Grundlegendes zu Technik und Semiotik

Über energetische und rechnende Maschinen

Peter Brödner

»Technik ist die Anstrengung,
Anstrengungen zu ersparen.«

José Ortega y Gasset

»Alles aus dem Nichts
zu entwickeln genügt Eins.«

Georg Wilhelm Leibniz

»Macht der Technik – Macht durch Technik«
RLS-Tagung Berlin, 2. März 2024

Prof. Dr.-Ing. Peter **Brödner**

Annäherung an ›**Technik**‹ (**nicht**: ›Technologie‹)

Laut **Aristoteles** (Nikomachische Ethik) ist **Technik** Teil der **praktischen Vernunft**: Ein absichtsvolles **Herstellen-Können nützlicher** Mittel (›Handwerk‹), beruhend auf Fachwissen, Übung & Erfahrung, auch auf **Einsicht** in **Wirkungszusammenhänge** (*technê,* eigentlich eine ›List‹).

Zur praktischen Vernunft gehören ferner

- kluges, situativ angemessenes Handeln (*phronêsis*) und
- intuitive Einsicht (*nous*),

im Unterschied zur **theoretischen Vernunft** als **Erkenntnis** & begrifflich-propositionales **Wissen** über die Welt sowie den **Methoden** ihrer Gewinnung (*epistêmê).*

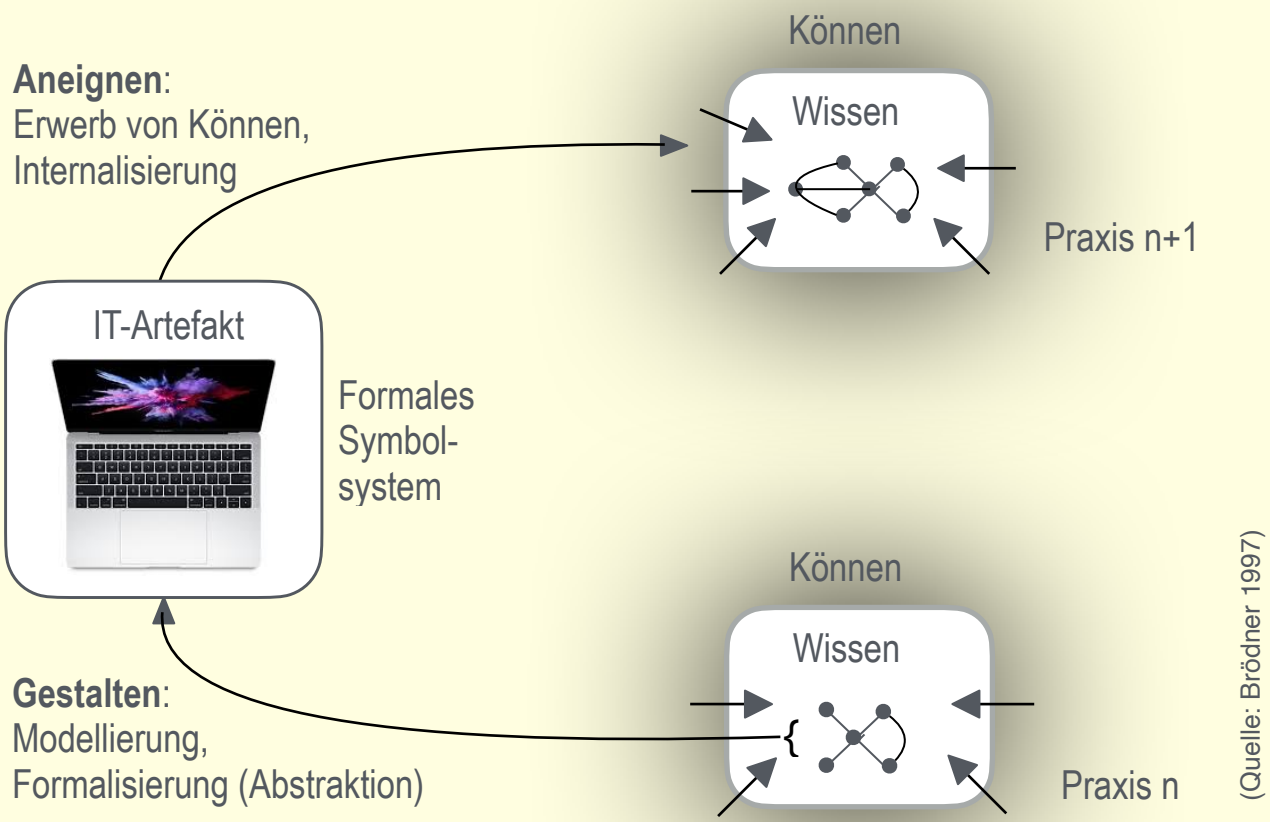
»Unter **Technik** soll jede **feste Kopplung** von **Ursachen** und **Wirkungen** verstanden werden, gleichgültig, ob es sich um Materialien physischer, chemischer oder biologischer Art oder um Symbole handelt wie im Recht oder in der Computertechnologie«
(Luhmann 1995: 112).

Technik ist die Gesamtheit von **Herstellung** & **Gebrauch nutzbarer Wirkungen** als Mitteln für **menschliche Zwecke**. Damit werden ihr nicht nur die **Artefakte** und **Sachsysteme** selbst zugerechnet (wie es oft fälschlich geschieht), sondern auch **Prozesse** zu deren **Entwurf**, **Herstellung** und **Gebrauch**, damit auch die »Systeme zweckrationalen Handelns« (Habermas), durch die Artefakte erst ihren Sinn erhalten und in ihrer Qualität zu beurteilen sind (vgl. Rohpol 1991, VDI 1991).
Begriffsverwirrung: Meist ist ›Technik‹ gemeint, wenn ›Technologie‹ gesagt wird.

Technologie ist die **Lehre** von der **Technik** (→ ›*logos*‹).

Prof. Dr.-Ing. Peter **Brödner**

Technik: **Explikation** von Können & **Aneignung** von Artefakten



(Quelle: Brödner 1997)

Technik (*technê*) bedeutet, **nutzbare Wirkungen** (Funktionen) **zweckgemäß** herzustellen & zu gebrauchen (als Teil praktischer Vernunft; Aristoteles). Deren **Gestaltung** ist Anpassung von Form, Funktion & Handlungskontext im **Spannungsfeld** des **technisch Machbaren** (Formbarkeit von Natur bzw. sozialer Praxis) und des **sozial Wünschenswerten** (Interessen- & Machtabhängigkeit).

Durch Gestaltung & Aneignung technischer Artefakte wird machtvoll in soziale Praxis **interveniert**, daher ist **soziotechnische Gestaltung** von **Arbeitsprozessen** gefordert.

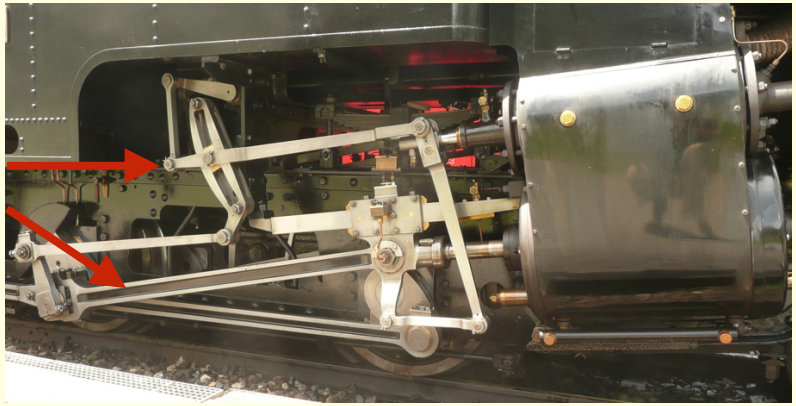
»Wer Technik nur technisch versteht, hat von Technik nichts begriffen.«

Maschinen: **Bündelung** von **Funktionen**

Mit der **Maschinen**entwicklung der industriellen Revolution werden menschliche Verrichtungen aus Produktionsprozessen schrittweise verdrängt, indem sie in Funktionen des **Antriebs** (»Kraftmaschine«, »Motor«), der **Werkzeugführung** (»Arbeitsmaschine«) & der **Steuerung** (Koordination) analytisch zerlegt und zu je eigenständigen **Funktionseinheiten** rekombiniert werden (Babbage 1832), z.B.:

Werkzeugmaschine
= (Haupt- & Vorschub-) Antrieb
+ Werkzeugführung
+ Steuerung (z.B. CNC)

oder **Lokomotive**
= Dampfantrieb
+ Schiebersteuerung



Um **gewünschte Funktionsabläufe** selbsttätig zu gewährleisten, operieren **Steuerungen** statt mit Kräften der Energie- und Stoffumwandlung mit **Signalen** zur **logischen Koordination** der maschinellen Bewegungen (z.B. Lage, Geschwindigkeit, Ventilstellung). Dazu nötige **logische Operationen** müssen **physisch realisiert** werden.

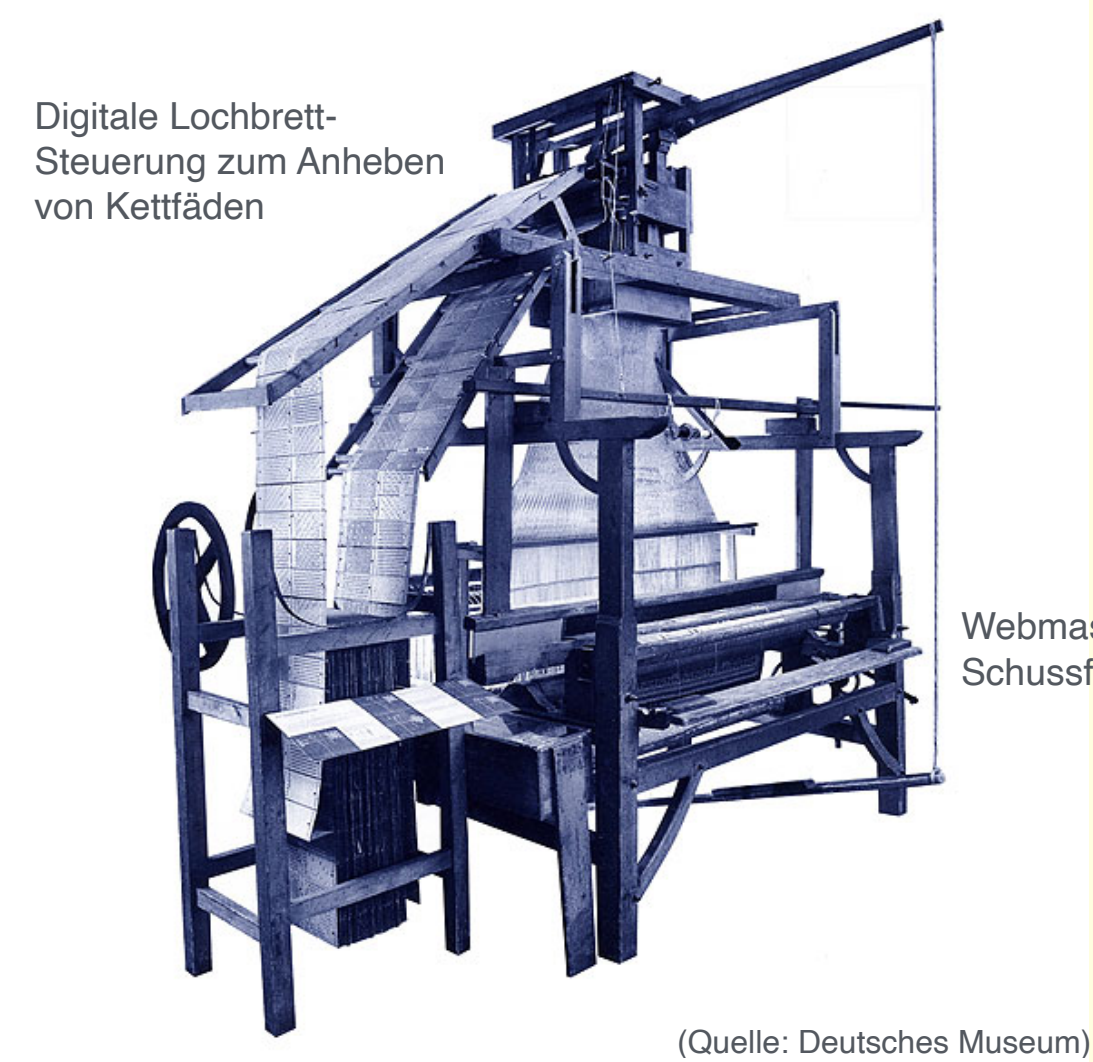
Werden **Bewegungszustände** durch **binäre Signale** erfasst & physisch repräsentiert, lassen sich sämtliche **Steuerungsfunktionen** durch binäre **Schaltsysteme** realisieren.

Eingedenk dessen kann ein **Computer** auch als **universelles Steuerungspotenzial** verstanden werden, das mittels **Programm** in eine **spezielle Steuerung** verwandelt wird (z.B. SPS, CNC; 1960/70er J.). Dessen Entwicklung ist daher von Beginn an mit der industriellen Revolution & Maschinenentwicklung eng verwoben – auch wenn die Bezeichnung »**cyber-physisches System**« (CPS) Neues vortäuscht.

Prof. Dr.-Ing. Peter **Brödner**

Jacquard-Webstuhl: Erste **digital gesteuerte Maschine**

Erste Realisierung 1805 durch *Joseph-Marie Jacquard*.
Jede Lochreihe steuert zur Erzeugung des Webmusters einen Kettfadensatz mit einzeln anhebbaren Fäden.



(Quelle: Deutsches Museum)

Prof. Dr.-Ing. Peter **Brödner**

Musikautomaten: **Mechanische Programmträger**

Verschiedene
Musikinstrumente

Antrieb durch aufziehbares
Gewicht

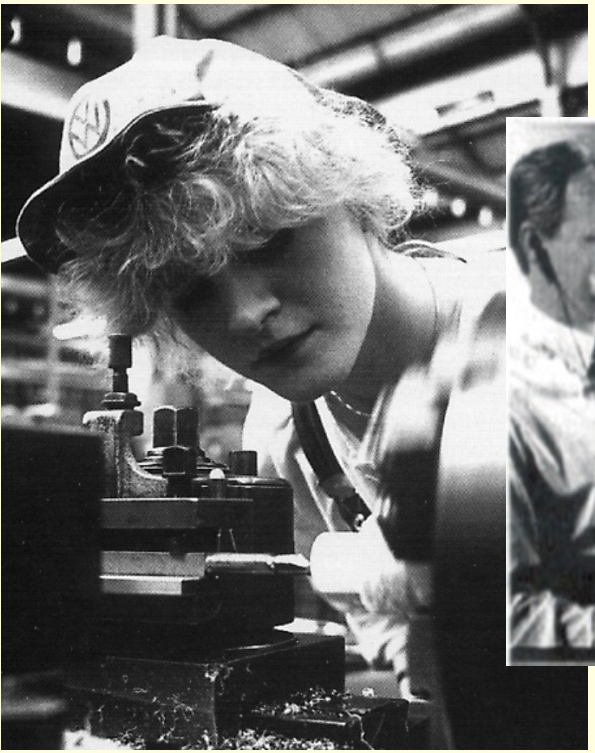
Mechanische **Abtastung**

Stiftwalze als
Programmträger:
Im Lochraster der Walze
gesteckte Stifte bilden das
Programm

Mechanischer Musikautomat im Museum in Bruchsal

Prof. Dr.-Ing. Peter **Brödner**

Maschinenklassen: **Energie-** & **stoffumwandelnd** vs. **semiotisch**



Energie- & stoffumwandelnde Maschinen

- Wirkbereich: Naturprozesse
- Zweck: Nutzung von Naturkräften & -effekten (Elektro-, Thermodynamik, Mechanik)
- Produktivität beruht auf Natureffekt-Nutzung
- Maschineller Kern: Mechanische Funktionen der Krafterzeugung & -übertragung; Maschinisierung körperlicher Arbeit

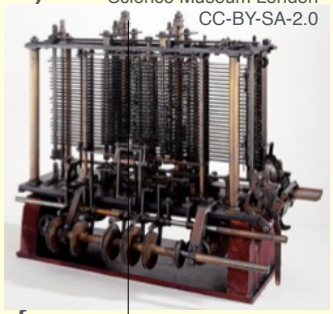
›Computer‹ als semiotische Maschinen

- Wirkbereich: Zeichenprozesse sozialer Praxis
- Zweck: Nutzung berechenbarer Vorgänge zeichen-basierter Interaktion (formale Modellierung, Algorithmik: Was kann wie berechnet werden?)
- Produktivität beruht auf Zeichenprozess-Organisation
- Maschineller Kern: Algorithmische Funktionen der Verarbeitung, Speicherung & Übertragung von Daten; Maschinisierung kognitiver Arbeit

Geschichtliche Meilensteine der ›Computing Science‹

1792-1801: Gaspard **de Prony** entwickelt ein **formularbasiertes Verfahren** zur extrem **arbeitsteiligen Neuberechnung** mathematischer Tafeln im Dezimalsystem.
In diesem frühen großen Softwareprojekt bildet das **Formularschema** der Rechenoperationen die **Urform** eines **Algorithmus** (bis z. 2. WK wurden z.B. V2-Flugbahnen u.v.a. so berechnet).

1805: **Jacquard-Webstuhl**, erste digital gesteuerte Arbeitsmaschine (Lochbretter).



Science Museum London
CC-BY-SA-2.0

1812: Charles **Babbage** konzipiert die »**Difference Engine**« zur Berechnung von Polynomen: $f(x) = a_n\, x^n + \ldots + a_1\, x + a_0$ (1822 prototypisch realisiert).

Um 1830: Charles **Babbage** entwirft & programmiert die »**Analytical Engine**«; sie nimmt die von-Neumann-Architektur programmierbarer Universalrechner (RW/Speicher/Steuerung) vorweg (scheitert aber an der Mechanik).

1847: Die de **Morganschen** Gesetze ($\neg(a \wedge b) = \neg a \vee \neg b$ und $\neg(a \vee b) = \neg a \wedge \neg b$) aufnehmend publiziert George **Boole** einen Logikkalkül (um 1888 von **Peano** als ›**Boolesche Algebra**‹ axiomatisiert); er bildet das logisch-funktionale Fundament für **binäre Schaltsysteme** (heutige Computer-Hardware).

1860-80: C.S. **Peirce** entwickelt erstmals einen Prädikatenkalkül 1. Stufe, arbeitet an »logischen Maschinen« und formuliert eine **triadische Zeichentheorie** (ohne die allg. Computereinsatz nicht erklärbar ist; äquivalent: Freges »Begriffsschrift« 1879).

1903 Mathematik-**Grundlagenkrise**: Anfangs ›naiver‹ Mengenbegriff, Russell-Antinomie, Hilbert-Progamm.

1931: Kurt **Gödel** beweist die **Unvollständigkeit** formaler Systeme wie das der *principia mathematica* von B. Russell & A.N. Whitehead.

1936: Alan **Turing** publiziert die Idee der ›Turingmaschine‹, definiert damit formal die Begriffe **Algorithmus** & berechenbare Funktion (äquivalent: λ-Kalkül (Church)).

1941/45: Konrad **Zuse** entwickelt die **Z3** (Relais) & den **Plankalkül** (erste algorithm. Sprache).

Prof. Dr.-Ing. Peter **Brödner**

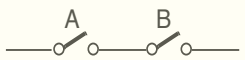
Binäres Schaltsystem zur **Addition zweier Dualzahlen**

Elementare **logische Operationen** werden durch steuerbare Schaltnetze **physisch verkörpert**:

UND-Verknüpfung (Konjunktion)

A	B	A&B
0	0	0
0	1	0
1	0	0
1	1	1

Wahrheitstafel



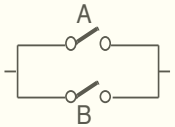
Schaltplan

Symbol

ODER-Verknüpfung (Disjunktion)

A	B	AvB
0	0	0
0	1	1
1	0	1
1	1	1

Wahrheitstafel



Schaltplan

Symbol

Negation

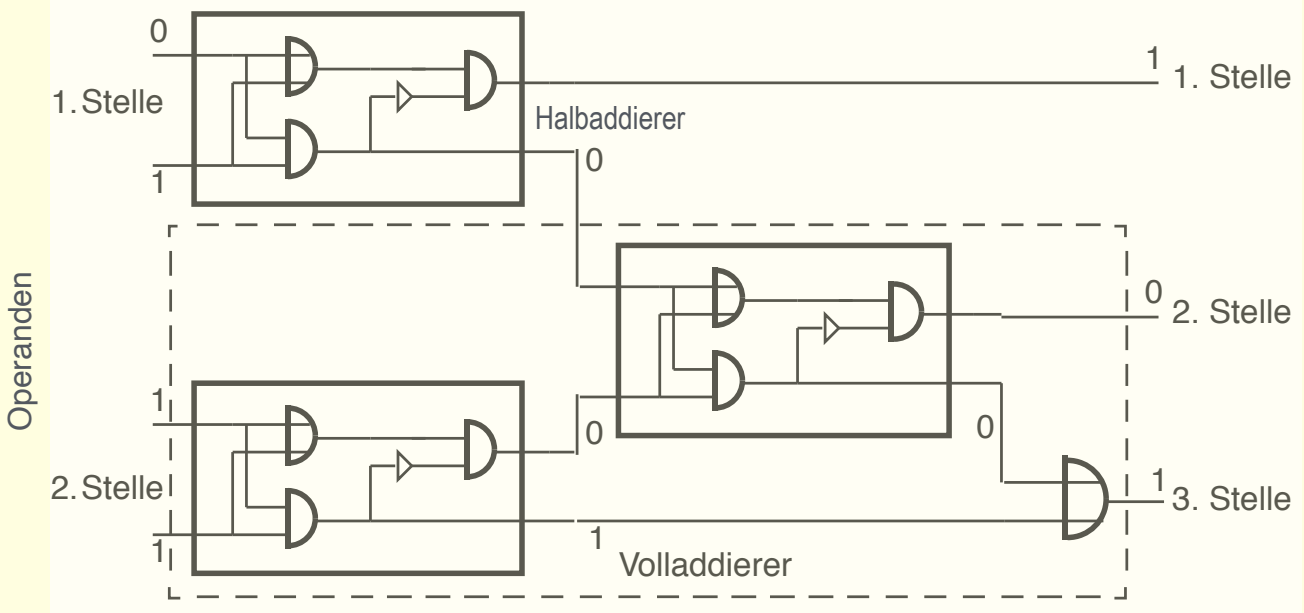
A	¬A
0	1
1	0

Wahrheitstafel

Symbol

Ein binäres **Schaltsystem** aus diesen ›Logikgattern‹, das während eines **Takts** zwei n-stellige **Dualzahlen** korrekt **addiert**, besteht aus insgesamt einem Halbaddierer und (n-1) Volladdierern:

Beispiel (2 + 3): $10 + 11 = 101 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5$



(Quelle: Brödner 1997)

Prof. Dr.-Ing. Peter **Brödner**

Minimales Rechnermodell: ›Random Access Machine‹

Die ›**Random Access Machine**‹ ist als **minimales Modell** maschinellen Rechnens **gleichmächtig** wie die **Turingmaschine** und setzt wie diese beliebig große Daten- & Programmspeicher für natürliche Zahlen einschließlich der Null voraus – als Basis für arithmetische Operationen mit beliebigen, auch negativen oder Gleitkommazahlen.

Gemäß Rekursionstheorie **genügen** zur Ausführung **beliebiger Algorithmen** auf dieser abstrakten Maschine die folgenden vier **elementaren Anweisungen**:

$x := 0$	Inhalt einer Speicherzelle löschen,
$x := x + 1$	Inhalt einer Speicherzelle um 1 erhöhen (›Nachfolger‹ bilden),
$x := x - 1$	Inhalt einer Speichezelle um 1 erniedrigen (›Vorgänger‹ bilden),
$\text{while } x \neq y \text{ do } \ldots \text{ end}$	Iteration: Die While-Schleife wird solange durchlaufen bis $x = y$.

Beispiele für weitere, darauf **aufbauende Anweisungen**:

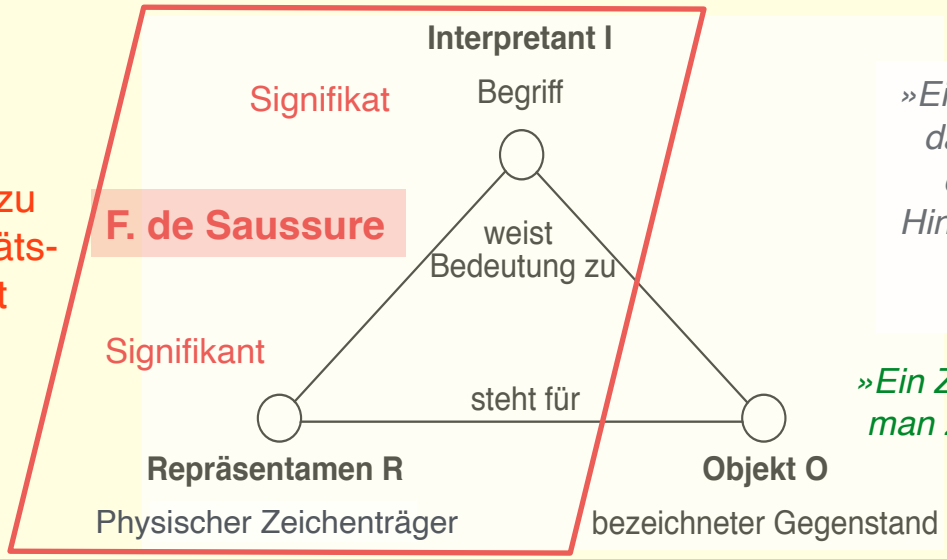
- Laden einer Speicherzelle (›Register‹) mit einer Konstanten n $(x := \mathrm{n})$:
  $x := 0;\ x := x + 1;\ \ldots;\ x := x + 1$ (x wird n-mal um 1 erhöht),
- Zuweisen einer Variablen y an eine Variable x $(x := y)$:
  $x := 0;\ \text{while } x \neq y \text{ do } x := x + 1 \text{ end}$,
- Erhöhen einer Variablen x um eine Variable y $(x := x + y)$ mittels der Hilfsvariablen h:
  $h := 0;\ \text{while } h \neq y \text{ do } x := x + 1;\ h := h + 1 \text{ end}$,
- Addition $(z := x + y)$ mittels Variablenzuweisung $(z := x)$ und -erhöhung $(z := z + y)$,
- Boolesche & Relationsausdrücke lassen sich in arithmetische transformieren mit 1 für *true* und 0 für *false*.

(Quelle: Rechenberg 1991)

Prof. Dr.-Ing. Peter **Brödner**

Zeichen

Führt zu Realitäts-verlust



»Ein Zeichen ist etwas, das für jemanden in einer bestimmten Hinsicht oder Fähigkeit für etwas steht.« (C.S. Peirce)

»Ein Zeichen ist alles …, was man zum Lügen verwenden kann.« (U. Eco)

Ein **Zeichen** ist eine **Relation** zwischen **drei Entitäten**:

(1) dem **Repräsentamen** R, dem physischen Zeichenträger oder -körper (ein als Zeichen gedeuteter Gegenstand oder Vorgang, ›Alphabetzeichen‹),

(2) dem **Objekt** O, dem bezeichneten (ggf. gedachten) Gegenstand oder Vorgang,

(3) dem **Interpretanten** I, der sozial konstruierten Bedeutung, die ein Interpret dem Paar (R, O) **situativ** zuschreibt (Begriff).

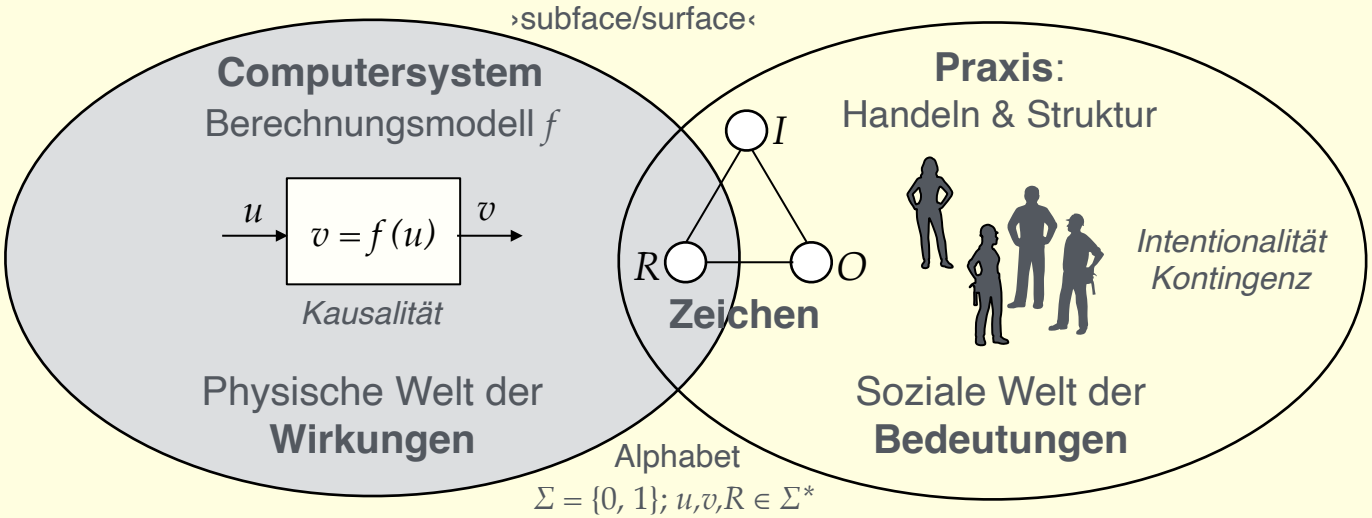
Ein **Zeichen** ist mithin die **3-stellige Relation** $((R - O) - I)$. Der Begriff ist **rekursiv**: Der Interpretant ist selbst ein Zeichen, das bezeichnet & interpretiert werden kann.

- **Syntaktik** beschreibt die auf R reduzierten Aspekte von Zeichenprozessen (**womit** wird bezeichnet? → **Daten** in Form bedeutungsloser Signale);
- **Semantik** beschreibt deren auf die Relation $(R - O)$ reduzierten Aspekte (**was** wird bezeichnet? ›**Bezeichnung**‹; → **Information** i.S. von ›steht für‹);
- **Pragmatik** beschreibt die sozio-kulturellen Aspekte von Zeichenprozessen (**wie** wird Bezeichnetes in einem Handlungskontext gedeutet? → begriffl. **Wissen**).

(Quellen: Peirce 1983; vgl. auch Eco 1976, Morris 1977, Nake 1998, 2001, FRISCO Report IFIP 1998)

Prof. Dr.-Ing. Peter **Brödner**

Zeichen: Logische **Verbindung** von **Physik** & **Semantik**



Software ← **Modellierung**

In perspektivischer Reduktion & **Abstraktion** werden Vorgänge gewünschter zeichenbasierter **sozialer Praxis** formal **modelliert** & in **Programme** als formaler Darstellung von **Berechnungsverfahren** überführt: $v = f(u)$; $u,v \in \Sigma^*$.

Durch **Selbst- & Fremdbeobachtung** mittels Sprache (Text, Grafik) vereinbarte, teils deskriptive, teils präskriptive **Modelle** gewünschter **sozialer Praktiken** liefern die Basis zur Konstruktion von **Software**.

Programme werden per **Compiler** automatisch in ausführbaren **Maschinencode** übersetzt. Während ihrer Laufzeit existiert die Maschine ›Computer‹ nur noch als **reine Hardware**, als Abfolge **bedeutungsloser Signalzustände** binärer Schaltsysteme.

Vom Computer für Eingaben per Programm erzeugte **Ergebnisse** lassen sich durch **Aneignung** & **Kenntnis** der Funktionalität des Berechnungsmodells f im Kontext des Gebrauchs **interpretieren**. Dadurch wird soziale Praxis insgesamt **restrukturiert**.

Signale → **Interpretation**

Prof. Dr.-Ing. Peter **Brödner**

Musterbeispiel **ignoranter IT-Systemeinführung**

Für den Münsteraner »**Kita-Navigator**« wurde im Feb. 2022 ein Algorithmus installiert, der jedem Kind ein Platzangebot zuordnet. Alsbald fiel auf, dass Geschwistern nicht mehr dieselbe Kita zugewiesen wurde, obwohl dies zuvor üblich war. Aufgrund von Beschwerden – auch darüber dass die angebotenen Betreuungsstunden nicht den angefragten entsprachen –, stoppte das Jugendamt das System. Es zeigte sich nicht nur vom massiven Protest, sondern auch über den gravierenden vermeintlichen »**Systemfehler**« **überrascht**.

Der dabei verwendete **Gale-Shapley-Matching-Algorithmus** (1962) ist aber bekanntermaßen nicht imstande, **Randbedingungen** wie die genannten zu berücksichtigen. **Ursache** des Desasters ist **nicht** ein »**Systemfehler**«, sondern die **ignorante** und **dilettantische Vorgehensweise** der Verantwortlichen, die alle seit spätestens der 1990er Jahre in der Praxis etablierten **Standards der Softwareentwicklung missachten**: ein **partizipativ** und **evolutionär** angelegtes **Vorgehen** bei der funktionalen Systemspezifikation sowie die sorgfältige **Analyse** und **Modellierung** (sprachlich und formal) der Vergabeprozesse.

Diese Ignoranz wurzelt darin, dass weder die Grundlagen der **Funktionsweise** von Computersystemen noch die **Bedingungen** für deren **gebrauchstaugliche Gestaltung** hinreichend **verstanden** werden.

(Quelle: Algorithm Watch 28.03.2023)

Prof. Dr.-Ing. Peter **Brödner**

›Maschinelles Lernen (ML)‹: **Trick** & **Täuschung**

Der ›ML‹-**Zauber-Trick** und Etikettenschwindel:

Die aufwendigen Mühen der **Modellierung** & **Formalisierung** (f-Bestimmung) werden mit Methoden ›maschinellen Lernens (ML)‹ **umgangen**, indem aufgrund von Erfahrung, durch theorieloses Probieren (»Bricolage«), eine Funktion f mit zunächst noch unbestimmten **Parametern** einfach **angenommen** wird, deren Werte an vorhandene Daten **angepasst** werden (**Funktionsapproximation**).

Seit langem **bekannte Verfahren** dafür sind beispielsweise:

- lineare Regression (geht zurück auf Gauß ~ 1820),
- logistische Regression (für Funktionen mit nur 2 Werten; geht zurück auf Verhulst ~ 1830/40),
- verschiedene Varianten von Entscheidungsbaumverfahren (Quinlan u.a.,1960/80er Jahre).

Neuerdings gilt sog. »**Deep Learning**« als neue ›**KI-Wunderwaffe**‹: Sie nutzt **mehrschichtige** künstliche neuronale Netze (**KNN**) als Funktionsträger und erfordert extrem **gesteigerte Leistungsfähigkeit** spezieller Hardware (›Tensoroperationen‹).

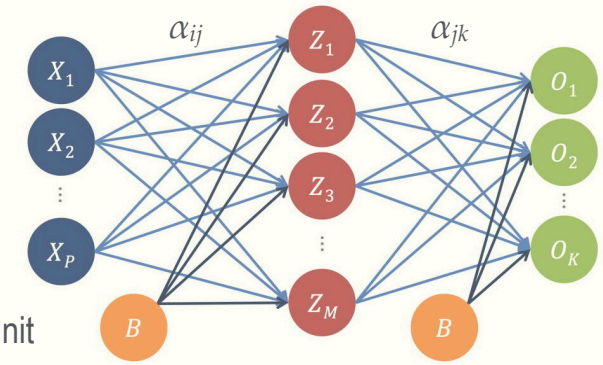
Diese Vorgehensweise wirft aber sogleich **neue** schwerwiegende **Probleme** auf:

- **mangelhafte Daten** methodisch unkontrollierter Herkunft & unbekannter Qualität, die aber den Ergebnisnutzen entscheidend bestimmen: »garbage in – garbage out«,
- nur **wahrscheinliche**, daher prinzipiell **unsichere** Ergebnisse,
- **undurchschaubares** ›Black-Box‹-Verhalten als Haupthindernis im Gebrauch.

Mit üblichem ›**Lernen**‹ hat das nichts gemein, verführt aber zu der **irrigen** Vorstellung ›**lernender Systeme**‹.

Prof. Dr.-Ing. Peter **Brödner**

›Deep Learning‹: Mathematische Modellierung von KNN



Bias Unit

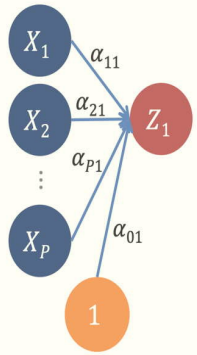
- Bias Units zur Modellierung der Konstanten bzw. Schwellenwerte des linearen Modells
- Input der Bias Units ist $+1$

(Quelle: Uni Köln)

Die **KNN-Struktur** (Anzahl der Schichten & Knoten) ist willkürlich gewählt. Sie wird je nach Aufgabe erfahrungsbasiert durch **theorieloses Probieren** festgelegt (sog. ›**Bricolage**‹). Vorerst noch unbestimmte Parameter α_{ij} werden durch bestmögliche **Anpassung** an gegebene **Kontextdaten** berechnet (sog. ›**training**‹).

Mathematisches Modell

Unit Betrachtung



$$\alpha_1^{*} \left\{ \alpha_1 \left\{ \begin{bmatrix} \alpha_{01} \\ \alpha_{11} \\ \vdots \\ \\ \alpha_{P1} \end{bmatrix}^T \right. \right. \begin{bmatrix} 1 \\ X_1 \\ X_2 \\ \vdots \\ X_P \end{bmatrix} = \boxed{\hat{Z}_1}$$

- Anwendung der **gewichteten Summe** auf Ausgabe der Input Units zur Ermittlung der **Netzeingabe** $\hat{Z}_1$: $\hat{Z}_1 = {\alpha_1^{*}}^T X^{*} = \alpha_{01} + {\alpha_1}^T X$
- Anwendung einer nicht-linearen **Aktivierungsfunktion** auf $\hat{Z}_1$ zur Ermittlung der Ausgabe Z_1: $Z_1 = \sigma(\hat{Z}_1)$

Mathematisches Modell einer Netzschicht

Matrixschreibweise mit Bias Unit

$$\begin{bmatrix} \alpha_{01} & \alpha_{02} & \cdots & \alpha_{0M} \\ \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1M} \\ \vdots & \ddots & & \\ & & & \\ \alpha_{P1} & \cdots & & \alpha_{PM} \end{bmatrix}^T \begin{bmatrix} 1 \\ X_1 \\ X_2 \\ \vdots \\ X_P \end{bmatrix} = \begin{bmatrix} \hat{Z}_1 \\ \hat{Z}_2 \\ \vdots \\ \hat{Z}_M \end{bmatrix}$$

$$\boxed{Z = \sigma(\alpha_0 + \alpha^T X)}$$

Prof. Dr.-Ing. Peter **Brödner**

Sprachverarbeitung: **Word Vector Embedding** (word2vector)

V: **Vokabulargröße** (Anzahl Worte w_j, z.B. 1.000)

N: Größe der **hidden layer** ($N < V$)

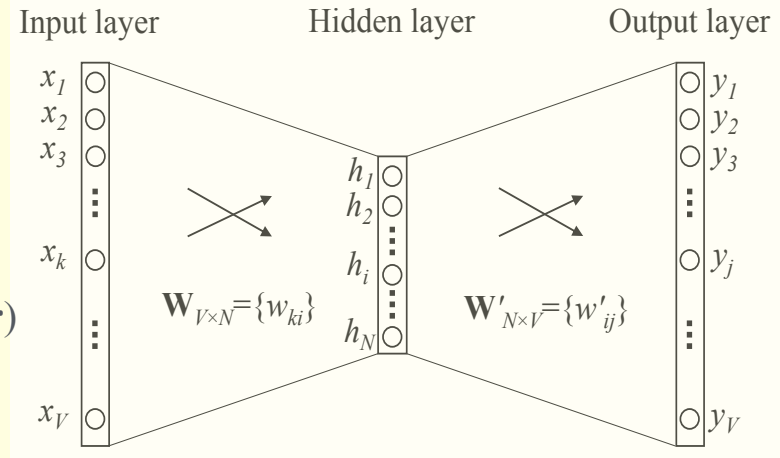
Eingabeschicht: $x_k = 1$ (k-tes Wort), $x_j = 0$ für $j \neq k$

Damit wird in der **hidden layer** jedes Wort w_k durch die k-te Zeile von **W** als Vektor repräsentiert:

$$\mathbf{h} = \mathbf{W}^T\mathbf{x} = \mathbf{W}^T_{(k,\cdot)} := \mathbf{v}^T_{w_I} \quad \text{(als } w\text{-\textbf{Eingabevektor}})$$

Mit der Matrix **W'** = $\{w'_{ij}\}$ wird zudem für jedes Wort w_j ein Zahlenwert u_j berechnet:

$$u_j = \mathbf{v}'^{\,T}_{w_j}\mathbf{h} \qquad \text{(mit } \mathbf{v}'_{wj} \textit{ als } w\text{-\textbf{Ausgabevektor}})$$



Simples Ein-Wort-Kontext-Modell (CBOW)

Damit werden in der **Ausgabeschicht** für jedes Wort w_j unter Verwendung der **Softmax-Funktion** (ein übliches log-lineares Klassifikationsmodell) die **bedingten Wahrscheinlichkeiten** berechnet:

$$p(w_j|w_I) = y_j = \frac{\exp(u_j)}{\sum_{j'=1}^{V}\exp(u_{j'})} \quad \text{bzw.} \quad p(w_j|w_I) = \frac{\exp\left(\mathbf{v}'^{\,T}_{w_j}\mathbf{v}_{w_I}\right)}{\sum_{j'=1}^{V}\exp\left(\mathbf{v}'^{\,T}_{w_{j'}}\mathbf{v}_{w_I}\right)} = y_j$$

(großer Rechenaufwand)

Aus der Forderung nach **Maximierung** dieser bedingten Wahrscheinlichkeiten für alle $j \in \{1,\ldots,V\}$ ergibt sich mit der **Verlustfunktion** $E = -\log p(w_O|w_I)$ & der ›**Lernrate**‹ η die **Korrektur** der **Gewichte** w'_{ij} (zwischen hidden & output layer) zu: sowie w_{ki} (zwischen input & hidden layer) zu:

$$\mathbf{v}'^{\,(\text{new})}_{w_j} = \mathbf{v}'^{\,(\text{old})}_{w_j} - \eta \cdot e_j \cdot \mathbf{h}$$

›Backpropagation‹

$$\mathbf{v}^{(\text{new})}_{w_I} = \mathbf{v}^{(\text{old})}_{w_I} - \eta \mathrm{E}\mathrm{H}^T$$

Der N-dimensionale Vektor EH^T (als gewichtete Summe der Ausgabevektoren $\mathbf{v}_{wI}$ der Worte des Vokabulars) fügt dem Eingabevektor des Kontextworts gewisse **Anteile** jedes **Ausgabevektors** des Vokabulars hinzu & bildet so **von Bedeutung abstrahierte**, stochastische **Wortbeziehungen** ab.

(Source: Xin, R. (2016): word2vec Parameter Learning Explained, arXiv: 1411.2738v4)

https://ronxin.github.io/wevi/

Prof. Dr.-Ing. Peter **Brödner**

Große Sprachmodelle: »**Stochastische Papageien**«

Große Sprachmodelle mit Komponenten wie Word Vector Embedding, LSTM oder vortrainierten Transformern (»GPT«) operieren allein mit **bedingten Wahrscheinlichkeiten** als stochastischen Beziehungen zwischen **Repräsentanten** für Wort(teil)e & Satzzeichen (sog. ›**Token**‹). So wird von jeder **Bedeutung abstrahiert**, indem gemäß triadischem Zeichenbegriff **Formen** sprachlichen Ausdrucks als **Expression** e von deren intentionaler **Interpretation** i **getrennt** werden, durch die soziale Akteure der Expression im situativen Kontext Bedeutung zuweisen. Bedeutung (B) ist eben keine Eigenschaft von Zeichenträgern, sondern wird mit diesen nur in Sprechakten zum Ausdruck gebracht und durch Interpretation erschlossen.
Formal: Sprechakte erzeugen Paare $(e, i) \in B \subseteq E \times I$, Transformer operieren nur mit $e \in E$.

Große Sprachmodelle mit sehr vielen Parametern ($> 175 \cdot 10^9$) werden mittels umfangreicher Textsammlungen trainiert, um für Wortfolgen bedingte Wahrscheinlichkeiten zwischen den ›Token‹ mittels **Softmax**-Funktion zu berechnen. Aufgrund solch stochastischer Beziehungen lassen sich auch relativ kurze Wortfolgen **syntaktisch korrekt** & scheinbar **kohärent fortsetzen**, vorgelegte Sprachausdrücke (sog. »**Prompts**«) quasi ›extrapolieren‹, damit **Bedeutung** aber nur **vortäuschen**. Sprachmodelle sind folglich »**stochastische Papageien**«, die in großen Textsammlungen enthaltene **Wendungen** & **Floskeln** ungeachtet ihrer Bedeutung einfach nachplappern, damit vorherrschende **Sprachgewohnheiten nachahmen** – ohne zu ›wissen‹, wovon dabei die Rede ist.

Damit werden zudem alle in großen Textsammlungen kodierten **vorurteilsbedingten Verzerrungen** reproduziert oder gar mangels Reflexion verstärkt. Die bloße Fortsetzung gegebener Texte kann auch **falsche Aussagen** enthalten, die in Ursprungstexten nicht vorkommen (sog. »Halluzination«). Dagegen hilft die schiere Größe der Modelle wenig.

(Quellen: Bender et al.(2021): On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? ACM Conference FAccT ’21; Marcus, G.: GPT-3 Bloviator, in MIT Technology Review 22.08.2020)

»**Stochastische Papageien**«: programmgeneriertes **Blendwerk**

GPT generieren Texte (Bilder, Noten), die **Sinn**, **Bedeutung** oder **Lüge** nur **vortäuschen**. Tatsächlich erzeugen sie nur syntaktisch korrekte, ihrer **Bedeutung beraubte Zeichenträger-**Ketten, die als »**stochastische Papageien**« & ohne Autor Allgemeinplätze reproduzieren.

On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜

Emily M. Bender*
ebender@uw.edu
University of Washington
Seattle, WA, USA

Timnit Gebru*
timnit@blackinai.org
Black in AI
Palo Alto, CA, USA

Angelina McMillan-Major
aymm@uw.edu
University of Washington
Seattle, WA, USA

ABSTRACT

The past 3 years of work in NLP have been characterized by the development and deployment of ever larger language models, especially for English. BERT, its variants, GPT-2/3, and others, most recently Switch-C, have pushed the boundaries of the possible both through architectural innovations and through sheer size. Using these pretrained models and the methodology of fine-tuning them for specific tasks, researchers have extended the state of the art on a wide array of tasks as measured by leaderboards on specific benchmarks for English. In this paper, we take a step back and ask: How big is too big? What are the possible risks associated with this technology and what paths are available for mitigating those risks? We provide recommendations including weighing the environmental and financial costs first, investing resources into curating and carefully documenting datasets rather than ingesting everything on the web, carrying out pre-development exercises evaluating how the planned approach fits into research and development goals and supports stakeholder values, and encouraging research directions beyond ever larger language models.

CCS CONCEPTS

• **Computing methodologies → Natural language processing.**

ACM Reference Format:
Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜. In *Conference on Fairness, Accountability, and Transparency (FAccT '21), March 3–10, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3442188.3445922

Timnit Gebru was a well-known scholar in the AI-ethics community long before she got fired by Google in December 2020 – but that messy and dramatic incident brought a new level of attention to her work. Google apparently exiled Gebru from its AI ethics team (and subsequently fired the other team leader) in response to a paper about the dangers of the large language models that have become so important to the world's biggest technology companies. The episode created a firestorm in the AI field.

(Source: IEEE Spectrum 11.07.23)

Industrie 4.0: **Interessengeleitete illusionäre Vision**

Foto:Acatech/Steffen Weigelt

Das ›Trio Infernale‹ der Industrie 4.0 Initiative 2011:
Wahlster (DFKI) – Kagermann (acatech) – Lukas (BMBF)

Diese **Dreieinigkeit** spiegelt **dominante Interessen** wider:

Forschung: Scheinbar Neues erforschen, viel Fördergeld, viele ›esoterische‹ Dissertationen

Wirtschaft: Profite für CPS-Realisierer, Breitbandprovider, Cloudservices, Datensammler

Politik: Profilierung als zukunftsweisende ›Modernisierer‹ (freilich von wenig Erfolg gekrönt).

Ruhmloser **Sturz** der **›KI‹-Propaganda**-Leuchttürme

Der in den letzten beiden Dekaden erneut aufgeflammte **›KI‹-Hype** wird durch scheinbar verblüffend erfolgreiche **Leuchtturmprojekte** immer wieder neu **befeuert**, obgleich sie nicht lange danach **ruhmlos** vom **Sockel gestürzt** werden:

- Das von **IBM** jahrelang entwickelte, 1996 eingesetzte Schach-Weltmeisterprogramm »**Deep Blue**« wurde nach dem Sieg über **Kasparov** aufgegeben. In der Spielstärke vergleichbare Programme (z.B.»Hydra«) wurden in nachfolgenden Tournieren mit erlaubter Spieler-Unterstützung durch eigene Hilfsprogramme **geschlagen;** eindrückliches Beispiel für **›Intelligenzverstärkung‹** (**IA**) (G. **Kasparov** hat das in der New York Book Review 2010 ausführlich reflektiert).
- Nach dem spektakulären Sieg in der Quiz Show »**Jeopardy**!« hat **IBM** zur wirtschaftlichen Verwertung seines aufwendig für Hochleistungs-Hardware entwickelten Programmsystems **Watson** für Sprach- und Wissensverarbeitung 2015 einen **eigenen Geschäftsbereich** gegründet. Nach kläglich gescheiterten Einsätzen u.a. in Krebsforschungszentren und Rückversicherungen hat IBM die Gesundheits-Anwendungen **verkauft**, noch brauchbar erscheinende Reste in seine wachsenden Clouddienste integriert (FAZ 03.06.2018, NYT 21.01.2022).
- Im März 2016 gelang **Google Alphabet** mit seinem ›KI‹-Programm **AlphaGo** im Go-Spiel der Sieg gegen den koreanischen Spitzenspieler Lee Sedol. Der *homo sapiens* schien einmal mehr durch aufwendig & scharfsinnig konstruierte Algorithmen entthront. Sechs Jahre danach gelang es dem US-Amerikaner Kellin Pelrine, zwei vergleichbar leistungsfähige ›KI‹-Systeme ohne Hilfe eines Computers zu **besiegen**, obwohl er lediglich auf dem Niveau eines gehobenen **Amateurs** spielt (heise online 20.02.2023).

Konsequenz: Statt sich in **infantil naiver Faszination** durch propagandistisch gehypte ›KI‹-Systeme immer neu **verblüffen** zu lassen, kommt es darauf an, die riskante **Entwicklung vorantreibende**, durch die Propaganda aber **verborgene Machtverhältnisse** & **Interessen** zu analysieren und zu **regulieren**.

Prof. Dr.-Ing. Peter **Brödner**

Auf die **Entwicklungsperspektive** kommt es an

Die **Gestaltung** soziotechnischer Systeme muss sich an **Eigenheiten** menschlichen Handelns, Lernens und der **Entfaltung** von **Arbeitsvermögen** orientieren, um gute Arbeit, Produktivität & Innovation zu ermöglichen.

Technikzentrierte Perspektive:
AI (**A**rtificial **I**ntelligence)
»Smart machines«, »autonome Agenten«, »Intentional stance« (Dennett 1987; vgl. Minsky 1988, Shoham 1993, Wooldridge 2002, Russell & Norvig 2009)

MAS sollen lebendige Arbeit **ersetzen**; mit ›ML‹ wird Flexibilität angestrebt, aber statt ›Lernfähigkeit‹ wird nur Adaptivität erreicht. **Arbeitsvermögen** wird improvisierend **nachgeahmt** und partiell ersetzt.

Für verbleibende Restarbeit gelten »Ironies of automation« (Bainbridge 1983): Verlust praktischer Handlungskompetenz, hilflose, weil entwöhnte ›Bediener‹.

Praxistheoretische Perspektive:
IA (**I**ntelligence **A**ugmentation)
»Things [machines] that make us smart« (Norman 1993; vgl. Licklider 1960, Ehn 1988, Winograd 1996, Mumford 2006, Ulich 2011)

CS **ertüchtigen** lebendige Arbeit, dienen methodisch der **Entfaltung** von **Arbeitsvermögen** für gesteigerte Produktivkraft und Innovationsfähigkeit **durch**:

Menschengerechte, reflexive Gestaltung von Arbeitsaufgaben, technischen Arbeitsmitteln und Interaktionsformen; Fokus auf Entfaltung praktischer Handlungskompetenz.

Trotz wiederholter AI-Versuche: **Erfolgreiche Nutzung** von **CS** folgt ganz überwiegend der praxistheoretischen **IA-Perspektive** (PC als Werkzeug: Tab'kalk., Dok'edit., CAD, ERP, EDM etc.; Internet als instrumentelles Medium der Kooperation: WWW, CSCW etc.), d.h. ›**Intelligenzverstärkung**‹ in Form von »**Computer-Aided** [human activity]«.

Prof. Dr.-Ing. Peter **Brödner**

Beispiel für IA: **Computer-assisted Chess**

Intelligence Augmentation (IA) heißt, die je besonderen Stärken von menschlicher Intelligenz und maschineller Rechenleistung zu lombinieren. **Beispiel Schach**:

»Moravec's Paradox, in chess, as in so many things, what computers are good at is where humans are weak, and vice versa. This gave me an idea for an experiment. What if instead of human versus machine we played as partners? My brainchild saw the light of day in a match in 1998 in León, Spain, and we called it ›Advanced Chess‹. Each player had a PC at hand running the chess software of his choice during the game. The idea was to create the ***highest level*** *of chess ever played,* ***a synthesis of the best of man and machine****.«*

»The computer could project the consequences of each move we considered, pointing out possible outcomes and countermoves we might otherwise have missed. With that taken care of for us, we could concentrate on strategic planning instead of spending so much time on calculations. Human creativity was even more paramount under these conditions. […] My advantage in calculating tactics had been nullified by the machine. […] The teams of ***human plus machine dominated even the strongest computers.*** *The chess machine Hydra, which is a chess-specific supercomputer like Deep Blue, was no match for a strong human player using a relatively weak laptop.* ***Human strategic guidance combined with the tactical acuity of a computer was overwhelming****.«*

More precise analysis revealed: *»Weak human + machine + better process was superior to a strong computer alone and, more remarkably, superior to a strong human + machine + inferior process. […] Where so many of these investigations fail … is by* ***not recognizing the importance of the process of learning and playing chess****.«*

(Source: Kasparov, G.: The Chess Master and the Computer, The New York Book Review, Feb. 11, 2010)

Prof. Dr.-Ing. Peter **Brödner**

Fazit: Der ›**KI**‹-**Wahn** und sein ›**Glaubensbekenntnis**‹

Gemäß ›**Glaubensbekenntnis**‹ der ›**KI**‹-**Gemeinde** (Entwickler, Auftraggeber und Anwender) wird Computern als **programmgesteuerten Rechenmaschinen** in üblichen **Definitionen grundlos** eigene ›**Handlungsträgerschaft**‹ (»agency«) zuerkannt:

- *»AI is that activity devoted to making machines intelligent, and intelligence is that quality that enables an entity to function appropriately and with foresight in its environment«* (Nilsson 2010);
- AI research investigates *»intelligent agents«*, i.e. devices *»that perceive their environment and take actions maximizing the chance of successfully achieving their goals«* (Russell & Norvig 2009: 2);
- *»Rational Agent«*: *»AI researchers use mostly the notion of rationality, which refers to the ability to choose the best action to take in order to achieve a certain goal, given certain criteria to be optimized and the available resources.«* (EU High-Level Expert Group on AI 2018: 1f).

Anthropomorphisch werden Rechensystemen **Intentionalität** & **rationales Handeln**, d.h. die Wahl geeigneter **Mittel** für das **Erreichen** von **Zielen**, einfach **zugeschrieben**. Computer führen aber nur **programmierte Berechnungsverfahren** aus, denen bereits die **Methoden** & denkbaren **Bedingungen** eingeschrieben sind, mittels derer zuvor festgelegte Ziele **optimal** erreicht werden – eine nur **verliehene** ›**Intentionalität**‹.
»Agency without Intelligence« (Floridi).

Mithin **verwechseln** diese **irreführenden** ›Definitionen‹ **Herstellen** mit **Hergetelltem**, die **kreativen Fähigkeiten** intelligenten Entwerfens und Programmierens mit den operativen **Programmfunktionen** als deren **Ergebnis** – ein krasser **Kategorienfehler**.

Entgegen diesem Etikettenschwindel **verkörpern** ›KI‹-Systeme nur die **Funktionen**, die ihre **Entwickler** aufgrund eigener **Einsichten absichtlich** programmiert haben. **Intelligent** sind allenfalls die **Entwickler**, **nicht** die **Systeme**.

Prof. Dr.-Ing. Peter **Brödner**

Rechenaufwand für **Training** großer Sprachmodelle

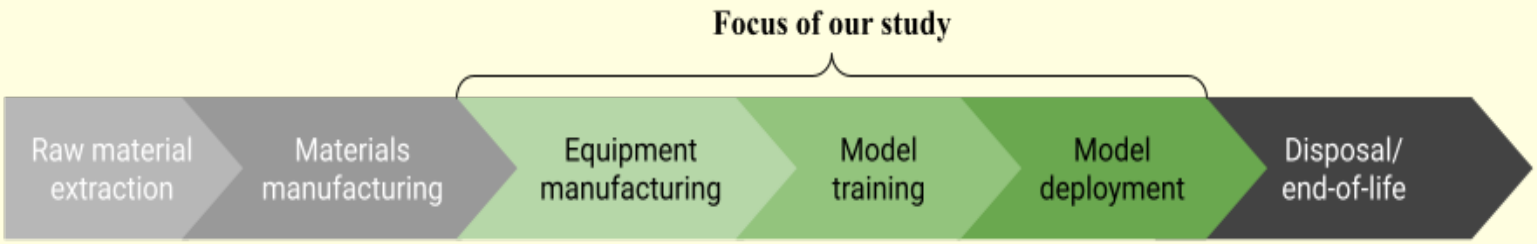


Figure 1: While the LCA approach encompasses all the stages of the product life cycle (from raw material extraction to disposal), we focus on those in green, which range from equipment manufacturing to model deployment.

Model name	Number of parameters	Datacenter PUE	Carbon intensity of grid used	Power consumption	CO_2eq emissions	CO_2eq emissions × PUE
GPT-3	175B	1.1	429 gCO_2eq/kWh	1,287 MWh	*502 tonnes*	552 tonnes *
Gopher	280B	1.08	330 gCO_2eq/kWh	*1,066 MWh*	*352 tonnes*	380 tonnes
OPT	175B	*1.09* [2]	*231gCO_2eq/kWh*	*324 MWh*	70 tonnes	*76.3 tonnes* [3]
BLOOM	176B	1.2	57 gCO_2eq/kWh	433 MWh	25 tonnes	30 tonnes

Table 4: Comparison of carbon emissions between BLOOM and similar LLMs. Numbers in *italics* have been inferred based on data provided in the papers describing the models. [B = 10^9]

GPT-3 by OpenAI (Microsoft); Gopher by DeepMind (Google); OPT by a research team (MetaAI); BLOOM by a research team.

(Source: A.S. Luccioni et al.: Estimating the Carbon Footprint of BLOOM, a 176 B Parameter Language Model, arXiv 2011.02001v1 [cs.LG] 3 Nov. 2022)

* equals appr. a fully bordered Airbus A 330 roundtrip flight Frankfurt - San Francisco

Prof. Dr.-Ing. Peter **Brödner**

Verzeichnis verwendeter **Quellen**

Babbage, C. (1832): On the Economy of Machinery and Manufactures, London: Knight, Re- print New York: Kelley 1971 (deutsch: Die Ökonomie der Maschine, Nachdruck der Originalübersetzung von 1833, mit einem ausführlichen Vorwort neu hg. von P. Brödner, Berlin: Kulturverlag Kadmos 1999)

Bender et al.(2021): On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? ACM Conference FAccT '21, https://dl.acm.org/doi/pdf/10.1145/3442188.3445922

Brödner, P. (2023): »Künstliche Intelligenz«: Dichtung und Wahrheit. Einblicke in die Technik des Berechnens und in Mythen um Intelligenz, International Reports on Socio-Informatics 20 (1), Bonn: IISI https://www.iisi.de/en/international-reports-on-socio-informatics-irsi/

Eco, U., 1976: A Theory of Semiotics, Bloomington (IN): Indiana University Press (deutsch: Semiotik. Entwurf einer Theorie der Zeichen, München: Wilhelm Fink Verlag 1991)

Falkenberg, E.D. et al., 1998: A Framework of Information Systems Concepts. The FRISCO Report, IFIP, www.mathematik.uni-marburg.de/~hesse/papers/fri-full.pdf

Kasparov, G.(2010): The Chess Master and the Computer, The New York Book Review, Feb. 11, 2010, https://marom.net.technion.ac.il/files/2016/07/Kasparov-2010.pdf

Marcus, G. (2020): GPT-3, Bloviator: OpenAI's language generator has no idea what it's talking about, MIT Technology Review 22.08.2020

Nilsson, N.J. (2010): The Quest for Artificial Intelligence: A History of Ideas and Achievements, Cambridge (UK): Cambridge University Press

Peirce, C.S. (1983): Phänomen und Logik der Zeichen, Frankfurt/M: Suhrkamp

Rechenberg, P. (1991): Was ist Informatik? Eine allgemeinverständliche Einführung, München: Hanser

Ropohl, G. (1991): Technologische Aufklärung. Beiträge zur Technikphilosophie, Frankfurt/ M: Suhrkamp

Russell, S. & Norvig, P. (2009): Artificial Intelligence: A Modern Approach, 3rd. ed., Essex (GB): Pearson

VDI (Hg.) (1991): Technikbewertung – Begriffe und Grundlagen. Erläuterungen und Hinweise zur VDI-Richtlinie 3780, VDI Report 15, Düsseldorf: VDI

Kontaktdaten

Honorarprofessor

Wirtschaftsinformatik und Neue Medien

Universität Siegen

https://www.wineme.uni-siegen.de/team/broedner/

Jüngste **Publikation** zum Thema:

»Künstliche Intelligenz«: Dichtung und Wahrheit. Einblicke in die Technik des Berechnens und in Mythen um Intelligenz, International Reports on Socio-Informatics 20 (1), Bonn: IISI 2023

https://www.wineme.uni-siegen.de/wp-content/uploads/2023/12/Broedner-KIDichtungWahrheitIRSI.pdf