging (in diesem Band S. 123) plädiert beispielsweise dafür, gezielt visualisierte Metaphern hinzuzuziehen, um linguistische Theorien zu erklären und so durch eine „Ästhetisierung von Linguistikvermittlung [...] rezipientenseitig zu einer verlängerten und bewussteren Wahrnehmung der musterbrechenden Elemente“ zu gelangen. Relativ pragmatisch machen Keim et al. (2006) auf dieses Problem aufmerksam: „User acceptability is a further challenge; many novel visualization techniques have been presented, yet their widespread deployment has not taken place, primarily due to the users' refusal to change their working routines.“ Als Rezept zur Akzeptanzförderung

wird schlicht und einfach vorgeschlagen, den Benutzerinnen und Benutzern die Vorteile der Visualisierung zu erklären – durchaus ein gangbarer Weg, doch dürfte das alleine nicht reichen, solange der jeweilige Denkstil diese Form noch nicht akzeptiert.

4.4 Listen und Partituren

Im letzten Teil dieses Beitrags möchte ich den Blick nochmals auf die Visualisierungspraxis in der Linguistik lenken. Neben Karten, Graphen/Netzen (etwa zur Darstellung von Kollokationen) oder Bäumen (für syntaktische Strukturen, Sprachfamilien etc.) und Vektoren (visualisiert als Werte in einem Koordinatensystem) gibt es in den Sprachwissenschaften zwei unauffälligere, aber deswegen nicht weniger wirkmächtige Methoden, die ich unter den Schlagworten „Listen“ und „Partituren“ fassen möchte (vgl. zu typischen Visualisierungsformen in der Linguistik auch den Beitrag von Perkuhn und Kupietz in diesem Band, S. 63). An beiden folgenden Beispielen zu Listen und Partituren kann gleichzeitig diskutiert werden, wie sich durch die Transformation oder Formatierung der Sprachdaten in diese Formen der Analysegegenstand verändert.

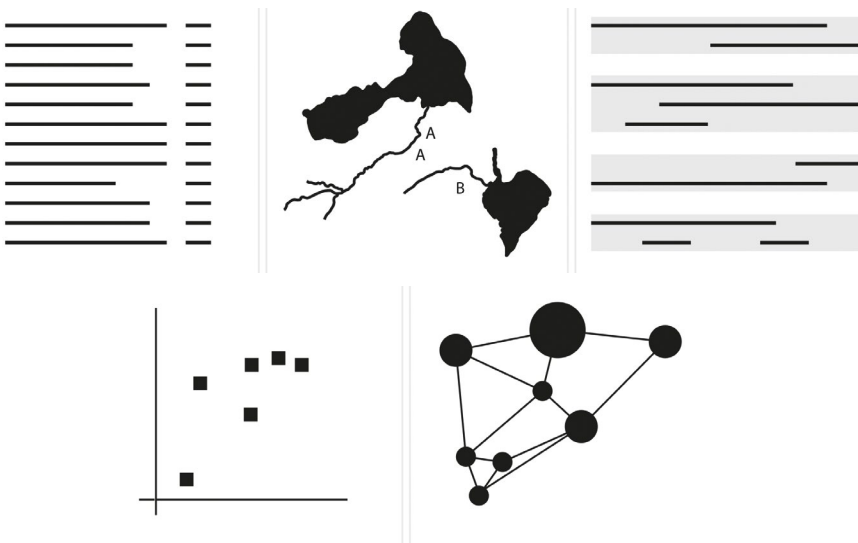


Abb. 2: Fünf in der Linguistik häufig anzutreffende Visualisierungstypen: Liste, Karte, Partitur, Vektoren, Graph/Netz (gerichtet: Baum; ungerichtet: Netz).

Aus diagrammatischer Perspektive ist klar, dass jegliche Strukturierung von Text, beispielsweise in Form von Spiegelpunkten, Aufzählungen o. Ä. bereits diagrammatischen Charakter hat (Steinseifer 2013). Im linguistischen Kontext sind zwei Beispiele naheliegend, bei denen Sprache umstrukturiert wird, um sie analysierbar zu machen: 1) In der Korpuslinguistik ist vor allem die „Key Word in Context“-Darstellung – KWIC – bekannt. Sie wird verwendet, um eine Ergebnismenge von Treffern übersichtlich darzustellen. Diese sehr alte und prinzipiell simple Form eines Index widerspiegelt eine völlig neue Lektüretechnik. 2) In der Gesprächsanalyse dienen verschiedene Notationssysteme, darunter insbesondere die Transkription in die sog. Partiturschreibweise dazu, gesprochene Sprache in ihrer Flüchtigkeit und Gleichzeitigkeit analysierbar zu machen. Beide Beispiele lohnen eine Analyse aus diagrammatischer Sicht.



Abb. 3: Das Bücherrad von Agostino Ramelli (Ramelli 1588: 317).

1588 erfindet Agostino Ramelli ein Bücherrad (Ramelli 1588: 317): Eine hölzerne Konstruktion, auf der gleichzeitig mehrere Bücher aufgeschlagen Platz finden und vor der die Leserin/der Leser Platz findet und bequem am Rad drehen kann, um jederzeit zum nächsten Buch wechseln zu können (Siegel 2009: 28). Zwar blieb die Erfindung ein Plan auf Papier, sie widerspiegelt jedoch einen Paradigmenwechsel humanistischer Lektüretechniken in der Frühen Neuzeit: „Anstelle kontemplativer Versenkung in einen einzigen, durch die Buchdeckel begrenzten Text, bedeutet Lektüre hier einen stetigen Akt des aufeinander Beziehen von ursprünglich distinkten Sinneinheiten.“ (Siegel 2009: 33). Dafür werden neue Methoden notwendig, etwa die Glossierung am Seitenrand oder das Exzerpieren

und die Verwaltung dieser Exzerpte, z. B. in Form des „Liber excerptorum“ von Placcius, bei dem Zettel flexibel in einem Buch eingesetzt werden können (Placcius 1689: 68ff.), oder dem „Scrinium litteratum“, dem Karteikasten, der noch größere Freiheit beim Reorganisieren der Daten erlaubt (Placcius 1689: 121ff.). In der gleichen Zeit entstehen die großen Enzyklopädien zur Systematisierung des Wissens und erweitern damit Lektüre zu einem komplexen Akt, „der im Ganzen die Struktur des gedruckten Fließtextes zu Formularen erweitert, deren hoher Anspruch eine systematische Ordnung der gelehrten Überlieferung ist“ (Siegel 2009: 47).

Die Nähe dieser Lektüertechniken zur Korpuslinguistik und dem modernen Data Mining ist unübersehbar: „Die [...] praktisch geübte und methodisch reflektierte Isolation einzelner Textpartien zum *locus [communis]* bedeutet, die Vielfalt tradierter Texte als einen riesenhaften Speicher zu behandeln, der in sich die Summe des verfügbaren Wissens inkorporiert“ (Siegel 2009: 37). Die Konzeptualisierung der Texte als Speicher geht mit einem neuen Zugriff darauf einher. Genauso bricht die Suche in einem Textkorpus nach sprachlichen Einheiten und die Repräsentation der Ergebnisse als Konkordanz – also die Isolation der Textstellen aus dem jeweiligen Text – mit dem traditionellen hermeneutischen Lektüreerlebnis. Gleichzeitig stellt die Konkordanz aber einen eigenen „locus“ dar, ist also von eigenem Aussagewert und ist die Voraussetzung dafür, daraus eine emergente Struktur ableiten zu können.

Natürlich sind auch Unterschiede zwischen Korpuslinguistik/Data Mining und den neuen Lektüreform in der Frühen Neuzeit sichtbar, was in der digitalen Form der Daten begründet liegt. Das betrifft sowohl die Möglichkeiten der Erstellung der Indizes als auch deren weitere Verarbeitung, bei der in der digitalen Welt die menschliche Leserin / der menschliche Leser an verschiedenen Stellen durch Algorithmen ersetzt werden kann.

Die KWic-Liste wird in der Linguistik (und benachbarten Disziplinen) zuweilen durchaus als Provokation wahrgenommen. Dies ist beispielsweise im Kontext der Diskurslinguistik sichtbar, wo keine Einigkeit über den Nutzen des korpuslinguistischen Zugangs herrscht, gerade auch, weil die bekannteste korpuslinguistische Präsentationsform von Ergebnissen, die Liste von Textstellen, die Texteinheit aufbricht. So wird argumentiert, die Diskurslinguistik arbeite mit dem Begriff des einheitlichen Textes, die Korpuslinguistik dagegen mit Textfragmenten (Leech 2000: 678; Spitzmüller und Warnke 2011: 32). Andererseits kann der korpuslinguistische Zugang auch gerade als passend angesehen werden, um textübergreifende Aussagensysteme zu untersuchen (Bubenhofer und Scharloth 2013: 247). Es handelt sich um eine Art „maschinengeleitete Lesetechnik“ (Scholz und Mattissek 2014: 87):

„Die dekontextualisierte Darstellung erlaubt es den Forschenden, frei vom ‚hermeneutischen Reflex‘, der die Lektüre von Texten und Textpassagen bestimmt, kreativ Ideen zu möglichen diskursiven Zusammenhängen einzelner Korpusteile zu entwickeln, die bei einer subjektiven Lektüre möglicherweise verdeckt blieben.“ (Scholz und Matissek 2014: 87)

Diese dekontextualisierte Darstellung ist in der Korpuslinguistik, wie erwähnt, zwingendes Ergebnis eines quantitativen Zugangs, der eine Lektüre des Einzeltextes je nach Ansicht ersetzen, ergänzen und/oder aber gerade nicht ersetzen kann. Die KWIC-Darstellungen (und viele weitere, avanciertere synthetisierende Darstellungen von quantitativen Ergebnissen) werden damit zum Symbol einer am Sprachgebrauch orientierten Sicht auf Diskurse und so im größeren Kontext zum Gegenstand lebhafter Diskussionen über die verschiedenen Spielarten von Diskursanalyse (Angermüller 2014; Meier u. a. 2014; Niehr 2015).

Ich meine damit zeigen zu können, wie stark eine diagrammatische Form – die Liste als Konkordanz – die Lektüre von Text beeinflusst und zu einem neuen Verständnis von Sprache führt: Ein Verständnis, das geprägt ist vom Interesse an der sprachlichen Form, an repetitiven Segmenten, Mustern und ihrer soziokulturellen Deutung – an Oberfläche und Performanz (Feilke und Linke 2008).

Eine ähnliche Verschränkung von diagrammatischer Form und neuer Sicht auf Sprache ist in der Analyse gesprochener Sprache zu beobachten. Im Rahmen der Dialektologie und der sprachtypologischen Forschung war die Transkription von Äußerungen gesprochener Sprache schon länger ein Thema (Redder 2001), doch erst mit Arbeiten wie jenen von Sacks et al. (1974) entsteht eine neue Perspektive, die auch radikal neue Darstellungsformen bedingt: Die Keimzelle dieser Perspektive liegt wahrscheinlich in der Fokussierung auf den „turn“: Eine Einheit, die Gespräche strukturiert und organisiert und sich nicht mit den klassischen grammatischen Kategorien geschriebener Sprache fassen lässt. Damit einher gehen Mechanismen des Sprecherwechsels, bei denen sich „turns“ nacheinander ablösen, aber oft sich eben auch überlappen. Die Sequenzialität sprachlicher Äußerungen, die bei der geschriebenen Sprache die Normalität ist, wird in der gesprochenen Sprache um die Dimension der Gleichzeitigkeit ergänzt.

Dies bedingt neue diagrammatische Formen der Transkription, mit der die Gleichzeitigkeit mehrerer „turns“ dargestellt werden kann. Es etablierten sich verschiedene Lösungen dafür, besonders erfolgreich ist jedoch die Partiturschreibweise (vgl. Abb. 4). Dabei werden die verschiedenen Sprecher/innen als „Stimmen“ eines Orchesters aufgefasst und die in der Musik übliche Notation der Partitur übernommen, bei der jede Stimme eine eigene Zeile einnimmt und sich übereinander angeordnet die Gleich- und Ungleichzeitigkeit der verschiedenen Stimmen

[142]	f2 jahre de fakto;=und da drüber brauchen wir jetzt gar ned lang diskutieren-	
m2		((klopft auf pult, senkt blick,
?	()	
[143]	f2 (.) weil wenn die komission in ihrer WUNderbaren vielfalt nach 6 jahren beschliesst, dass	
m2	bläst luft aus, blickt sie wieder an))	
[144]	f2 sie alle noch VI:EL lieber und noch VI:EL billiger bei uns durchfahren	
m1	<<all>kann sie gar nicht>	
m2		hehehe
?	(aso)	
[145]	f2 möchten, (.) ä:h dann sind WIR der billige jakob? (.)	
m1	((grinst))	
m2	frau lichtenberger ich schätze sie.	
[146]	f2 <<wehrt äusserungen gestisch ab>na bitte do (redens) mer da jetzt nit rein?> dann (-) sein	
m2	machen wir (kei innerpolitisches parkett da);	((grinst, senkt
[147]	f2 mir der billige jakob von ÖSCHterreich? (-) wir haben nicht einmal eine geringfügige	
m2	blick, lehnt sich zurück))	

Abb. 4: Beispiel für ein Transkript in Partiturschreibweise (aus: Stocker u. a. 2004).

sofort überblicken lässt. Transkriptionsstandards wie GAT (Selting u. a. 1998) oder HIAT (Ehlich und Rehbein 1976, 1979; Rehbein u. a. 2004) definieren die Details der Umsetzung.

Es lohnt sich, einen Blick in die Geschichte der Notation von Musik und dort insbesondere der Partitur zu werfen, um die diagrammatische Innovation dieser Schreibweise, gleichzeitig aber auch die funktionalen Differenzen zwischen der Musikpartitur und der Partiturschreibweise von Gesprächstranskripten herauszuarbeiten. Bei der Notation polyphoner Musik muss unterschieden werden zwischen Ensemble-Musik, bei der mehrere Personen gleichzeitig spielen, und Solo-Musik, bei der eine Person auf dem gleichen Instrument gleichzeitig mehrere Stimmen spielt, beispielsweise die linke und rechte Hand beim Klavierspiel (Apel 1961: XXV; Sachs und Röder 1989). Bereits aus dem 9. Jahrhundert sind Notationen bekannt, die als Vorläufer von Partituren angesehen werden, so z. B. in der *Musica Enchiridis*, einer Art Handbuch, das dazu dienen sollte, das Singen von Gregorianischen Chorälen zu unterrichten (vgl. Abb. 5⁹). Dort ergab sich die Notwendigkeit, die sich hauptsächlich in Quart- oder Quintabständen parallel bewegendes Stimmen untereinander, aber im gleichen Notensystem, abzubilden.

9 Die gesamte Handschrift ist bei der Staatsbibliothek Bamberg digital einsehbar: <http://bsbsbb.bsb.lrz.de/~db/0000/sbbooooo078/images/> (letzter Zugriff: 20. Januar 2018).

51.

tris sempiternus
pa er
Tu fi
li us

tris sempiternus
pa er
Tu fi
li us

Sic enim in infinitum sonorum consequentia
 progreditur: ut ab unoquoque sono locis
 octavis renata ut ita dicam uoce ordo nouus
 emergat. & dierum more octauasit quæ
 prima. prima quæ octaua. Unde & in
 uirgilio. apud elisium orpheus obloquitur
 numeris septem discrimina uocum. quod
 scilicet sonorum ordo disparibus septem
 continuetur uocibus. at in octauis in noua
 mutetur. Et enim sicut denario numero
 qui fuerit additus. intra eum positus

Abb. 5: Beispiel für eine partiturartige Notation aus der Musica Enchiriadis; Handschrift, Kopie Msc.Var.1, fol. 51r, Scolica enchiriadis de arte musica – u. a. musiktheoretische Texte. Werden (?), um 1000. Staatsbibliothek Bamberg. Foto: Gerald Raab.

Damit weist das Notationssystem zwei Achsen auf: Die vertikale Achse gibt das Tonsystem in der sog. Dasia-Notation (später Tonbuchstaben) wieder, während in der horizontalen der Verlauf der Melodien notiert war (Sachs und Röder 1989). Mit der Notation war ein didaktischer Nutzen verbunden, um durch die „Verbindung von Hören und Sehen“ (Sachs und Röder 1989) die Intervall-Bezüge zwischen den Stimmen deutlich zu machen.

Allen Partitur-ähnlichen Notationen bis ins 15. Jahrhundert ist gemein, dass eine exakte vertikale Anordnung nicht angestrebt wurde, da sich erst eine systematische Notation von Tonlängen, Taktstrichen und konventionalisierter vertikaler Anordnung entwickeln musste. Auch danach schienen die ersten Partituren, die nun für jede Stimme ein eigenes Notensystem mit entsprechendem Schlüssel nennen (etwa 1537 bei Lampadius, vgl. Sachs und Röder 1989: 1431), eine nebengeordnete Rolle zu spielen: Sie dienten als Vorlage, um daraus die Einzelstimmen zu extrahieren, wobei bei der Aufführung eine dieser Einzelstimmen auch als Dirigiergrundlage verwendet wurde. Im Verlauf des 16. Jahrhunderts etabliert sich die Partitur allmählich, da sie als Aufzeichnungssystem diente, um „ein (fremdes) Werk genauer prüfen zu können“, und dazu, aus „aufführungspraktischen Zwecken [...] ein Tasteninstrument, in der Regel die Orgel, als Stütze heranzuziehen“ und die dafür nötige Notengrundlage als (Teil-)Partitur zu erstellen (Sachs und Röder 1989: 1429).

Deutlich ist also auch bei musikalischen Partituren, dass sie zunächst dazu dienten, die Polyphonie von Musik sichtbar zu machen, weniger als Vorlage für die musikalische Wiedergabe. Die Erfindung der Partiturschreibweise ermöglicht einen neuen Blick auf den bereits bekannten Gegenstand (das Lied, das Musikstück, das Gespräch), indem die zeitliche Interaktion der Stimmen bzw. Sprecher/innen hervortritt. Die Partitur visualisiert die Gleichzeitigkeit von Gleichwertigem – im Unterschied zu Glossen oder Marginalien, die zwar auch ein diagrammatisches Ausdrucksmittel von Gleichzeitigkeit sind, doch deutlich priorisieren, eben: marginalisieren.

Trotzdem sind auch Differenzen zwischen Musik- und Gesprächspartitur ersichtlich: Die Musikpartitur hat zwar eine ähnliche ordnende Funktion von Stimmen wie Gesprächspartituren, nutzt aber zusätzlich mit dem Notationssystem von Tonhöhen, -längen und Rhythmen ein System, das sich in die Partiturschreibweise problemlos integriert: Die Viertelnote der ersten Stimme wird, ergänzt mit Freiraum, denselben Platz einnehmen wie die gleichzeitig spielenden zwei Achtelnoten der zweiten Stimme – die Taktmarkierungen sind eine zusätzliche Hilfe, um diesen Effekt zu erreichen. Bei Gesprächstranskripten hingegen entspricht die orthografische und typografische Umsetzung des gesprochenen Lautes nicht dessen Länge; visuell kann deshalb ein „turn“ der einen Sprecherin denjenigen des zweiten Sprechers überdauern, ohne es real ebenso gemacht zu haben (Redder 2001: 1048). Moderne Transkriptionsprogramme begeben

diesem Problem durch die Integration einer grafischen Umsetzung des Frequenzspektrums in Form eines Spektrogramms.

Den beiden Anwendungsbereichen gleich ist wiederum die Einsicht, dass die Notationssysteme nicht reichen, den Gegenstand, das Gespräch bzw. die Musik, komplett zu erfassen (vgl. für die Musik z. B. Schneider 1987: 317), sie aber trotzdem unverzichtbar sind, um die Flüchtigkeit des Gegenstands zu fixieren und die Komplexität handhabbar zu machen.

Ähnlich wie die Liste erlaubte die Übernahme der Partiturschreibweise aus der Musik in die Linguistik einen neuen Blick auf Sprache – oder schuf damit einen neuen Analysegegenstand: Das Gespräch. Doch nicht nur in den engen Grenzen der Gesprächslinguistik wirkte diese Darstellungsweise innovativ: Das Prinzip der Annotation von Textdaten beruht in der Darstellung ebenfalls auf dem Partiturprinzip, da die Gleichzeitigkeit von Text und Annotation(en) in der ganzen Komplexität dargestellt werden muss (siehe zu den Schwierigkeiten solcher Darstellungen den Beitrag von Burghardt in diesem Band, S. 315). Bereits die in der Korpuslinguistik übliche Darstellung eines annotierten Textes in einer Mischung aus XML-Auszeichnung und vertikalisierter Spaltendarstellung folgt einer (vertikalen) Partiturschreibweise (vgl. Abb. 6).

5. Fazit

Diagrammatische Formen haben das Potenzial, Denkstile zu verändern und wissenschaftliche Tatsachen zu schaffen. In der Linguistik verändern sie den Gegenstand „Sprache“ auf ihre je eigene Art, wie ich am Beispiel von Listen und Partituren gezeigt habe. Karten, Bäume und Netze wären weitere in der Linguistik gängige Formen, deren Auswirkungen untersucht werden müssten.

Doch welcher Art sind diese Transformationen von „Sprache“? Noch von einer erschöpfenden Systematik weit entfernt, nenne ich im Folgenden sechs grundlegende Transformationstypen, die mir für die Analyse von Sprache besonders relevant zu sein scheinen (aber natürlich auch bei anderen Daten auftauchen):

- Rekontextualisierung: Die Listendarstellung, z. B. von Belegen in einem Korpus als KWIC-Liste, aber auch die Platzierung von sprachlichen Daten auf einer Karte, erzeugen für diese sprachlichen Daten einen neuen Kontext. Weiter oben habe ich im Zusammenhang der Korpuslinguistik von „Dekontextualisierung“ gesprochen – ich meine, es ist angemessener von einer Rekontextualisierung zu sprechen, da durch die neuen Darstellungsweisen eine neue Einheit gebildet wird, die durchaus in einem Kontext steht. Bei der KWIC-Liste beispielsweise formt das Paradigma der einzelnen

```

<s lang="de" n="a2-s87">
Unter APPR unter a2-s87-w1
den ART d a2-s87-w2
zahllosen ADJA zahllos a2-s87-w3
Bergen NN Berg a2-s87-w4
vor APPR vor a2-s87-w5
unsern PPOSAT unser a2-s87-w6
Augen NN Auge a2-s87-w7
ragte VVFIN empor+ragen a2-s87-w8
der PDS d a2-s87-w9
ferne ADV ferne a2-s87-w10
<mountain id="g_38" stid="g23" level="geo">
Mont NE Mont a2-s87-w11
Blanc NE Blanc a2-s87-w12
</mountain>
über APPR über a2-s87-w13
die ART d a2-s87-w14
andern PIS ander a2-s87-w15
empor PTKVZ empor a2-s87-w16
; $. ; a2-s87-w17
</s>
<s lang="de" n="a2-s88">
näher ADJD nah a2-s88-w1
bei APPR bei a2-s88-w2
uns PRF wir a2-s88-w3
thronten VVFIN thronen a2-s88-w4
<mountain id="g_39" stid="s7302510" level="geo">
Schreckhorn NE Schreckhorn a2-s88-w5
</mountain>
, $, , a2-s88-w6
<mountain id="g_40" stid="s7296734" level="geo">
Wetterhorn NE Wetterhorn a2-s88-w7
</mountain>
und KON und a2-s88-w8
<mountain id="g_41" stid="s7308060" level="geo">
Jungfrau NE Jungfrau a2-s88-w9
</mountain>
, $, , a2-s88-w10
dem PRELS d a2-s88-w11
Scheine NN Schein a2-s88-w12
nach APPO nach a2-s88-w13
weniger ADV weniger a2-s88-w14
hoch ADJD hoch a2-s88-w15
als KOKOM als a2-s88-w16
unser PPOSAT unser a2-s88-w17
Standort NN Standort a2-s88-w18
. $. . a2-s88-w19
</s>

```

Abb. 6: Vertikalisierte Text mit XML-Auszeichnungen, Beispiel für die Partiturschreibweise in der Korpuslinguistik.

- Syntagmen einen neuen Kontext. Deutlicher noch bei einem Kollokationsprofil: Diese Liste ist Ausdruck eines statistischen Distributionsverhaltens; die Distribution ist der Kontext, in dem die einzelnen Einträge der Liste gelesen werden müssen.
- Desequenzialisierung: Sie ist manchmal ein Nebeneffekt der Rekontextualisierung. Der springende Punkt vieler Diagramme in der Linguistik ist die Auflösung der der Sprache innewohnenden Sequenzialisierung. Die Wortfrequenzliste rekombiniert den Text zu einer geordneten Liste von Wörtern und ignoriert deren ursprüngliche Sequenz. Nicht jede diagrammatische Transformation von Sprache muss zwingend zu kompletter Desequenzialisierung führen. Die Visualisierung von typischen Narrativen in seriellen Geschichten (Bubenhofers u. a. 2013) führt zu einer Rekontextualisierung (Typizität von Narrativen), ohne die ursprüngliche Sequenz zu zerstören. Allgemeiner könnte Desequenzialisierung als Typus einer Dimensionsreduktion angesehen werden.
 - Dimensionsanreicherung: Alle Formen von Partituren, aber auch Netze, Bäume oder Karten reichern sprachliche Daten um weitere Dimensionen an. Partituren ermöglichen die Darstellung beliebig vieler weiterer Ebenen der Gleichzeitigkeit, Netze stellen Bezüge zu anderen Entitäten her, gerichtete Graphen wie Bäume fügen die Dimension der hierarchischen Gliederung hinzu und Karten bieten eine geografische Anreicherung.
 - Rematerialisierung: Die diagrammatische Transformation überführt sprachliche Daten in eine neue Materialität. Das Kollokationsprofil stellt die statistische Zusammenfassung eines Distributionsverhaltens des entsprechenden Lexems dar und ist damit ein neuer Gegenstand, der z. B. als semantisches Lesartenspektrum des Lexems behandelt werden kann. Mit einer Karte assoziierte sprachliche Einheiten ergeben einen Gegenstand von Sprache „in situ“. Der Kollokationsgraph ergibt den Gegenstand des Bedeutungsgewebes.

Die ersten vier Typen formen unmittelbar sprachliche Daten in eine Form um, die die neue Perspektive auf sie ermöglicht. Daraus ergeben sich zwei für Diagramme allgemein typische Funktionen, Operationalität und das Potenzial für Emergenz:

- Ermöglichung von Operationalität: Erst wenn durch die oben genannten Transformationen das Diagramm entsteht, kann damit operiert werden: Aus einer Karte, auf der Aussprachevarianten markiert sind, können Dialekträume herausgelesen werden; im Kollokationsgraph können Knoten hervorgehoben werden, um deren Position im Netz genauer zu untersuchen. Bei algorithmisch erstellten und digital verfügbaren Visualisierungen

- dienen oft Mittel der interaktiven Beeinflussung dazu, mit dem Diagramm operieren zu können. Allerdings ist digitale Interaktivität keinesfalls notwendig, um Operationalität zu erzeugen, denn diese kann auch mit Stift und Papier oder nur in Gedanken vollführt werden. Und andererseits geht bei digitalen Daten diagrammatische Operationalität über Interaktivität hinaus, da sie auch die zugrunde liegenden Algorithmen berührt.
- **Emergenz-Erzeugung:** Die Visualisierung der Daten ermöglicht im besten Fall, darin ein emergentes Phänomen sichtbar zu machen (vgl. dazu auch Barbaresi Seite 167 in diesem Band). Bei Netzgraphen gibt die generelle Ausprägung des Netzes – etwa ob viele oder wenige Cluster von Knoten sichtbar sind, wie dicht das Netz ist etc. – eine Information über die Art des Netzes. Bei der Betrachtung eines Kollokationsprofils erscheinen auf einer emergenten Ebene „Bedeutungen“ des Lexems, auf einer Dialektkarte sind es „Dialekte“, bei anderen Darstellungen vielleicht „Diskursräume“ oder „sprachliche Muster“, also alles reichlich abstrakte Phänomene, die aus der Einzelbetrachtung von Belegen oder Datenpunkten nicht abgeleitet werden könnten und erst mit der Visualisierung als generelle Form sichtbar werden.

Generell handelt es sich bei diesen diagrammatischen Operationen um Verfahren „rekursive[r] Transkriptivität‘ der Sprache“ (Jäger 2010: 315):

„Symbolische Systeme tendieren dazu, als Gewinn aus der für sie charakteristischen Verfahrensform der *rekursiven Selbstverarbeitung* Eigensinn zu generieren. Dies gilt insbesondere für Sprache, die in paradigmatischer Weise über die Eigenschaft verfügt, sich rekursiv auf sich selbst zurückzubiegen und so die eigene Zeichenverwendung fortlaufend zum Gegenstand weiterer thematisierender, kommentierender, explizierender oder zitierender Zeichenverwendungen zu machen, zum Objekt also selbstbezüglicher semiologischer Operationen, in denen sich das zeigt, was man die ‚rekursive Transkriptivität‘ der Sprache nennen könnte.“ (Jäger 2010: 315)

Grundlegende Fragestellung einer Visual Linguistics ist also, genauer zu verstehen, welche diagrammatischen Operationen in welchen wissenschafts- und technikkulturellen Umgebungen zu welchen Deutungsmöglichkeiten (Semantiken) von Sprache führen.

Die Beiträge in diesem Band sind Zeugnis einer lebendigen Visualisierungspraxis in der Sprachwissenschaft. Die Digitalisierung der Daten und das neu erwachte Interesse im Fach für die Analyse großer Textdatenmengen ermöglicht

Visualisierungsexperimente, die den Kanon gegenwärtig erheblich erweitern und Einfluss auf die herrschenden Denkstile haben. Es scheint mir dabei zwingend, aber auch lohnenswert, gerade aus geisteswissenschaftlicher Sicht den Algorithmus und das Programmieren nicht Spezialist/innen komplett zu überlassen, sondern zumindest die kulturellen Prägungen dieser Praktiken zu verstehen. Ebenso wichtig ist, die Visualisierungspraktiken im Fach nicht als Mittel zum Zweck, sondern als gegenstandskonstituierend zu verstehen, als Praktiken, die sowohl Ausdruck von Denkstilen sind als auch diese mitkonstituieren.

Danksagung: Der vorliegende Beitrag entstand im Rahmen des vom Schweizer Nationalfonds (SNF) geförderten Projektes „Visual Linguistics“.

6. Bibliografie

- Angermüller, Johannes. 2014. „Der‘ oder ‚das‘ Korpus? Perspektiven aus der Sozialforschung.“ In *Diskursforschung. Ein interdisziplinäres Handbuch*. Berlin, Boston: de Gruyter, 604–611.
- Apel, Willi. 1961. *The notation of polyphonic music, 900-1600*. Cambridge, Mass.: Mediaeval Academy of America.
- Bauer, Matthias und Christoph Ernst. 2010. *Diagrammatik: Einführung in ein kultur- und medienwissenschaftliches Forschungsfeld*. Bielefeld: transcript.
- Böhme, Gernot. 2006. „Technical Gadgetry: Technological Development in the Aesthetic Economy.“ In *Thesis Eleven* 86 (1): 54–66, doi: 10.1177/0725513606066240.
- Bostock, Michael, Vadim Ogievetsky und Jeffrey Heer. 2011. „D3: Data-Driven Documents.“ In *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2301–2309.
- Bostock, Mike. 2016. What Makes Software Good? <https://medium.com/@mbostock/what-makes-software-good-943557f8a488> (letzter Zugriff am 24. August 2016).
- Bredenkamp, Horst, Birgit Schneider und Vera Dünkel, Hrsg. 2008. *Das Technische Bild: Kompendium zu einer Stilgeschichte wissenschaftlicher Bilder*. Berlin: Akademie Verlag.
- Bubenhofer, Noah. 2016. „Drei Thesen zu Visualisierungspraktiken in den Digital Humanities.“ *Rechtsgeschichte Legal History – Journal of the Max Planck Institute for European Legal History* (24): 351–355.
- Bubenhofer, Noah, Nicole Müller und Joachim Scharloth. 2013. „Narrative Muster und Diskursanalyse: Ein datengeleiteter Ansatz.“ *Zeitschrift für Semiotik, Methoden der Diskursanalyse* 35 (3–4): 419–444.
- Bubenhofer, Noah, Klaus Rothenhäusler, Katrin Affolter und Danica Pajovic. 2018. „The Linguistic Construction of World – an Example of Visual Analysis and Methodological Challenges.“ In *Quantifying Approaches to Discourse for*

- Social Scientists*, herausgegeben von Ronny Scholz. Basingstoke: Palgrave Macmillan.
- Bubenhofer, Noah, Joachim Scharloth. 2013. „Korpuslinguistische Diskursanalyse: Der Nutzen empirisch-quantitativer Verfahren“. In *Diskurslinguistik im Spannungsfeld von Deskription und Kritik*, herausgegeben von Ingo Warnke, Ulrike Meinhof und Martin Reisigl. Berlin: Akademie-Verlag, 147–168 (Diskursmuster – Discourse Patterns).
- Bühler, Karl. 1934. *Sprachtheorie*. Stuttgart: G. Fischer.
- Chen, Chun-houh, Wolfgang Härdle und Antony Unwin, Hrsg. 2008. *Handbook of data visualization*. Berlin: Springer (Springer Handbooks of Computational statistics).
- Coleman, E. Gabriella. 2012. *Coding Freedom: The Ethics and Aesthetics of Hacking*. Princeton: Princeton University Press.
- Dourish, Paul. 2014. „No SQL: The Shifting Materialities of Database Technology: Computational Culture“. *Computational Culture. A Journal of Software Studies* (4). <http://computationalculture.net/no-sql-the-shifting-materialities-of-database-technology/> (letzter Zugriff am 08. Dezember 2017).
- Eco, Umberto. 2002. *Einführung in die Semiotik*. 9., unveränd. Aufl.. München: Fink.
- Eco, Umberto. 1977. *Zeichen. Einführung in einen Begriff und seine Geschichte*. Frankfurt am Main: Suhrkamp (es).
- Ehlich, Konrad und Jochen Rehbein. 1979. „Erweiterte halbinterpretative Arbeitstranskriptionen (HIAT₂): Intonation.“ *Linguistische Berichte* 59: 51–75.
- Ehlich, Konrad und Jochen Rehbein, Jochen. 1976. „Halbinterpretative Arbeitstranskriptionen (HIAT).“ *Linguistische Berichte* 45: 21–41.
- Feilke, Helmuth und Angelika Linke. 2008. „Oberfläche und Performanz – Zur Einleitung.“ In *Oberfläche und Performanz*, herausgegeben von Helmuth Feilke und Angelika Linke. Berlin: de Gruyter, 3–18.
- Fix, Ulla. 2011. *Denkstile und Sprache. Die Funktion von „Sinn-Sehen“ und „Sinn-Bildern“ für die „Entwicklung einer wissenschaftlichen Tatsache“*. home.uni-leipzig.de/fix/Fleck.pdf.
- Fleck, Ludwik, Sylwia Werner und Claus Zittel, Hrsg. 2011. *Denkstile und Tatsachen: Gesammelte Schriften und Zeugnisse*. Originalausgabe. Berlin: Suhrkamp.
- Fleck, Ludwik, Lothar Schäfer und Thomas Schnelle, Hrsg. 1980. *Entstehung und Entwicklung einer wissenschaftlichen Tatsache: Einführung in die Lehre vom Denkstil und Denkkollektiv*. 10. Aufl. Frankfurt am Main: Suhrkamp.
- Fleck, Ludwik, Lothar Schäfer und Thomas Schnelle, Hrsg. 1983. *Erfahrung und Tatsache: gesammelte Aufsätze*. Frankfurt am Main: Suhrkamp.
- Ford, Paul. 2015. *What Is Code? If You Don't Know, You Need to Read This*. In *Businessweek*. (June 11, 2015), <https://www.bloomberg.com/graphics/2015-paul-ford-what-is-code/> (letzter Zugriff am 18. Januar 2018).

- Fuller, Matthew. 2003. *Behind the blip: essays on the culture of software*. Brooklyn: Autonomedia.
- Geertz, Clifford. 1987. „Dichte Beschreibung. Bemerkungen zu einer deutenden Theorie von Kultur“. In *Dichte Beschreibung. Beiträge zum Verstehen kultureller Systeme*. Frankfurt am Main: Suhrkamp, 7–43 (stw).
- Goffey, Andrew. 2014. „Technology, Logistics and Logic: Rethinking the Problem of Fun in Software“. In *Fun and software: exploring pleasure, paradox, and pain in computing*, herausgegeben von Olga Goriunova. New York: Bloomsbury Academic, 21–40.
- Gries, Stefan Thomas. 2008. *Statistik für Sprachwissenschaftler*. Göttingen: Vandenhoeck & Ruprecht (Studienbücher zur Linguistik).
- Heilmann, Till A. 2012. *Textverarbeitung: Eine Mediengeschichte des Computers als Schreibmaschine*. Bielefeld: Transcript (MedienAnalysen).
- Hörl, Erich. 2008. „Wissen im Zeitalter der Simulation. Metatechnische Reflexionen“. In *Simulation: Präsentationstechnik und Erkenntnisinstrument*, herausgegeben von Andrea Gleiniger und Georg Vrachliotis. Berlin: de Gruyter, 93–106.
- Jäger, Ludwig. 2010. „Intermedialität – Intramedialität – Transkriptivität. Überlegungen zu einigen Prinzipien der kulturellen Semiosis.“ In *Sprache intermedial: Stimme und Schrift, Bild und Ton*, herausgegeben von Arnulf Deppermann und Angelika Linke. Berlin: de Gruyter, 301–324.
- Jäger, Ludwig. 2008. „Transkriptive Verhältnisse. Zur Logik intra- und intermediärer Bezugnahmen in ästhetischen Diskursen.“ In *Transkription und Fassung in der Musik des 20. Jahrhunderts: Beiträge des Kolloquiums in der Akademie der Wissenschaften und der Literatur, Mainz, vom 5. bis 6. März 2004*, herausgegeben von Gabriele Buschmeier, Ulrich Konrad und Albrecht Riethmüller. Stuttgart: Steiner, 103–134.
- Kay, A. und A. Goldberg. 1977. „Personal Dynamic Media.“ *Computer* 10 (3): 31–41. doi: 10.1109/C-M.1977.217672.
- Keim, Daniel A., Jörn Kohlhammer, Geoffrey Ellis und Florian Mansmann. 2010. *Mastering the Information Age – Solving Problems with Visual Analytics*. Goslar: Eurographics Association. <http://diglib.eg.org/handle/10.2312/14803> (letzter Zugriff am 8. Dezember 2017).
- Keim, Daniel A., Florian Mansmann, Jörn Schneidewind und Hartmut Ziegler. 2006. „Challenges in Visual Data Analysis.“ In *Tenth International Conference on Information Visualization (IV 2006)*, 9–16. <https://doi.org/10.1109/IV.2006.31>.
- Kittler, Friedrich. 1989. „Die künstliche Intelligenz des Weltkriegs: Alan Turing.“ In *Arsenale der Seele. Literatur- und Medienanalyse seit 1870*, herausgegeben von Friedrich Kittler und Georg Christoph Tholen. München: Fink, 187–202.

- Kittler, Friedrich. 1993. *Draculas Vermächtnis. Technische Schriften*. Leipzig: Reclam.
- Knorr Cetina, Karin. 2001. „Viskurse‘ der Physik. Konsensbildung und visuelle Darstellung“. In Heintz, Bettina; Huber, Jörg (Hrsg.) *Mit dem Auge denken: Strategien der Sichtbarmachung in wissenschaftlichen und virtuellen Wellen*. Zürich: Wien; New York: Voldemeer; Springer, 305–320 (Theorie:Gestaltung).
- Krämer, Sybille. 2009. „Operative Bildlichkeit. Von der ‚Grammatologie‘ zu einer ‚Diagrammatologie‘?“ In *Logik des Bildlichen. Zur Kritik der ikonischen Vernunft*, herausgegeben von Martina Heßler und Dieter Mersch. Bielefeld: Transcript, 94–123 (Metabasis).
- Krämer, Sybille. 2012a. „Punkt, Strich, Fläche. Von der Schriftbildlichkeit zur Diagrammatik.“ In *Schriftbildlichkeit. Wahrnehmbarkeit, Materialität und Operativität von Notationen*, herausgegeben von Sybille Krämer, Eva Cancik-Kirschbaum und Rainer Totzke. Berlin: Akademie, 79–100.
- Krämer, Sybille. 2012b. „Was ist eigentlich eine Karte? Wie Karten Räume darstellen und warum Ptolemaios zur Gründerfigur wissenschaftlicher Kartografie wird.“ In *Politische Räume in vormodernen Gesellschaften. Gestaltung – Wahrnehmung – Funktion*, herausgegeben von Friederike Fless, Rudolf Haensch, Felix Pirson, Susanne Sievers, Ortwin Dally, Rahden/Westf.: Leidorf, 47–53.
- Kuhn, Thomas S. 1996. *Die Struktur wissenschaftlicher Revolutionen*. 13. Aufl. Frankfurt am Main: Suhrkamp.
- Leech, Geoffrey 2000. „Grammars of Spoken English: New Outcomes of Corpus-Oriented Research.“ *Language Learning* 50 (4): 675–724, doi: 10.1111/0023-8333.00143.
- Levy, Steven. 2010. *Hackers: Heroes of the Computer Revolution – 25th Anniversary Edition*. Sebastopol, CA: O’Reilly and Associates.
- Mackenzie, Adrian. 2006. *Cutting Code: Software And Sociality*. New York: Lang (Digital Formations).
- Manovich, Lev. 2014. „Software is the Message.“ *Journal of Visual Culture*. 13 (1): 79–81, doi: 10.1177/1470412913509459.
- Manovich, Lev. 2002. *The Language of New Media*. Reprint. Cambridge, Mass.: The MIT Press.
- Meier, Stefan, Martin Reisigl und Alexander Ziem. 2014. „Vom (Kon-)Text zum Korpus. Ein diskursanalytisches Kaminesgespräch.“ In *Diskursforschung: Ein interdisziplinäres Handbuch*. Berlin: de Gruyter, 436–464.
- Moles, Abraham. 1959. „Kybernetik, eine Revolution in der Stille.“ In *Epoche Atom und Automation: Enzyklopädie des technischen Jahrhunderts in zehn Bänden; VII: Kybernetik, Elektronik, Automation*. Genf: Lempert, 7.

- Möller, Torger. 2007. „Kritische Anmerkungen zu den Begriffen Denkkollektiv, Denkstil und Denkverkehr – Probleme der heutigen Anschlussfähigkeit an Ludwik Fleck.“ In *Von der wissenschaftlichen Tatsache zur Wissensproduktion: Ludwik Fleck und seine Bedeutung für die Wissenschaft und Praxis*, herausgegeben von Božena Choluj und Jan C. Joerden. Frankfurt am Main: Lang, 397–413.
- Naumann, Carl Ludwig. 1982. „Kartographische Datendarstellung.“ In *Dialektologie. Ein Handbuch zur deutschen und allgemeinen Dialektforschung*. Berlin: de Gruyter, 667–692 (Handbücher zur Sprach- und Kommunikationswissenschaft).
- Niehr, Thomas. 2015. „Die Universität im öffentlichen Sprachgebrauch. Ein Plädoyer für das Zusammenwirken von quantitativen und qualitativen Methoden der Diskursforschung.“ In *Universität, Öffentlichkeit: Festschrift für Jürgen Schiewe*, herausgegeben von Kersten Sven Roth, Jürgen Spitzmüller, Birte Arendt und Jana Kiesendahl. Bremen: Hempen, 134–146.
- Peirce, Charles S., Charles Hartshorne, Paul Weiss und Arthur W. Burks, Hrsg. 1994. *The collected papers of Charles Sanders Peirce*. Charlottesville: InteLex Corp.
- Placcius, Vincentius. 1689. *De arte excerpenti. Vom Gelahrten Buchhalten*. Hamburg: Liebezeit.
- Ramelli, Agostino. 1588. *Le diverse et artificiose machine del capitano Agostino Ramelli: nellequali si contengono varij et industriosi movimenti, degni digrandissima speculatione, per cavarne beneficio infinito in ogni sorte d'operatione: composte in lingua Italiana et Francese*. doi: 10.3931/e-rara-8944.
- Redder, Angelika. 2001. „Aufbau und Gestaltung von Transkriptionssystemen.“ In *Text- und Gesprächslinguistik / Linguistics of Text and Conversation*. Berlin: de Gruyter, 1038–1059 (Handbücher zur Sprach- und Kommunikationswissenschaft).
- Rehbein, Jochen, Thomas Schmidt, Bernd Meyer, Franziska Wazke und Annette Herkenrath. 2004. *Handbuch für das computergestützte Transkribieren nach HIAT*. Hamburg: Universität Hamburg (Arbeiten zur Mehrsprachigkeit, Folge B. Working Papers in Multilingualism, Series B).
- Rheinberger, Hans-Jörg. 1994. „Experimentalsysteme, Epistemische Dinge, Experimentalkulturen: Zu einer Epistemologie des Experiments.“ *Deutsche Zeitschrift für Philosophie* 42 (3): 405–418.
- Sachs, Klaus-Jürgen und Thomas Röder. 1989. „Partitur.“ In *Die Musik in Geschichte und Gegenwart*, herausgegeben von Ludwig Finscher (Die Musik in Geschichte und Gegenwart).
- Sacks, Harvey; Emanuel A. Schegloff und Jefferson, Gail. 1974. „A Simplest Systematics for the Organization of Turn-Taking for Conversation.“ *Language* 50 (4): 696–735, doi: 10.2307/412243.

- Schneider, Albrecht. 1987. „Musik, Sound, Sprache, Schrift: Transkription und Notation in der Vergleichenden Musikwissenschaft und Musikethnologie.“ *Zeitschrift für Semiotik* 9 (3-4): 317-343.
- Scholz, Ronny und Annika Mattissek. 2014. „Zwischen Exzellenz und Bildungstreik Lexikometrie als Methodik zur Ermittlung semantischer Makrostrukturen des Hochschulreformdiskurses.“ In *Diskursforschung: Ein interdisziplinäres Handbuch*. Berlin: de Gruyter, 86-112.
- Scott, Brett. 2015. *How yuppies hacked the hacker ethos*. *Aeon Magazine*. <https://aeon.co/essays/how-yuppies-hacked-the-original-hacker-ethos> (Letzter Zugriff am 10. August 2015).
- Selting, Margret, Peter Auer, Birgit Barden, Jörg Bergmann, Elizabeth Couper-Kuhlen, Susanne Günthner, Christoph Meier, Uta Quasthoff, Peter Schlobinski, Susanne Uhmann. 1998. „Gesprächsanalytisches Transkriptionssystem (GAT).“ *Linguistische Berichte* 173: 91-122.
- Siegel, Steffen. 2009. *Tabula: Figuren der Ordnung um 1600*. Berlin: Akademie-Verlag.
- Spitzmüller, Jürgen und Ingo H. Warnke. 2011. *Diskurslinguistik: eine Einführung in Theorien und Methoden der transtextuellen Sprachanalyse*. Berlin: de Gruyter.
- Steinseifer, Martin. 2013. „Texte sehen – Diagrammatologische Impulse für die Textlinguistik.“ *Zeitschrift für germanistische Linguistik* 41 (1): 8-39.
- Stetter, Christian. 2005. „Bild, Diagramm, Schrift.“ *Schrift. Kulturtechnik zwischen Auge, Hand und Maschine*. München: Fink, 115-136 (Kulturtechnik).
- Stocker, Christa, Daniela Macher; Rebekka Studler, Noah Bubenhofer, Daniel Crvelin, Reto Liniger, Martin Volk. 2004. *Studien-CD Linguistik: multimediale Einführungen und interaktive Übungen zur germanistischen Sprachwissenschaft*. Tübingen: Niemeyer.
- Turing, Alan M. 1936. „On Computable Numbers.“ *Proceedings of the London Mathematical Society, 2nd series* 42 (3-4): 230-265.
- Wall, Larry. 1999. *Perl, the first postmodern computer language*. <http://www.wall.org/~larry/pm.html> (letzter Zugriff am 19. August 2015).