

DEversAI: Training und Visualisierung deutsch lokalizierter direktionalkomplementärer LLMs

Leo Blume (16 J.)

Projektbetreuer: Prof. Dr. André Grüning

Erarbeitungsort: Hochschule Stralsund (HOST)

Fachgebiet: Mathematik/Informatik

Bundesland: Mecklenburg-Vorpommern

Wettbewerbsjahr: 2025

Inhaltsverzeichnis

1.	Projektüberblick	0
2.	Einleitung	1
2.1.	Einordnung in existierende Literatur	1
3.	Mathematische Grundlagen großer Sprachmodelle	2
3.1.	Die Transformer-Architektur	2
3.2.	GPT-2 small	2
3.3.	Training großer Sprachmodelle	3
3.4.	Kausale und antikausale Inferenz	4
4.	Infrastruktur und Methodik	5
4.1.	Datengrundlage	5
4.2.	Tokenisierung	5
4.3.	Trainingsdurchführung	5
4.4.	Infrastruktur der HOST	6
5.	Vorgehen und Resultate	6
5.1.	Anfängliche Misserfolge <code>cf1</code> und <code>cf2</code>	6
5.2.	Training von <code>anticausal1</code> und <code>causal1</code>	7
5.3.	Training von <code>anticausal-fw2</code> und <code>causal-fw2</code>	7
5.4.	Statistische Signifikanzprüfung durch Zweistichproben- <i>t</i> -Test	7
5.5.	Qualitative Analyse durch Inferenzbeispiele	8
5.6.	Finetuning	9
5.6.1.	Deutsche Gesetzestexte	9
5.6.2.	Plenarprotokolle des Deutschen Bundestags	9
5.7.	Vergleich	10
6.	Visualisierung	10
6.1.	Token-Embedding	10
6.1.1.	Dimensionalitätsreduktion und Clustering	10
6.1.2.	Scree-Plots und konkatenierte Räume	11
6.1.3.	Kanonische Korrelationsanalyse	13
6.2.	Positions-Embedding	13
7.	Webanwendung	14
7.1.	Architektur	14
7.2.	Funktionen von Unterseiten (Auswahl)	14
7.2.1.	Unterseite <code>/token/[id]</code>	14
7.2.2.	Unterseite <code>/token/embedding-space</code>	14
7.2.3.	Unterseite <code>/chat</code>	14
8.	Fazit und Ausblick	15
9.	Danksagung	A
10.	Quellen- und Literaturverzeichnis	B

1. Projektüberblick

Im Projekt DEversAI untersuche ich, ob KI-Sprachmodelle besser funktionieren, wenn sie Texte vorwärts oder rückwärts verarbeiten. Dazu habe ich zwei KI-Modelle auf Deutsch trainiert: eines erzeugt Text vorwärts, das andere rückwärts. Ziel ist es, herauszufinden, ob Rückwärts-Modelle neue Möglichkeiten eröffnen und ob Erkenntnisse aus englischer Forschung im Deutschen gelten.

Die Ergebnisse sind vielversprechend. Das Vorwärts-Modell liefert präzisere Vorhersagen, aber das neue Rückwärts-Modell kann auch gute Texte vom Ende aus verfassen - so bei Kochrezepten, Gesetzen und Bundestagsreden. Eigene komplexe Visualisierungen der Modellstrukturen zeigen, dass beide sprachliche Muster lernen, aber sich in Aufbau und Ausgabe unterscheiden.

In der entwickelten interaktiven Webanwendung kann die KI ausprobiert und getestet werden. Die Resultate belegen, dass die Textverarbeitungsrichtung einen wesentlichen Einfluss auf die Leistungsfähigkeit von KI in der Sprachverarbeitung hat.

„What I cannot create, I do not understand.“

— Richard Feynman

2. Einleitung

Seit der Einführung des Transformer-Modells[1] sind große Sprachmodelle (*large language models*, LLMs) Vorzeige- und zugleich wichtigster Untersuchungsgegenstand der angewandten KI-Forschung[2]. Dabei haben sich autoregressive kausale Modelle auf Basis von Decoder-Only-Transformern wie GPT in der Praxis trotz theoretischer Schwachstellen[3] gegenüber nicht-kausalen Modellen wie BERT durchgesetzt[4–6]. Eher geringe Betrachtung, darunter in [7], fanden dagegen *antikausale* Modelle, welche den Zeitschritt und damit die Direktionalität des Textes umkehren. Daher sollen in diesem Projekt kausale mit parametergleichen antikausalen Modellen verglichen und anhand verschiedener Metriken evaluiert werden.

Dabei wird die Projekt- und damit auch Modellsprache auf die deutsche Sprache beschränkt. So kann überprüft werden, ob Erkenntnisse aus der angloamerikanisch dominierten Forschung sprachübergreifend gelten. Zudem ist eine weiterführende Abgrenzung zu existierenden Modellansätzen und -untersuchungen möglich.

Der Hauptteil der schriftlichen Arbeit beginnt mit einer illustrierten Erklärung großer Sprachmodelle, einer Herleitung ihrer Parameteranzahl und einer groben Erklärung ihres Trainings (3.). Die die Methodik und Vorgehensweise der Projektausarbeitung wird beschrieben (4.), ehe die konkreten trainierten Modelle vorgestellt und quantitativ sowie qualitativ und exemplarisch, einhergehend mit einer Untersuchung sprachlicher Eigenschaften der Korpora selbst evaluiert werden (5.). Nach ausführlicher Visualisierung latenter Räume der beiden Hauptmodelle (6.) wird die die Webanwendung kurz vorgestellt (7.). Abschließend werden die Ergebnisse, insbesondere die wesentlichen korpuspezifischen Unterschiede der direktionalkomplementären Modelle, zusammengefasst und ein Ausblick geplanter Weiterentwicklungen präsentiert (8.).

2.1. Einordnung in existierende Literatur

Die vorliegende Arbeit ist von ähnlichen und bereits extensiv thematisierten Ideen der Forschung an Sprachmodellen abzugrenzen. Nicht-autoregressive bidirektionale Modelle wie BERT[8] sollen daher nicht thematisiert werden, ebenfalls wird FIM[9], ein effektiver Ansatz, um Sprachmodelle zu trainieren, die beidseitige Kontextfenster einbeziehen können, nicht verglichen, da die durchgeführte Inferenz stets kausal bleibt. Im Gegensatz zu Reversal-Curse[10] wird auf LLM-Faktenzugriff im Projekt aufgrund kleinerer verwendeter Modellarchitekturen nicht näher eingegangen; auch inkrementelle Verbesserungen großer LLMs durch Training auf Rückwärts-Daten, wie [11] oder [12], stehen nicht im Fokus. Bezug genommen auf und komplementiert wird das unveröffentlichte Paper [7], welches primär große LLMs auch antikausal finetuned und trainiert.

Laut arXiv wurde [7] noch nie referenziert, sodass es sich bei den präsentierten Ergebnissen im Bezug auf autoregressive antikausale und direktonalitätskomplementäre Sprachmodelle nach bester Kenntnis der Autorin um neue handelt. Das Projekt hebt sich auch durch die Beschränkung auf die deutsche Sprache zur sprachspezifischen linguistischen Untersuchung hervor.

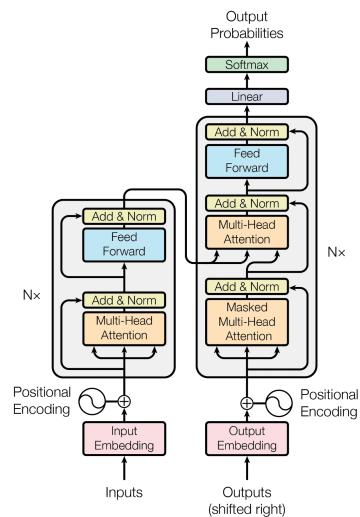


Abb. 1: Das in [1] gezeigte architekturelle Modell, Abb. übernommen.

3. Mathematische Grundlagen großer Sprachmodelle

3.1. Die Transformer-Architektur

Die heute vorherrschende Architektur zur Definition großer Sprachmodelle ist die 2017 in [1] eingeführte Transformer-Architektur, welche rekurrente und konvolutionale Netzwerke für die Texterzeugung ablöste, indem nur noch auf den Attention-Mechanismus gesetzt wird. Abb. 1 zeigt die Originaldarstellung des Modells, welche bereits die duale Encoder-Decoder-Struktur aufzeigt.

Aufgrund der Fülle an öffentlich verfügbaren Informationen und Erklärungen dieser Modelle^[1] und der Seitenbegrenzung wird auf eine ausführliche Erklärung des allgemeinen Modells verzichtet und stattdessen das konkrete GPT-2 small-Modell, welches als handhabbares und dennoch leistungsfähiges Modell gewählt wurde, im Folgenden präsentiert.

3.2. GPT-2 small

Die von OpenAI 2019 vorgestellte GPT-2-Modellfamilie[16] beruht ebenso wie das Vorgängermodell GPT auf einer Decoder-Only-Architektur, sodass der Enkodierungsteil des Netzwerks nicht implementiert wird. Die Verarbeitung erfolgt in drei Schritten:

1. Einbettung der Tokens in den Embedding-Raum,
2. Mehrfache Verarbeitung der Embedding-Vektoren in Transformer-Blöcken, sodass die Vektoren über Self-Attention kontextualisiert werden sowie
3. Abbildung auf Wahrscheinlichkeitsdistribution über Tokens.

Hyperparameter des Modells sind dabei die Embeddingdimensionalität E , Kontextgröße P , Vokabulargröße V , Anzahl von Transformerblöcken n_{layer} und Anzahl von Attention-Heads pro Block n_{head} . Abweichend von der ursprünglichen GPT-2-Architektur wird in hier trainierten Modellen in allen `LayerNorm`- und `Linear`-Schichten kein Bias verwendet, da dieser sich leicht negativ auf das Training auswirkt[17]. In Abb. 2 wird ein grober Überblick des Datenflusses zur Inferenzzeit dargestellt.

Am Anfang der Datenverarbeitung steht eine Einbettung der Tokens in den latenten E -dimensionalen Embedding-Raum des

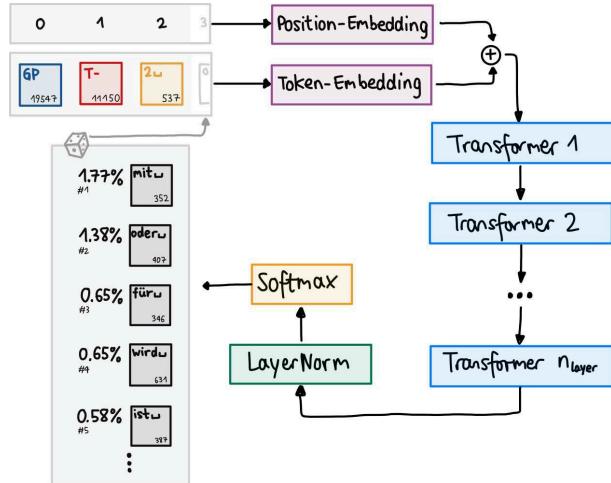


Abb. 2: Aufbau des GPT-2-Modells am Beispiel der Kausalinferenz mit Eingabe `GPT-2`.

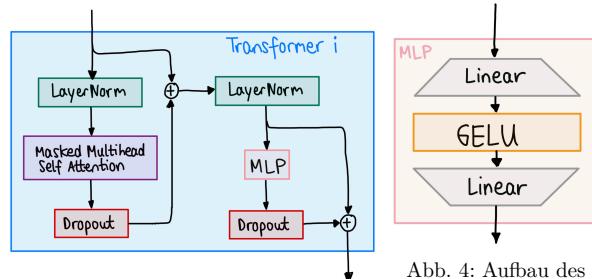


Abb. 3: Die Bestandteile eines Transformer-Blocks des GPT-2-Modells.

Abb. 4: Aufbau des Multi-Layer Perzepton.

^[1]Zu erwähnen seien neben den einleitenden Fachartikeln insbesondere [13], [14] und [15].

Modells. Dieser erfolgt über eine Parametermatrix der Größe $V \times E$. Durch das positionale Embedding, einer lernbaren $P \times E$ -Lookup-Tabelle, deren Ergebnisse im Anschluss mit den Vektoren aus E aufaddiert werden, kann der Embedding-Vektor mit der encodierten Positionsinformation augmentiert werden^[2].

Jeder der folgenden n_{layer} Decoder-Transformer-Blöcke ist nun identisch aufgebaut (s. Abb. 3). Zu Beginn steht eine `LayerNorm`-Schicht mit E Parametern, welche entlang jeder Embedding-Dimension die Distribution der Elemente des Eingangstensors auf $\mu = 0, \sigma = 1$ normalisiert und eine lineare Transformation ausführt. Sie stabilisiert den Trainingsprozess und beschleunigt die Generalisierung[19,20].

Als definierendes Element folgt darauf eine Masked Multihead Self Attention-Schicht (s. Abb. 5). Sie besteht aus zwei Unterschichten mit lernbaren Parametern, die beide lineare Transformationen beschreiben. Die erste erzeugt aus den Eingaben die drei für Attention benötigten *Query*, *Key*- und *Value*-Matrizen^[3]. Sie wird als zu multiplizierende $E \times 3E$ -Matrix gespeichert.

Die zweite parametrisierte Unterschicht bildet die durch Matrixmultiplikation berechneten Attention-Werte zurück auf Embedding-Vektoren ab und wird als $E \times E$ -Matrix gespeichert. Zudem liegt eine Dropout-Schicht vor, welche mit einer zuvor festgelegten Wahrscheinlichkeit p_{drop} einzelne Neuronen deaktiviert, um Überanpassung an Trainingsdaten zu vermeiden[21,22].

Die Ausgaben des Attention-Blocks werden nach erneutem Dropout zu den ursprünglichen Werten addiert – dieser Schritt verhindert katastrophales Vergessen der Eingabewerte – und nach weiterer `LayerNorm` mit E Parametern dienen die Resultate als Eingabe für eine MLP-Schicht, welches eine versteckte Schicht mit $4E$ Neuronen enthält und dann zurück auf E abbildet. Dieses Feedforward-Netzwerk hat insgesamt $8E^2$ Parameter: je $4E^2$ Gewichte zwischen zwei Schichten.

Insgesamt hat jeder der n_{layer} Transformerblöcke $E + (3E^2 + E^2) + E + (8E^2) = 12E^2 + 2E$ Parameter. Zusammen mit den anfänglichen Embeddings und einer finalen bias-freien `LayerNorm` ergibt sich $(V + P)E + n_{\text{layer}}(12E^2 + 2E) + E$ als Gesamtparameterzahl. Für die verwendeten Hyperparameter, die bis auf eine in Abschnitt 4.2 erwähnte Ausnahme denen von GPT-2 small entsprechen (s. Abb. 6), ergibt sich folglich[23]:

$$\begin{aligned} & (50'304 + 1024) \cdot 768 + 12 \cdot (12 \cdot 768^2 + 2 \cdot 768) + 768 \\ &= 50'304 \cdot 768 + 1024 \cdot 768 + 12 \cdot 7'079'424 + 768 \\ &= 38'633'472 + 786'432 + 84'953'088 + 768 \\ &= 124'373'760 \end{aligned}$$

V	50304
E	768
P	1024
n_{layer}	12
n_{head}	12

Abb. 6: Hyperparameter des Modells.

Somit hat das Modell etwa 124.37M zu trainierende Parameter.

3.3. Training großer Sprachmodelle

GPT-2 ist ein autoregressives LLM, was bedeutet, dass ein Inferenzschritt im kausalen Fall dem Vorhersagen des nächsten Tokens nach einer Liste gegebener Tokens, dem Kontextfenster,

^[2]Im ursprünglichen Transformer[1] wurde diese Enkodierung statt durch lernbare Parameter als unterschiedliche sinusoidale Funktionen fixiert; seit [18] gilt dies jedoch als überholt.

^[3]Auf eine genaue Beschreibung des Attention-Mechanismus wird hier ebenfalls der Kürze halber verzichtet, es sei auf die zitierte Literatur verwiesen.

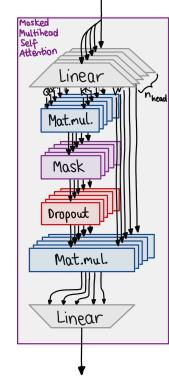


Abb. 5:
Schematischer Aufbau der Masked Multihead Self Attention.

entspricht. Um längeren Text erzeugen zu können, wird dementsprechend das Kontextfenster mit dem neuen Token konkateniert und die Inferenz erneut ausgeführt. Das Training erfolgt daher über überwachtes Lernen, indem eine große Anzahl an Beispielen von Kontextfenstern und folgenden Tokens bereitgestellt wird, anhand derer das Modell Syntax, Grammatik, Struktur und Bedeutung der bereitgestellten Sprache erlernen soll.

Im konkreten Prozess werden die Parameter des Modells dabei zu Beginn zufällig initialisiert. Es beginnt die Trainingsschleife, in welcher für eine festgelegte Anzahl an Iterationen jeweils ein Trainingsschritt ausgeführt wird. Dieser besteht darin, dass das Modell für eine Anzahl an Beispielen (der Batchgröße) das nächste Token vorhersagt. Da das vorliegende Token bekannt ist, kann dessen Wahrscheinlichkeit der finalen Schicht entnommen werden. Häufig wird der negative natürliche Logarithmus, (*negative log likelihood*, NLL) dieses Werts als Verlustfunktion gewählt[24] und ist im Laufe des Trainings über alle Trainingsbeispiele zu minimieren. So wird die Wahrscheinlichkeit, das vorliegende Token auszugeben, im Trainingsverlauf maximiert.

Zu diesem Zweck wird über den Backpropagation-Algorithmus der Gradient aller Parameter bezüglich des Verlusts berechnet, damit diese über einen geeigneten Optimizer im Anschluss angepasst werden können. In der Praxis bewährt hat sich AdamW[25,26], eine Verbesserung von Adam[27], einem adaptiven Optimizer, welcher neben dem Hyperparameter der Lernrate η jedem Parameter basierend auf dessen Gradientenverlauf eine individuelle Lernrate zuweist.

Die Lernrate selbst bleibt dabei nicht konstant, sondern wird am Anfang linear erhöht (Warmup), um dann über Kosinus-Annealing[28] im Trainingsverlauf reduziert zu werden. So wird eine Überanpassung in den ersten Schritten verhindert, im Anschluss ein schnelles Training gestattet und zum Schluss Fluktuation vermieden. Um Anpassungen auf Basis zufälligerweise nicht repräsentativer Trainingsbeispiele sowie die klassischen Probleme der explodierenden Gradienten[29,30] zu verhindern, wird zudem Gradient Clipping[31] genutzt, sodass die Anpassung der Parameter pro Schritt zusätzlich limitiert wird. Weight Decay[32,33] verbessert ebenfalls Generalisierung durch Bestrafung von Parametern zu hohen Beträgen.

3.4. Kausale und antikausale Inferenz

Der Einsatz eines großen Sprachmodells nach dem beschriebenen Training zur Textgeneration wird als Inferenz bezeichnet. Die Inferenz eines autoregressiven Modells auf Basis einer Eingabe läuft wie folgt ab: die Eingabe wird (als Liste von Tokens) ins Kontextfenster des Modells gelegt, um die Wahrscheinlichkeitsdistribution eines weiteren Tokens mittels der Modellschichten zu berechnen. Aus dieser wird ein Token stochastisch gewählt, ins Kontextfenster hinzugefügt und der Prozess wiederholt, bis eine genügende Anzahl an Tokens generiert wurde.

Alle laut [34] relevanten LLMs basieren dabei auf der natürlich erscheinenden[7] kausalen Inferenz. Dabei wird basierend auf dem Kontextfenster stets das folgende Token vorhergesagt und so der Text in Leserichtung vervollständigt. Im Gegensatz dazu steht die antikausale Inferenz, welche entgegen der Leserichtung arbeitet. Beispielhaft illustriert werden beide in Abb. 7.

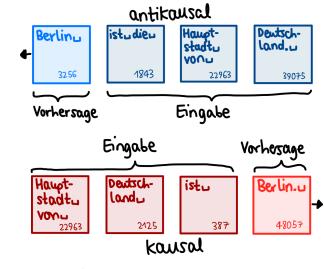


Abb. 7: Beispiele der Direktionalität.

4. Infrastruktur und Methodik

4.1. Datengrundlage

Noch vor dem Training steht die Wahl einer unter den Gesichtspunkten der Verfügbarkeit und Qualität geeigneten Datengrundlage. Hauptdatensatz ist das deutsche Fragment des FineWeb2-Datensatzes[35], eines auf Common Crawl beruhenden sprachlich aufgeteilten Korpus.^[4]

4.2. Tokenisierung

Dieser Datensatz wurde als Grundlage für einen in Rust programmierten nebenläufigen Tokenizer auf der Grundlage von Byte Pair Encoding (BPE)[38] verwendet, um insgesamt 50000 zusammengesetzte Tokens zu generieren. Dabei wurden zwei korpusentsprechende Anpassungen vorgenommen:

1. Es findet keine Pre-Tokenization statt. Dies gewährleistet, dass auch mehrwortige Tokens (wie `in_der_`) gebildet werden können, und stellt sicher, dass kein Sonderzeichen eingesetzt werden muss, um Tokens mit Leerzeichen repräsentieren zu können.^[5]
2. Statt in jedem Iterationsschritt nur das häufigste Tokendigramm (l_1, r_1) zu einem neuen Token $l_1 \circ r_1$ zusammenzufügen, wird ein Parameter $\eta \in [0, 1]$ eingeführt. Hat (l_1, r_1) eine bestimmte Häufigkeit f , so werden auch alle weiteren Tokens (l_n, r_n) hinzugefügt, für deren Häufigkeit $f_n \geq \eta \cdot f_1$ gilt und die nicht durch ein bisheriges Token bereits aufgelöst werden können, also $\forall i \in [1, n - 1] : l_n \neq r_i \wedge r_n \neq l_i$.

Im Randfall $\eta = 1$ ergibt sich das typische BPE-Verfahren, während für sinkende Werte von η mehr Tokenpaare pro Schritt zusammengefügt werden. Heuristisch wurde für das Training der stückweise lineare η -Scheduler $\eta : [0, 1] \rightarrow [0, 1], \eta(t) \mapsto \begin{cases} 0.8 & \text{falls } t \leq 0.3 \\ 0.8 + \frac{t}{7} & \text{falls } t > 0.3 \end{cases}$ genutzt, wobei $t \in [0, 1]$ den Anteil bereits ersteller Tokens beschreibt.

Zu den 50000 zusammengesetzten hinzu kommen die $2^8 = 256$ anfänglichen Tokens jedes möglichen ursprünglichen Bytewertes, die als Basistokens bezeichnet werden. Die Gesamtvokabulargröße ist somit $50000 + 256 = 50256$ ^[6]. Abb. 8 zeigt einen Auszug aus diesem Vokabular, wobei `_` ein enthaltenes Leerzeichen repräsentiert.

ID	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275
Token	<code>en</code>	<code>er</code>	<code>ch</code>	<code>e</code>	<code>en</code>	<code>ei</code>	<code>t</code>	<code>er</code>	<code>un</code>	<code>s</code>	<code>n</code>	<code>,</code>	<code>d</code>	<code>st</code>	<code>an</code>	<code>ie</code>	<code>.</code>	<code>ch</code>	<code>el</code>	<code>au</code>
ID	2500		5000		7500		10000		12500		15000		17500		20000		22500		25000	
Token	<code>son</code>		<code>Samm</code>		<code>zeichni</code>		<code>wissenschaftlichen</code>		<code>Gesam</code>		<code>:<</code>		<code>dlicher</code>		<code>das Haus</code>		<code>schenkt</code>		<code>saubere</code>	
ID	27500		30000		32500		35000		37500		40000		42500		45000		47500		50000	
Token	<code>interessanter</code>		<code>space</code>		<code>Morgens</code>		<code>und werde</code>		<code>zo</code>		<code>Grundgesetz</code>		<code>Sept</code>		<code>Neben einem</code>		<code>gekocht</code>		<code>akte</code>	

Abb. 8: Eine Auswahl zusammengefügter Tokens aus dem deutschen `fineweb2`-Vokabular.

Der gesamte Korpus wurde über einen Zeitraum von zirka drei CPU-Monaten in die Tokenrepräsentation encodiert und als Binärdateien, jeweils von Bytepaaren (Big Endian), gespeichert.

4.3. Trainingsdurchführung

Der Trainingscode beruht auf Karpathys nanogpt[17], welcher dem bereits erwähnten Aufbau des GPT-2-Modells von OpenAI vollständig folgt und das Modell in PyTorch[40] implementiert.

^[4]Vorherige Modelltrainings nutzten den OSCAR-Datensatz[36], welcher sich jedoch aufgrund von Qualitätsdefiziten trotz Augmentation mit der deutschen Wikipedia[37] für längerfristig ungeeignet erwies, s. etwa Abb. 12.

^[5]Im ursprünglichen GPT-2-Vokabular wird hier `G`, `LATIN CAPITAL LETTER G WITH DOT ABOVE`, genutzt[39].

^[6]Typischerweise wird noch ein End-Of-Text (EOT)-Token ergänzt, sodass die Vokabulargröße 50257 beträgt; da der zu encodierende Text in diesem Fall jedoch auf UTF-8 beschränkt ist, kann das ansonsten nicht auftauchende Basistoken `0xff` = 255 diesen Zweck erfüllen.

Vollständig neu wurde nur die Logik zum Laden der Datenfragmente sowie die zur CLI-interaktiven Evaluation geschrieben; die Modelldefinition wurde zielgemäß beibehalten.

Anzumerken ist, dass es zum Training des antikausalen Modells keiner Anpassung der Modellstruktur bedarf, da es genügt (s. Abb. 9), die Tokens der Eingabedatei umzukehren und Modelleingabe X mit erwünschter Vorhersage Y zu tauschen. Durch diese Umkehr der Tokenzeit agiert die kausale Attention-Maske antikausal.

	Eingabe	Ausgabe
kausal	i	$i+1$
	$i+1$	$i+2$
	\dots	\dots
	$i+P-1$	$i+P$
antikausal	$i+P$	$i+P-1$
	$i+P-1$	$i+P$
	\dots	\dots
	$i+1$	i

Abb. 9: Indexintervalle der Trainingsbeispiele ab i .

Zum Zwecke von Training und Inferenz wurde durch die Hochschule Stralsund (HOST) das GPU-Cluster LLM-HOSTed („Kira“) dankenswerterweise bereitgestellt. Es handelt sich bei den in diesem Projekt verwendeten um die ersten auf den GPUs dieses Clusters[41] trainierten Sprachmodelle. Zur Verfügung stehen insgesamt zehn NVIDIA A100-GPUs, von denen jedoch zwei dauerhaft und vier regelmäßig anderweitig beansprucht werden, sodass das mittelfristige Training der LLMs auf vier GPUs mittels DDP^[7] parallelisiert ausgeführt werden kann.

5. Vorgehen und Resultate

5.1. Anfängliche Misserfolge `cf1` und `cf2`

Die ersten Versuche des Trainings eines kausalen LLM begannen Anfang Dezember 2024, nachdem das ursprüngliche OSCAR-basierte Vokabular erfolgreich trainiert und der Korpus über diese Tokens enkodiert worden war. Abb. 10 zeigt das Verhalten der Verlustfunktion über den Zeitraum der Trainingsschritte für die ersten beiden Anläufe. Die x-Achse zeigt dabei die Trainingsschritte, die y-Achse den NLL-Verlust.

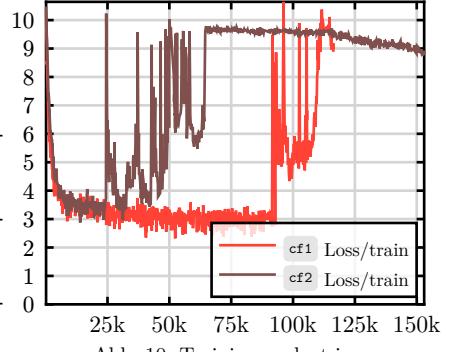


Abb. 10: Trainingsverlust im Trainingsschrittverlauf mit Lernrate $\eta := 1 \times 10^{-6}$.

Es ist zu erkennen, dass der Verlust erwartungsgemäß innerhalb der ersten 5k Schritte beständig sinkt und der Betrag der Steigung bis Schritt 25k stark sinkt.

Ab Schritt 90k für `cf1` (bzw. 25k für `cf2`) ist allerdings zu erkennen, dass der Wert der Verlustfunktion schnell ansteigt, zeitweise stark fluktuiert und dann vergleichbar mit dem anfänglichen Wert um 10 NLL stagniert, was insbesondere für `cf2` erkennbar wird. Die Modelle verlernen demzufolge die Fähigkeit der Textinferenz vollständig; die Ausgabe ist eine inkohärente Tokenkette, nicht von einem zufälligen Modell zu unterscheiden.

Der Grund für dieses Verhalten konnte bisher nicht eindeutig identifiziert werden^[8]. In folgenden Trainingsverläufen wurde der Wert des Gradienten genauer überprüft, jedoch konnte keine Anomalie festgestellt werden. Der Versuch, das Modell von einem bisherigen Checkpoint wiederherzustellen, scheiterte ebenfalls: bei weiterführendem Training von `cf1` ab Schritt 90k, 80k und 50k mit bis auf Seed gleichen Hyperparametern konnte ein vergleichbarer hoher Anstieg des Verlusts beobachtet werden.

Heinz Müller (Rechtswissenschaftler)
Heinz Müller (* 25. September 1931 in Wien; † 8. Februar 2024 in Imst, Oberösterreich) war ein österreichischer Rechtswissenschaftler.

Leben
Heinz Müller wuchs in Wien auf. Er studierte Wirtschaftsjurisprudenz und wurde 1955 in Wien zum Dr. jur. promoviert. In der Folge erhielt er die Lehrkanzel für Zivilrecht und wurde 1955 als Rechtsanwalt zugelassen. Später war er von 1965 bis 1970 als Richter am Amtsgericht in Linz tätig. Im Jänner 1972 nahm er dort als Beamter seine Tätigkeit als selbständiger Richter als Anwalt auf. Im gleichen Jahr wurde er wegen „Wirtschaftsblockaden“ wieder entlassen. Anschließend war er als Rechtsanwalt tätig, bis er im Jahr 2000 aus Ruhestand wieder entlassen wurde. Er wird bei Feiern oft als „außergewöhnlichster Mann Europas“ bezeichnet. Auch international wird Müller von vielen Sehern als Rekordmeister Österreichs angesehen. Heinz Müller starb 2024 an Krebs.

Abb. 11: `anticausal1`-generierte Biographie

^[7]Kurz für *distributed data parallel*[42], eine Methode zur Parallelisierung des Modelltrainings auf mehrere GPUs in PyTorch, mittlerweile für größere Modelle teils überholt[43].

^[8]Es scheint sich um einen Extremfall des in der Literatur als *loss spike*[44–46] bekannten Problem zu handeln.

5.2. Training von `anticausal1` und `causal1`

Durch Ablationstests der relevanten verwendeten Hyperparameter wurde die Lernrate als für den zu verhindernden zuvor festgestellten Verlustanstieg entscheidenden Faktor bestimmt. Eine sechsfache Erhöhung der Lernrate auf 6×10^{-6} (nach Anwendung des Gradient Clipping wie bisher) führte zwar zu höheren lokalen Fluktuationen, insgesamt aber zu stabilerem Training und einer schnelleren Modellkonvergenz. Das Training erfolgte hier bis Schritt 300k für das kausale und antikausale Modell. Die so trainierten Modelle, als `causal1` und `anticausal1` bezeichnet, konnten bereits erfolgreich bis zum Landeswettbewerb (s. Abb. 11) eingesetzt werden; jedoch spiegelte sich die geringe Token- und Korpusqualität in den Modellausgaben (s. Abb. 12) wieder. Alle nachfolgenden Auswertungen wurden auch mit diesen Modellen bereits zuvor reproduziert.

5.3. Training von `anticausal-fw2` und `causal-fw2`

Daher wurde ab Mitte Februar auf Grundlage des Fineweb2-Korpus und -Vokabular^[9] das Training der neuen Grundmodelle `causal-fw2` und `anticausal-fw2` durchgeführt. Die folgende Evaluation dieser Modelle erfolgte nach dem Landeswettbewerb. Abb. 13 zeigt den Wert der Verlustfunktion für Trainings- und Validierungsdaten im Trainingsverlauf^[10].

Abb. 14 zeigt den Verlust ausgewählter Checkpoints auf nicht zum Training verwendeter Validierungsdaten. Hervorzuheben sind die inkrementellen Verbesserungen im Trainingsverlauf, insbesondere jedoch, dass `causal-fw2` zu allen ausgewählten Trainingszeitpunkten und für alle Validierungsshards einen leicht geringeren Verlust im Vergleich zu `anticausal-fw2` aufweist, was eine höhere Generalisierungsfähigkeit kausaler Modelle indiziert.

5.4. Statistische Signifikanzprüfung durch Zweistichproben-*t*-Test

Für die konkreten trainierten Modelle sind diese Unterschiede statistisch signifikant, wie ein Zweistichproben-*t*-Test^[48] in SciPy^[49] belegen konnte, für den zuvor $\alpha = 0.05$ ^[50] festgelegt worden war. Für $H_0 := \text{"Verlust } \text{anticausal-fw2} = \text{Verlust } \text{causal-fw2} \text{"}$ und $n = 500k$ gemessenen Batches, OSCAR bzw. Wikipedia, liegt die Wahrscheinlichkeit der Observationen unter Annahme von H_0 (*p*-Wert^[51]) bei 1.34×10^{-153} . Daher gilt $p < \alpha$ und H_0 wird verworfen, sodass geschlussfolgert wird, dass die Modelle signifikant verschieden sind. Die Effektstärke nach

Schaden (hier klicken) Verwendest du Cookies & Datenschutzeinstellungen, damit wir Ihnen beim Besuch unserer Internetsite einen besseren Service bieten können. Datenschutzerklärung Einsehen Wie wir Cookies verwenden Wir können Cookies anfordern, die auf Ihrem Gerät eingestellt werden. Wir verwenden	Dienstleistungen gezeigte Landesverbirgsgeenschaft passen Senioren in Land der richten und benker auf einen Abweg geraten. 30. Engelmonat 1999. Gieseking, Süd: Baudenkmale, Denkmale über Kunstwerke der Galionsfahrt, 1. galvanischen Überzug. Nymphenburger Verlagshaus, München 1995. ISBN 3-485-04074-0	angeboten. Die Eingabe der E- Mail-Adresse für die Ermittlung Ihrer E- Mail-Adresse ist ausgeschlossen, wenn Sie eine gültige E-Mail- Adresse angegeben haben, jedoch keine E- Mail-Adresse geschenkt bekommen. Bitte andere Personen benötigt, um sie anzumelden. Für die E-Mail- Werbung benötigen Sie eine valide E-Mail- Adresse, damit wir
---	--	---

Abb. 12: Niedrigqualitative promptlose Ausgaben von `causal1`.

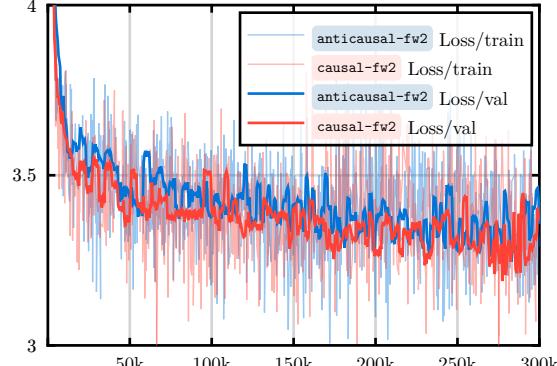


Abb. 13: Trainingsverlust der kausalen und antikausalen Modelle mit Lernrate $\eta := 6 \times 10^{-6}$.

Modell \ Shard	FineWeb2		
	1	2	3
causal-fw2	3.4248	3.3984	3.3912
	3.3788	3.3528	3.3455
	3.3435	3.3176	3.3102
anti-causal-fw2	3.4624	3.4351	3.4293
	3.4071	3.3798	3.3716
	3.3750	3.3481	3.3409

Abb. 14: Verlust auf Validierungsdaten.

^[9]Als Jaccard-Koeffizient^[47] zwischen den Mengen an Tokenwerten der beiden Vokabulare wurde $\frac{29591}{70921} \approx 0.4172$ bestimmt; diese Unterschiede folgen jedoch nicht nur aus Qualitätsunterschieden, sondern auch Änderungen der Tokenerzeugung.

^[10]Zur besseren Veranschaulichung skaliert auf $3 \leq y \leq 4$.

Cohen[52] liegt bei 0.0528.^[11]

5.5. Qualitative Analyse durch Inferenzbeispiele

Um das Grundmodell testen und mit anderen kleinen LLMs vergleichen zu können, können neben den quantitativen Analysen auch qualitative Überprüfungen durchgeführt werden. Hierfür wurden vier Beispiele für typische Textstrukturen – ein politischer Artikel, ein Kochrezept, ein wissenschaftsjournalistischer Bericht und ein Haushaltsratgeber – gewählt, Abb. 15 zeigt die Kausalinferenz.

<p>Wie Vertreter der amerikanischen Regierung am Freitag mitteilten, hätte der republikanische Senator John Boehner am Donnerstag eine Anhörung zum Thema Umweltdelikte und strafrechtliche Verfolgung wahrnehmen sollen. Bei der Anhörung hatten sich mehrere Senatoren von Demokraten und Republikanern um eine Aussage bemüht, zudem war für den Samstag eine Kundgebung mit rund 2000 Teilnehmern geplant. Wie auf einer Pressekonferenz in Oklahoma City vom Freitag weiter bekannt wurde, sollen am Donnerstag auch mehrere Republikaner und Demokratische Politiker zu einer Anhörung erscheinen. Laut dem Sprecher des republikanischen Senats, Kevin Johnson, wurde damit möglicherweise das Vorgehen der US-Regierung gegen Umweltdelikte wie Abholzung und Wasserverschmutzung infrage gestellt. Er sprach von einem „Akt der Wahrheit.“ Kritiker werfen Trump vor, mit seinen harschen Worten gegen die Umweltgesetze in den USA zu verstören. Trump habe sich unter anderem über die „Vertuschung“ des Mordes an einer Polizeigewerkschaft</p>	<p>Rezept: Eierkuchen mit Apfelmus Zutaten: - 200 g Mehl - 200 g Zucker - 75 g Butter - 2 Eigelb - Jetzt hier kaufen! - 150 g Apfelmus - Jetzt hier kaufen! Den Backofen auf 170°C Ober- und Unterhitze vorheizen. Die Butter mit dem Zucker und dem Zimt schaumig rühren, Eier einzeln und einzeln unterrühren. Die Eigelbe mit dem Mehl so gut wie möglich verquirlen, dann die Mandelblättchen in die Mischung einröhren. Die Mehlmischung gleichmäßig auf der Masse verstreichen, sodass eine homogene, fast krümelige Masse entsteht. Die Mischung auf den Teig streichen. Den Kuchen jetzt etwa 10-15 Minuten backen. Die Hälfte der Apfelstücke entfernen und vom Kuchen lösen. Dann mit Apfelmus mischen und mit dem restlichen Zucker in eine Schüssel geben. Dann etwas abkühlen lassen. Die restlichen Apfelstücke mit dem</p>	<p>Forschende der Hochschule Stralsund haben herausgefunden, dass Menschen mit Hörbehinderungen oft am häufigsten wegen ihres Körpergewichtes am Hörerleben beteiligt sind. Diese Zahlen geben nun einen Überblick über die Art und Weise der Hörsehädigung sowie ihrer Ursachen. „Hörsehädigung kann unter anderem unterschiedliche Ursachen haben, unter anderem zu wenig Sauerstoff, schlechte Nasenatmung oder zu wenig Sauerstoffaufnahme“, erklärt Prof. Dr. Christian Gillmann, Professor an der HS Stralsund, Leiter des Instituts für Hörneurologie und Neurochirurgie. Die Forschung der Hochschule kombiniert daher neurochirurgische und spezialisierte Hörsysteme mit funktionellen Hörgeräten, um die Hörsehädigungen zu beseitigen und die Lebensqualität der Hörbeeinträchtigten deutlich zu erhöhen. Die Hörgeräte der HNO verfügen über spezielle Hörsysteme mit Mikrofonen, die quasi die Stimme des Audiometers imitieren. Das heißt, die Hörverstärker strahlen dieses Konzept aus, so dass sowohl das Gehör als auch die Atmung des Hörorgans verbessert werden.</p>	<p>Die 5 wichtigsten Tipps zum Ausmisten 1. Den Schreibtisch aufräumen Zuerst sollten Sie Ihren Schreibtisch entrumpeln. Verstauben Sie alles, was Sie nicht mehr brauchen, wie z.B. alte Aktenordner, Klebezettel, Fotos etc. So haben Sie einen sauberen, geordneten Haushalt. Suchen Sie sich vorher gute und weniger gute Unterlagen aus. Und suchen Sie sich Ihre Lieblingsspirituose: Wenn Sie keine finden können, dann schauen Sie doch einmal auf Wikipedia nach. Es gibt dort viele nützliche Links für alte Bücher, DVDs und CDs. Nehmen Sie sich also Zeit. Machen Sie den Aufräum-Plan und dann lassen Sie alles stehen und liegen. 2. Rund ein oder zwei Stunden Zeit einplanen Wenn Sie ein Haus besitzen kann es durchaus sinnvoll sein, Ihr Haus und Grundstück zu staubsaugen. Wenn Sie keine Lust haben, eine teure Staubsaugermaschine zu kaufen, können Sie natürlich auch alles mieten. Und bei eBay gibt es ja auch Staubsauger zu sehen. Die müssen dann alle zusammen geputzt werden. 3. Wer braucht schon Staubsauger?</p>
---	--	--	--

Abb. 15: Vier Inferenzbeispiele von causal-fw2 : ein politischer Artikel, ein Kochrezept, ein wissenschaftsjournalistischer Bericht und ein Haushaltsratgeber. Prompt hier und in folgenden Beispielen stets *kursiv*.

Diese Ausgaben zeigen, dass das Grundmodell dazu in der Lage ist, Struktur und den Aufbau typischer deutscher Textstrukturen zu imitieren und plausible Ausgaben zu erzeugen, jedoch teils inkohärente Anweisungen und unpassende Begriffe ausgibt. Die Schemata werden korrekt induziert, die Promptadhärenz ist vergleichsweise hoch. Zudem sind alle Ergebnisse grammatisch und syntaktisch korrekt und Begriffe verschiedener semantischer Klassen (z.B. Zutaten, Verbindungswörter und wissenschaftliche Konzepte) werden abgesehen von ihrer Bedeutung grundsätzlich auch fehlerfrei eingesetzt. Durchgeführte Vergleiche mit einem vortrainierten englischsprachigen GPT-2-small-Modell bestätigen ähnliche Ergebnisse.

Versucht man, das antikausale Modell auf identische Art und Weise zu testen, enttäuschen die Ergebnisse. Denn das Modell sagt den Text vorher, auf den das Rezept folgt – im Fall von Websites oft entweder ein weiteres unzusammenhängendes Rezept oder wie in Abb. 16 eine typische Einleitung für ein Rezept.^[12] Stattdessen sollte wie in Abb. 17 ein am Ende des gewünschten zu erzeugenden Textes stehender Teil übergeben werden, um eine passende Generation zu gewährleisten.

<p>Kaum ein Gericht schmeckt so gut wie ein griechischer Salat. Das sind zum Beispiel einfache, griechische Pellkartoffeln mit Käse. Dazu passt ein griechischer Salat sehr gut. Auch mit Fleisch oder Fisch schmeckt dieser Salat ziemlich gut. Rezept: Griechischer Salat Zutaten:</p>
--

Abb. 16: Fehlerhaftes Prompting `anticausal-fw2`.

^[11]Für die OSCAR-basierten Modelle wurde der t-Test für die beiden Korpora (OSCAR bzw. Wikipedia) separat mit je 200k Batches ausgeführt, der *p*-Wert beträgt hier 4.78×10^{-8} bzw. 4.32×10^{-145} , die Effektstärke *d* liegt bei 0.017 bzw. 0.081.

^[12]Im vorherigen Modell wurde hier repetitiver regulatorischer Text (AGBs, Cookie-Banner oder Impressumstexte) ausgegeben. Diese Art von Textfragmenten kam auch in anderer Nutzung der Modelle vor und war auf die Eigenschaften des Korpus zurückzuführen.

<p>von Kunstern und Künstlerinnen über Werke von Künstlern. Die Museen bieten auch Zeugnisse aus der Geschichte des Landes an.</p> <p>Nach Aufhebung der Sanktionen, auf die sich Regierung und Parlament in Washington am Freitag geeinigt haben, könnte es weitere staatliche Beihilfen geben. Zuvor sollen sich bereits mehrere US-Politiker gegen diese Forderungen gestellt haben. Die Regierung steht sich auf einen Entwurf ein, der im April in das Plenum eingebracht wird. Sollte es darüber keine Klarheit geben, soll eine Ausschusssitzung in den USA weitere Gestalte zur Entscheidung treffen.</p> <p>Die USA haben sich offenbar auf eine engere Zusammenarbeit mit ihren Partnern verständigt. Die geplanten US-Wertpapierkäufe sollen von Regierung, Pensionsfonds und anderen Banken refinanziert werden. Weitere Zusagen stehen bereit, wie Vertreter der amerikanischen Regierung am Freitag mitteilten.</p>	<p>Zutaten für Kräuterbrühe</p> <ul style="list-style-type: none"> - Kümmel, z.B. Majoran aus der Mühle - 100 g Parmesan - Pfeffer aus der Mühle - 100 g Parmesan, frisch - Salz - 150 g Kohlrabi, trocken <p>Zubereitung</p> <p>Für die Brühe die Zwiebeln schälen und achteln. Den Knoblauch fein hacken, mit der Brühe verrühren und zum Kochen bringen. Die Zwiebel, den Lauch, den Parmesan, Salz und Pfeffer aus der Mühle unterrühren und 3 bis 4 Minuten dünsten. Mit der Brühe übergießen und ca. 5 Minuten köcheln lassen. In der Zwischenzeit die Kräuter waschen und in mundgerechte Stücke schneiden. Die Kräuter in einen Schüssel geben und das Salz unterrühren. Die Brühe in einem Topf unter Röhren erwärmen, bis sich alles aufgelöst hat. Die Brühe durch ein Sieb gießen. Mit frischen Kräutern servieren. Guten Appetit!</p>	<p>Experten für Lebensökologie und Gesundheit zusammen.</p> <p>Einer aktuellen Studie zufolge ist Übergewicht einer der Hauptursachen für Männer, die die Deutsche Physikalische Gesellschaft in Berlin nun wissenschaftlich untersucht hat. Im Sommer wird vielen Frauen der Verlust auf Arbeit, Schlaf und Training zu Qual, Kälte, Hitze und dicke Matratzen sind quälend. Kohlenhydrate in Kombination mit einer kalorienarmen Ernährung können aber Grund für Rückenschmerzen bei Frauen sein - die Ursache ist unklar. Wenn wir weniger Fett essen, spart unser Körper mehr Kalorien ein. Welche Eier sind dagegen gut? Das haben Wissenschaftler leider mehr als 300 Frauen getestet. Das Ergebnis fällt bislang sehr unterschiedlich aus. Wer sich vollwertig ernährt, verdient mehr: Frauen, die vermehrt „Süßes“ suchen, weisen deutlich höhere Werte auf, wie Forschende der Hochschule Stralsund herausfanden.</p>	<p>1. Sonderabfall mit dem Sperrmüll:</p> <p>Nicht solten landet der Müll mit Elektroschrott. Das bedeutet viel Verpackungsmüll.</p> <p>2. In den Müll zu werfen: Alte Geräte</p> <p>werden oft über den Hausmüll im Sperrmüll entsorgt. Das sollte nachgedacht werden.</p> <p>3. Gesammelter Müll: Wo der Plastikmüll wirklich die Umwelt verschmutzt, kommt der Müll oft mit dem Hausmüll in den Abfall – das ist nicht gut für die Umwelt.</p> <p>4. Ressourcen sparen: Gar nicht so einfach den Plastikmüll zu reduzieren, um die Umwelt zu schonen.</p> <p>5. Sammeln statt wegwerfen: So viel Plastik ist täglich im eigenen Müll zu finden. Wer dagegen etwas für die Umwelt tun möchte, fährt in den Wald undräumt den Restmüll mit Plastikmüll auf. Hier braucht sich niemand die Finger schmutzig zu machen.</p> <p>6. Klein anfangen: Um auszumisten,</p>
--	--	---	--

Abb. 17: Vier Inferenzbeispiele von `anticausal-fv2`.

Diese Beispiele legen ebenfalls ein Verständnis der globalen und lokalen Struktur, aber begrenztes Verständnis der konkreten Semantik der Textstrukturen nahe. Die Ausgaben scheinen im Vergleich repetitiver zu sein, ansonsten lernt das antikausale Modell ebenfalls den typischen Aufbau.

5.6. Finetuning

Um die Generalisierungsfähigkeit der Modelle zu vergleichen und domänenspezifischere Texte erzeugen zu können, wurden beide Grundmodelle im überwachten Finetuning auf zwei verschiedene Datensätze trainiert. Das Finetuning verläuft dabei gleich zum bisherigen Training, jedoch wird die Lernrate η auf 6×10^{-5} fixiert und p_{drop} aufgrund des kleineren Datensatzes zur Verhinderung von Überanpassung auf 0.1 erhöht.

5.6.1. Deutsche Gesetzestexte

Der Gesamtkorpus der geltenden deutschen Bundesgesetzgebung wird vom Deutschen Bundestag in einem offiziellen Git-Repository[53] veröffentlicht. Die heruntergeladenen 170MB Markdown-Textdateien wurden in Rust tokenisiert, die Tokens dann im üblichen 90-10-Split auf 29.6M Trainings- und 3.3M Validierungstokens aufgeteilt. Der Trainingsverlauf wird in Abb. 18 gezeigt. Es ist zu erkennen, dass anfänglich aufgrund des Dropouts der Validierungsverlust stärker sinkt als der Trainingsverlust, etwa nach Schritt +1k^[13] durch Overfitting der Validierungsverlust stagniert, während der Trainingsverlust weiter abnimmt.

5.6.2. Plenarprotokolle des Deutschen Bundestags

Ebenfalls vom Deutschen Bundestag herausgegeben werden die Plenarprotokolle aller Bundestagssitzungen seit 1949 im XML-Format[54]. Die 895MB umfassenden XML-Dateien wurden in ein Textdateiformat umstrukturiert und analog der Split in 170M Trainings- und 18.9M Validierungstokens ausgeführt. Das Training wurde bis auf Erhöhung der Trainingsschritte durch eine größere Datenmenge analog zu Abschnitt 5.6.1

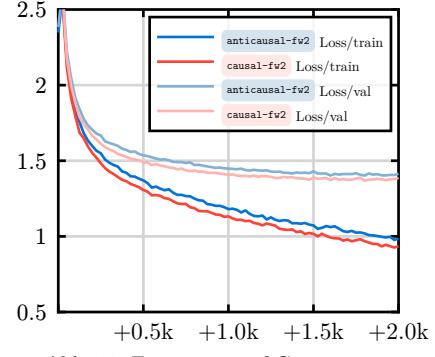


Abb. 18: Finetuning auf Gesetzestexte

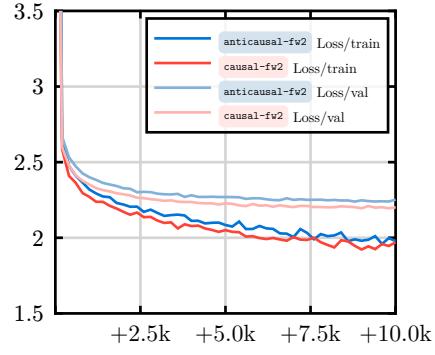


Abb. 19: Finetuning auf Plenarprotokolle

(3) Die Studierenden kommen nach Ablauf der Regelstudienzeit mit der Hochschulprüfung begonnen werden. Wenn diese Prüfung nicht bestanden ist, so kann die entsprechende Prüfung als der technischen Ausbildung erforderlich ist. Die Art und Weise der Prüfung ist auf Vorschlag der Hochschule durch Rechtsverordnung zu bestimmen. § 14 Abs. 2 des Bundesgleichstellungsgesetzes ist entsprechend anzuwenden; § 3 Absatz 4 ist entsprechend anzuwenden.

§ 6 Eignungsprüfung

(1) Soweit in dieser Verordnung nicht etwas anderes bestimmt ist, ist auch eine Abschlussprüfung einschließlich einer Eignungsprüfung nach den Vorschriften des zweiten Abschnitts nach § 78a Absatz 1 und § 78e als Bestandteil der Hochschulprüfung. Das §§ 8 und 9 über die Prüfungsanforderungen nach § 8 und die Prüfung und Prüfungsrecht der Hochschulen bleiben unberührt. (2) Die Prüfungen nach Ablauf der in § 2 genannten Fristen (Fristen) beginnen nach erfolgreichem Abschluss der Hochschulprüfung auf dem Gebiet der Zahntechnik des Landes, indem die Hochschulen auf Grund landesrechtlicher Vorschriften bestehen. Die Fachhochschulen sind von § 2 a) ausgenommen.

Abb. 20: Antikausal generiertes Gesetz.

^[13]Die Schritte werden als zusätzlich zu den vorherigen gezählt, der absolute Trainingsschritt ist hier 300k + 1k = 301'000.

ausgeführt; Abb. 19 zeigt die Resultate, zu beachten seien die geänderte Skalen.

Abb. 20 und Abb. 22 zeigen zwei exemplarische Ausgaben der respektiven antikausalen Modelle. Das auf Plenarprotokollen trainierte Modell schafft es dabei, im Kontext des verwendeten Prompts zu bleiben, jedoch nicht, die Bundeskanzlerin als Frau zu bezeichnen. Dies ist als wiedergegebene Verzerrung der Trainingsdaten erklärbar, da in nur vier (Merkel I-IV) der 19 Wahlperioden das Amt durch eine Frau bekleidet wurde.

5.7. Vergleich

	FW2	Gesetze	Protokolle
antikausal	∅ Verlust	3.3951	1.4499
	SD Verlust	0.5485	0.5704
	∅ Genauigkeit	4.158%	27.614% 14.215%
kausal	∅ Verlust	3.3629	1.4124
	SD Verlust	0.5462	0.5626
	∅ Genauigkeit	4.277%	28.528% 14.791%
Gesamt	∅ Δ Verlust	0.0322	0.0374
	p-Wert	1.64×10^{-39}	2.43×10^{-49}
	Effektstärke d	0.0588	0.0661
	Korr.koeff. ρ	0.995735	0.996789
			0.998504

Abb. 21: Berechnete statistische Kennwerte für je 100k Batches.

Abb. 21 zeigt statistische Kennwerte zu Modellkausalitäten und Korpora im Vergleich. Dabei wurden stets 100k Batches aus dem Validierungssatz gewählt und die Modelle evaluiert.

Groß-Geran) (CDU/CSU):
Sehr geehrte Frau Präsidentin! Meine lieben Kolleginnen und Kollegen! Ein Problem, das noch nicht geregelt wurde, ist der Prüfauftrag, den sowohl der Bundesrat als auch der Bundesrat und der Bundesregierung vorgegeben haben. Wir können uns der Bundesregierung wünschen, dass sie das, was sie in ihrer vorletzten Legislaturperiode und in der letzten Legislaturperiode beschlossen hat, in dieser Legislaturperiode beibehalten werden, die vorher im Bundesrat und der Bundesregierung formuliert und besprochen worden sind. Ich glaube darüber brauchen wir heute nicht zu reden.
(Horst Friedrich [Bayreuth] [FDP]: Jetzt doch nicht. Frau Kollegin)

Das Problem, um das es heute geht, liegt in den Jahren der rot-grünen Bundesregierung. Mit der Vorratsdatenspeicherung, von der Sie, Frau Kollegin Kopp, geredet haben, ist genau das Gegenteil dessen eingetreten, was wir uns in der Koalitionsvereinbarung vorgenommen haben.
(Gudrun Kopp [FDP]: Was?)
In der heutigen Zeit ist technologischer Fortschritt wichtiger denn je.

Frau Präsidentin! Meine Damen und Herren! Meine Damen, meine Herren, wir hören immer wieder auf die Einlassung, deswegen habe ich den Ausspruch des Herrn Bundeskanzlers immer noch nicht mitbekommen. Frau Bundeskanzler, hier sollten Sie den Mut haben, ins Detail zu gehen. Ich empfehle Ihnen, ihn nachzulesen. Ich weiß, ich habe ihn noch nicht gelesen.
(Beifall bei Abgeordneten der LINKEN)
Ich würde Ihnen empfehlen, ihn einmal zu lesen. Das können Sie bei der FDP immer noch nicht mitbekommen!
(Heiterkeit bei der CDU/CSU)

Das Wort gibt mir Gelegenheit, etwas zu dem zu sagen, was wir in der Bundesrepublik Deutschland entwickelt haben. Worauf kann man sich in der Bundesrepublik Deutschland freuen? Sie sind zufrieden damit, dass wir den wirtschaftlichen Aufstieg und den wirtschaftlichen Aufstieg geschafft haben, auch in der Bundesrepublik Deutschland.
(Bernhard Brinkmann [Hildesheim] [SPD]: Ach!
Frau Bundeskanzlerin, Sie lassen die Menschen im Regen stehen!)

Abb. 22: Antikausale Plenardebatten.

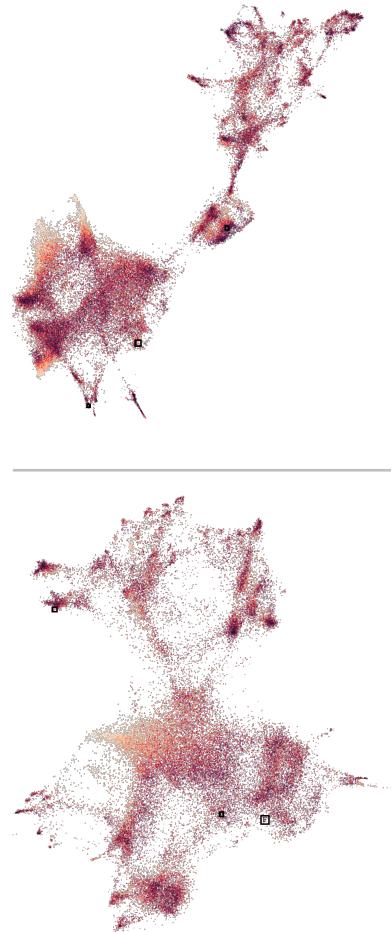


Abb. 23: Das 2D-Embedding von `anticausal-fw2` (oben) und `causal-fw2` (unten). Jeder der 50256 Punkte repräsentiert ein Token, eingefärbt nach dessen ID. Erstellt mit Pillow[56].

diese $\mathbb{R}^{50256 \times 768}$ -Matrix enthält semantische Informationen, die Aufschlüsse über die Funktion des Modells bieten können.

Um diese Daten anschaulich visualisieren zu können, wurde ein Python[57]-Programm geschrieben, um die Vektorliste mithilfe des Dimensionalitätsreduktionsalgorithmus PaCMAP[58] auf drei sowie zwei Dimensionen zu reduzieren. Der Algorithmus zeichnet sich durch die Fähigkeit aus, sowohl globale (wie PCA[59]) als auch lokale (wie t-SNE[60] oder UMAP[61]) Strukturen auf den niedrigdimensionalen Raum abbilden zu können[62].

Abb. 23 zeigt die Räume des neuen direktionalkomplementären Modellpaars. Durch das Betrachten spezifischer Cluster dieser Räume lassen sich Informationen über die sprach- und korpuspezifische Repräsentation der Tokens im Modell gewinnen. Diese können syntaktischer, grammatischer oder semantischer Natur sein: erstere enthalten die Klasse der Suffixcluster, welche insbesondere **causal-fw2** nach der Eigenschaft partitionieren, mit einem Leerzeichen zu enden; zweitere umfassen unterschiedlich deklinierte Verben sowie Nomen unterschiedlicher Kasus, während als Beispiele für letztere die ebenfalls geometrisch segmentierbaren Cluster deutscher Vornamen, geografischer Bezeichnungen oder Temporalwörter zu nennen sind. Abb. 27 zeigt für vier der Modelle je drei beispielhafte und nennenswerte Cluster.

6.1.2. Scree-Plots und konkatenierte Räume

In Abb. 25 wird für die Embedding-Räume der verbleibende unerklärte Varianzanteil einer Hauptkomponentenanalyse (PCA) mit x Komponenten dargestellt, oben absolut, unten relativ zur Kontrolllinie einer zufällig initialisierten Embedding-Matrix. Im Vergleich zu typischen Ellenbogendiagrammen ist der geringe erklärte Varianzanteil erkennbar; insbesondere unten kann zudem erkannt werden, dass durch Finetuning stärkere lineare Korrelationen zwischen Embedding-Dimensionen entstehen.

Durch Konkatenation der Embedding-Vektoren des antikausalen und kausalen Modells wird jedem Token ein $768 \times 2 = 1536$ -dimensionaler Vektor zugeordnet. Der entstehende Embedding-Raum kann nun ebenfalls dimensionaliätsreduziert werden und wird in Abb. 26 dargestellt. Stärker als zuvor und emergent werden die Tokens in zwei Cluster segmentiert; das linke scheint primär vollständige Wörter und -kombinationen zu beinhalten, während das rechte unvollständige Tokens und Nomen zusammenfasst.

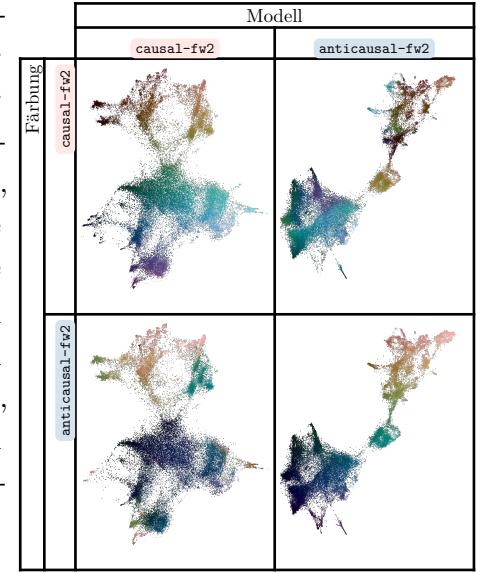


Abb. 24: Die Embedding-Räume; jedes Token eingefärbt nach Position des jeweils anderen Embedding-Raums.

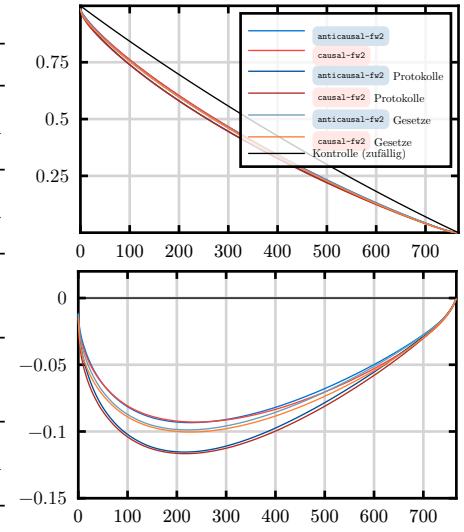


Abb. 25: Scree-Plots der Embedding-Räume.

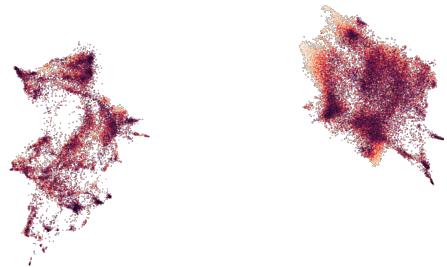


Abb. 26: 2D-Embedding-Raum des konkatenierten Raums.

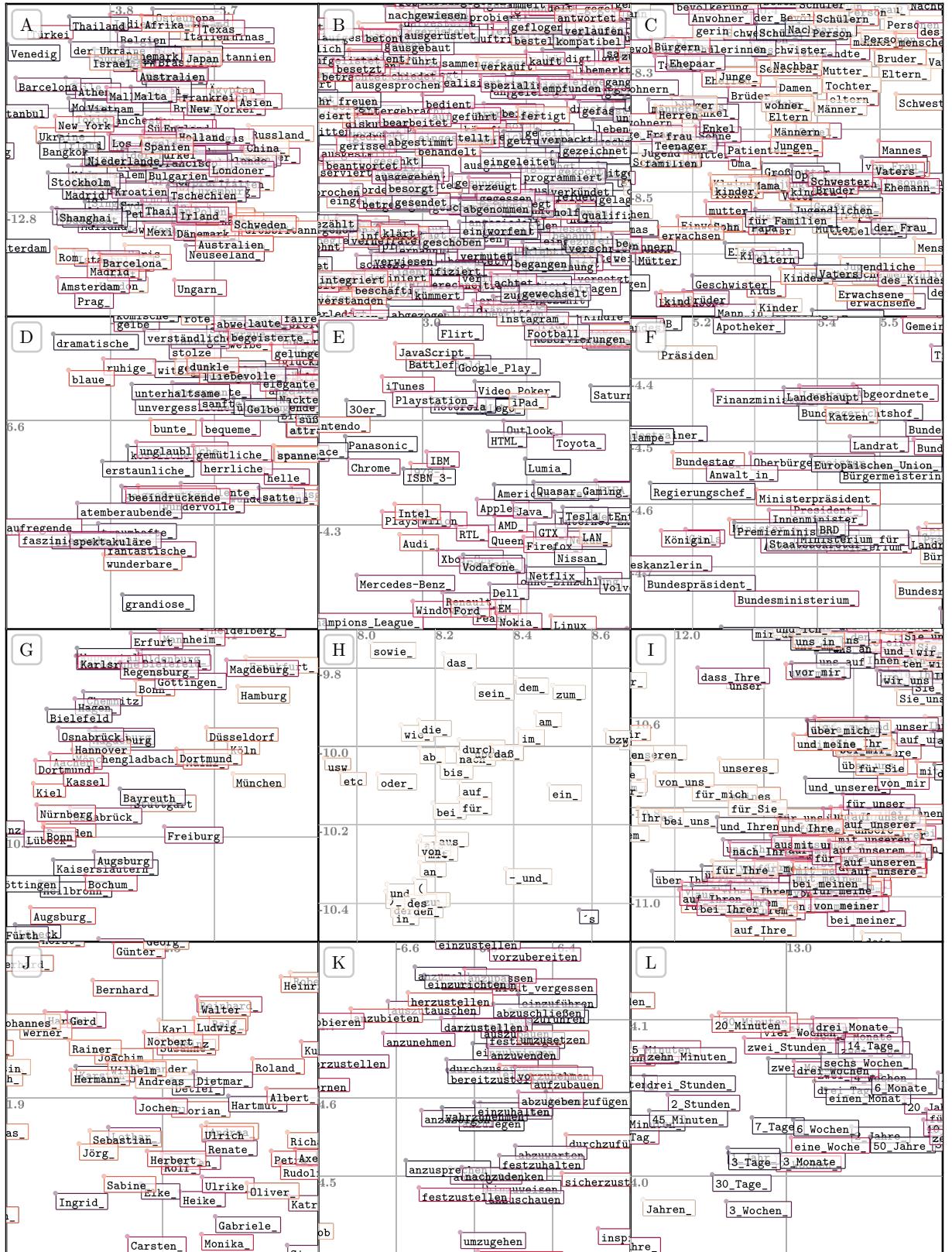


Abb. 27: (A-F) Zwölf Cluster aus den dimensionalitätsreduzierten latenten Token-Embedding-Räumen der Modelle.

A-C: `anticausal-fw2`, D-F: `causal-fw2`, G-I: `anticausal-fw2-laws1`, J-L: `causal-fw2-plenar1`.

A: Länder und Städte im Ausland. B: Verben im Partizip II. C: Soziokulturelle Beschreibungen von Menschen. D: Attributive zumeist positive Adjektive. E: Marken und Technologien. F: Politische Ämter und Institutionen. G: Städte in Deutschland. H: Kurze deutsche Funktionswörter. I: Personalpronomen und Präpositionen in Possessivphrasen. J: nach Geschlecht separierte deutsche Vornamen. K: zu-Infinitivkonstrukte. L: Quantifizierte Zeitabschnitte. Erstellt mit Pillow[56].

6.1.3. Kanonische Korrelationsanalyse

Eine quantitativer Auswertungsmethode ist die kanonische Korrelationsanalyse (CCA), welche ebenfalls über SciPy ausgeführt wurde. Hierzu werden die 50'256 Vektoren des latenten Embedding-Raums jeder Direktonalität als Datenpunkte betrachtet, um für jede Direktonalität eine Linearkombinationen ihrer Embedding-Dimensionen zu finden, sodass die angewandten Linearkombinationen maximal korrelieren. Das erste solche Paar von Linearkombinationen korreliert mit $\rho \approx 0.9852$, sodass davon auszugehen ist, dass es Tokeneigenschaften gibt, die beide Embedding-Räume repräsentieren. Abb. 28 zeigt für die ersten zehn Komponenten die extremal eingeordneten Tokens.

Abb. 28: Die zehn ersten CCA-Komponenten und Tokens mit Minimal- oder Maximalwerten pro Komponente und Direktionalitt.

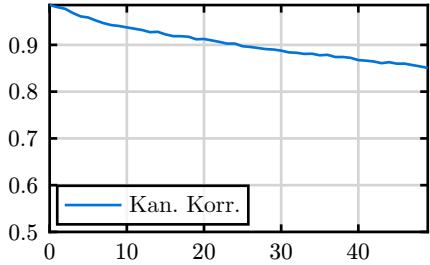


Abb. 29: Korrelation der Paaren von Linearkombinationen nach CCA.

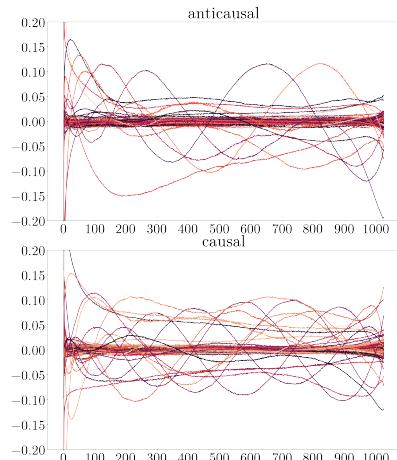


Abb. 30: Gelernte Werte des Positions-Embedding des Modellpaars, erstellt mit Matplotlib[63].

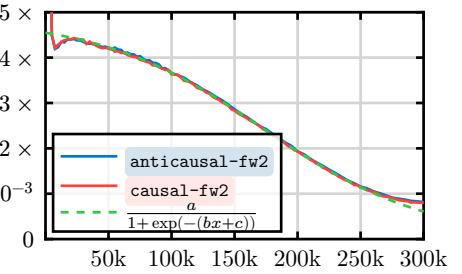


Abb. 31: Arithmetisches Mittel der MADs über alle Dimensionen im Laufe des Trainings samt logistischer Regression mit $R^2 = 0.9995$ im Intervall [50k 250k]

6.2. Positions-Embedding

Während das Token-Embedding jedem Token einen Vektor im Embedding-Raum zuweist, assoziiert das Positions-Embedding jeder Position im Kontextfenster einen solchen. So können die Vorhersagen des Modells von der absoluten Position der Tokens abhängen. Aufgrund des von Natur aus heuristischen Tokenisierungsprozesses und der häufigen Austauschbarkeit von Begriffen in natürlicher Sprache indiziert eine „Rauheit“ oder ein Rauschen eine fehlende Generalisierung des Modells auf die Daten. In Abb. 30 repräsentiert jede Kurve eine Dimension, die Abszisse die absolute Position des jeweiligen Tokens. Die relative Glattheit der Graphen verrät, dass die Positions-Embeddings beider Modelle parameterinsensitiv sind, was eine Generalisierungsfähigkeit nahelegt. Eine quantitative Auswertung der Rauheit über die mittlere absolute Distanz konsekutiver Elemente (MAD) konnte (im Gegensatz zum OSCAR-Modellpaar) keinen signifikanten Unterschied feststellen. Abb. 31 zeigt die MAD im Trainingsverlauf.

7. Webanwendung

Zum Zwecke der Nutzung und Visualisierung der in diesem Projekt trainierten Modelle wurde die interaktive Webanwendung [64] entwickelt. Der Quelltext[65] ist quelloffen und frei lizenziert.

7.1. Architektur

Der in der Webanwendung verwendete Technologiestack umfasst Svelte[66] samt SvelteKit[67] als Framework mit TypeScript[68,69] im Frontend. Zum Zweck der Daten- und Verantwortungsséparation gibt es zwei in Python[57] mit FastAPI[70] geschriebene Backends, die im Code als `shallow-backend` und `deep-backend` bezeichnet werden. `shallow-backend` läuft dabei auf einem externen Linux-Server ([71]), während `deep-backend` auf dem hochschuleigenen Server ausgeführt wird. Reine Datenbankaufrufe wie die Abfrage von Korpusbeispielen für ein bestimmtes Token werden direkt vom `shallow-backend`-Webserver beantwortet, während Inferenz- und andere Modellanfragen über ihn relaisartig an `deep-backend` weitergeleitet werden. Zur bidirektionalen Kommunikation werden Websockets[72] verwendet, wobei das `shallow-backend` eine eigene Clientenverwaltung beinhaltet muss, um an jeden Client nur die angefragten Tokens zurückzusenden.

7.2. Funktionen von Unterseiten (Auswahl)

Es folgt eine kurze Beschreibung dreier Unterseiten der Webanwendung, welche die interaktiven Funktionen dieser exemplarisch illustrieren.

7.2.1. Unterseite `/token/[id]`

Diese Seite enthält Informationen zum Token der ID `id`. Diese visualisiert wie in Abb. 32 mithilfe von d3.js[73] den Tokenstammbaum, d.h., die vollständige Ableitung bis hin zu Basistokens, über ein Dendrogramm. Es werden auch alle aus `id` abgeleiteten Tokens („Kinder“) gezeigt und über eine Beispielansicht kann das Token im Kontext des Trainingskorpus betrachtet werden. Die nächsten Nachbarn in den Embedding-Räumen können betrachtet und verglichen werden, ebenso die Inferenz beider Modelle mit dem Token als Eingabe.

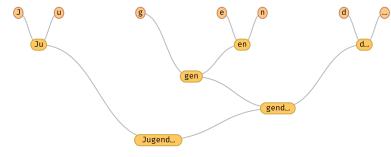


Abb. 32: Tokenstammbaum des Tokens `Jugend`.

7.2.2. Unterseite `/token/embedding-space`

Visualisiert den latenten Embedding-Raum des ausgewählten Modells. Dabei kann sowohl der zweidimensionale (wie in Abb. 23) als auch der dreidimensionale Raum dargestellt und navigiert werden. Die grafische Oberfläche und interaktive Visualisierung werden in deck.gl[74] GPU-beschleunigt realisiert, was auch flüssige Animationen zwischen verschiedenen Repräsentationen ermöglicht. Neben der Färbung nach Token-ID wie in bisherigen Abb. kann jeder Punkt nach verschiedenen Klassen eingefärbt werden. Eine Legende beschreibt die Farbbe bedeutung; ein Histogramm zeigt die Verteilung über alle Tokens auf. Durch Anklicken eines Punktes können Tokens ausgewählt und hervorgehoben werden, sodass Cluster evident werden.

7.2.3. Unterseite `/chat`

Enthält das Haupt-Inferenzinterface. Auf dieser Seite können das kausale und antikausale Modell genutzt werden, um Texte in respektive gegen die Leserichtung zu vervollständigen. Ebenfalls können die Tokenwahrscheinlichkeiten für ein beliebiges Token angezeigt werden.

8. Fazit und Ausblick

In diesem Projekt ist es mir gelungen, eigene direktionalkomplementäre große Sprachmodelle zu trainieren. Die beiden auf GPT-2 basierenden Modelle `causal-fw2` und `anticausal-fw2` illustrieren bereits jetzt Unterschiede in der richtungsabhängigen Textverarbeitung. Die interaktive Webanwendung leistet zusammen mit den gezeigten Statistiken und der komparativen Visualisierung latenter Modellräume einen Beitrag in Richtung erklärbarer KI und zeigt fundamentale Eigenschaften der deutschen (im Spezifischen) und menschlichen Sprache (im Allgemeinen) auf.

Es konnte gezeigt werden, dass das Training von großen deutschsprachigen antikausalen Sprachmodellen möglich ist und dass ihre Generalisierungsfähigkeit zur Inferenzzeit mit kausalen Sprachmodellen vergleichbar ist. Zugleich konnten elementare Unterschiede in der Repräsentation der Embedding-Schichten, sowohl im Token-Embedding als auch der Positionsencodierung, gemessen und statistisch ausgewertet werden.

Direkt geplant ist die Nutzung dieser Modelle als anschauliche selbst trainierte deutschsprachige kleine LLMs für die Hochschullehre der Hochschule Stralsund. Da in Gänze auf lokalen Servern der Hochschule im Sinne der *Edge AI*[75] ausgeführt, kann das Projekt auch als Beitrag zur zweiten[76] der fünf *Grand Challenges der Informatik 2025* [77,78] der GI betrachtet werden.

Die Richtungssensitivität verschiedener Datensätze im Finetuning konnte ebenfalls evaluiert werden. Insgesamt weisen kausale Modelle für alle angeführten Anwendungsgebiete bei gleicher Anzahl von Trainingsschritten und identischer Architektur einen signifikant geringeren Verlust auf; der Leistungsunterschied der beiden Modelle variiert zwischen Domänen und Datensätzen. Diese Erkenntnis ist kompatibel mit dem intuitiven menschlichen Verständnis von Text und Textinferenz. Der introspektive Ansatz der Analyse antikausaler Grundmodelle komplementiert bestehende praktische Ansätze und wirft zugleich neue Fragen bezüglich Textgerichtetheit und den textbezogenen dahinterliegenden kausalen Denk- und Handlungsprozessen auf.

Dennoch ist das Projekt nur ein anfänglicher Schritt auf dem Weg zu praktisch einsetzbaren direktionalkomplementären Sprachmodellen und der vollständigen Analyse von Unterschieden und Gemeinsamkeiten kausaler und antikausaler autoregressiver Textinferenz.

In Zukunft kann der Fokus dabei darauf liegen, weitere neue Erkenntnisse, auch im Bereich der Computerlinguistik, bezüglich Direktionalität natürlicher Sprache durch Training und Auswertung weiterer Sprachmodelle und Datensätze zu gewinnen. Hierzu wird ein Finetuning der Modelle auf den Projekt Gutenberg-Datensatz fiktionaler Literatur[79] angestrebt. Es gilt auch, die latenten Räume dieser spezialisierten Modelle zu visualisieren sowie durch Eingabe weiterer Tokensequenzen ins Modell auch kontextualisierte Eingaben räumlich einzuordnen. Auch die statistischen Analysen werden weitergeführt, um semantische direktionalitätsbezogene Korrelationen auswerten und genauer interpretieren zu können.

Geplant wird auch ein Training eines Mamba[80]-Modells auf Grundlage von `mamba.py`[81] zur Untersuchung der Modellunabhängigkeit der getroffenen Aussagen über Direktionalität. Theoretisch geht die Suche nach kogenerativer „Zusammenarbeit“ der trainierten Modelle weiter, indem überprüft wird, inwiefern die beispielsweise abwechselnde Tokeninferenz basierend auf einem gegebenen Eingabefragment zu Reduktion der direktionalen Perplexität führen kann. Zudem wird die Webanwendung aktiv weiterentwickelt, etwa um weiterführende Inferenzmodi auf Grundlage von Constrained Generation[82] oder Beam Search[83] durch eine offene Implementierung bereitzustellen. Unter der Annahme der Gültigkeit neuronaler Skalierungsge setze[84,85] auf die präsentierten Ergebnisse stellt die antikausale Rückinferenz bereits jetzt eine vielversprechende neuartige Möglichkeit der KI-Nutzung dar.

9. Danksagung

Ich danke meinem Projektbetreuer, Herrn Prof. Dr. André Grüning, für wertvolle Hinweise zum Training von KI-Modellen und weiterführenden Ausarbeitungsideen und für eine fachliche Überprüfung der vorliegenden Arbeit.

Zudem danke ich meinem Vater, Norman Wojak, für eine sprachliche und gestalterische Überprüfung der Arbeit. Ich danke der Hochschule Stralsund sowie meinem Projektbetreuer für die Bereitstellung des für Training und Inferenz genutzte GPU-Cluster und das zur Projektarbeit benötigte Material. Analog danke ich meiner Vorgesetzten im FJN (Freiwilliges Jahr in Wissenschaft, Technik und Nachhaltigkeit), Silke Krumrey, dafür, an der Hochschule am Projekt arbeiten zu können.

Ich danke auch allen Entwickler:innen der in der Open-Source-Anwendung und für das Modelltraining genutzten freien Softwarebibliotheken, ohne die die Ausarbeitung in ihrer aktuellen Form nicht möglich gewesen wäre. So danke ich auch allen Wissenschaftler:innen, insbesondere den Autor:innen direkter und transitiver Referenzen dieser Arbeit, ohne deren Einsatz die moderne KI-Entwicklung ausgeblieben wäre.

10. Quellen- und Literaturverzeichnis

Sofern nicht anders angegeben, handelt es sich bei allen Abbildungen um eigene.

- [1] A. Vaswani *u. a.*, „Attention is All you Need“, in *Neural Information Processing Systems*, 2017.
- [2] W. X. Zhao *u. a.*, „A Survey of Large Language Models“, 2024, Verfügbar unter: <https://arxiv.org/abs/2303.18223>
- [3] C.-C. Lin, A. Jaech, X. Li, M. R. Gormley, und J. Eisner, „Limitations of Autoregressive Models and Their Alternatives“, 2021, Verfügbar unter: <https://arxiv.org/abs/2010.11939>
- [4] S. Gong *u. a.*, „Scaling Diffusion Language Models via Adaptation from Autoregressive Models“, 2024, Verfügbar unter: <https://arxiv.org/abs/2410.17891>
- [5] Z. ul Abideen, „Autoregressive Models for Natural Language Processing“. Zugriffen: 12. Februar 2025. Verfügbar unter: <https://medium.com/%40zaiinn440/autoregressive-models-for-natural-language-processing-b95e5f933e1f>
- [6] T. A. Chang und B. K. Bergen, „Language Model Behavior: A Comprehensive Survey“, *Computational Linguistics*, Bd. 50, Nr. 1, S. 293–350, 2024, doi: 10.1162/coli_a_00492.
- [7] S. Yu *u. a.*, „Reverse Modeling in Large Language Models“, 2024, Verfügbar unter: <https://arxiv.org/abs/2410.09817>
- [8] M. V. Koroteev, „BERT: A Review of Applications in Natural Language Processing and Understanding“, 2021, Verfügbar unter: <https://arxiv.org/abs/2103.11943>
- [9] M. Bavarian *u. a.*, „Efficient Training of Language Models to Fill in the Middle“, 2022, Verfügbar unter: <https://arxiv.org/abs/2207.14255>
- [10] L. Berglund *u. a.*, „The Reversal Curse: LLMs trained on "A is B" fail to learn "B is A"“, 2024, Verfügbar unter: <https://arxiv.org/abs/2309.12288>
- [11] O. Golovneva, Z. Allen-Zhu, J. Weston, und S. Sukhbaatar, „Reverse Training to Nurse the Reversal Curse“, 2024, Verfügbar unter: <https://arxiv.org/abs/2403.13799>
- [12] J. C.-Y. Chen *u. a.*, „Reverse Thinking Makes LLMs Stronger Reasoners“, 2025, Verfügbar unter: <https://arxiv.org/abs/2411.19865>
- [13] H. Naveed *u. a.*, „A Comprehensive Overview of Large Language Models“, 2024, Verfügbar unter: <https://arxiv.org/abs/2307.06435>
- [14] S. Wolfram, „What is ChatGPT Doing... and Why Does It Work?“, 2023, Zugriffen: 4. Dezember 2024. Verfügbar unter: <https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work/>

- [15] A. Karpathy, „Deep Dive into LLMs like ChatGPT“. Zugegriffen: 8. Februar 2025. Verfügbar unter: <https://www.youtube.com/watch?v=7xTGNNLPyMI>
- [16] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, und I. Sutskever, „Language Models are Unsupervised Multitask Learners“, 2019. Verfügbar unter: <https://api.semanticscholar.org/CorpusID:160025533>
- [17] A. Karpathy, „NanoGPT“, *Github repository*, 2022, Zugegriffen: 10. November 2024. Verfügbar unter: <https://github.com/karpathy/nanoGPT>
- [18] X. Liu, H.-F. Yu, I. Dhillon, und C.-J. Hsieh, „Learning to Encode Position for Transformer with Continuous Dynamical Model“, 2020, Verfügbar unter: <https://arxiv.org/abs/2003.09229>
- [19] J. L. Ba, „Layer normalization“, *arXiv preprint arXiv:1607.06450*, 2016.
- [20] R. Xiong u. a., „On layer normalization in the transformer architecture“, in *International Conference on Machine Learning*, 2020, S. 10524–10533.
- [21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, und R. Salakhutdinov, „Dropout: a simple way to prevent neural networks from overfitting“, *The journal of machine learning research*, Bd. 15, Nr. 1, S. 1929–1958, 2014.
- [22] P. Baldi und P. J. Sadowski, „Understanding dropout“, *Advances in neural information processing systems*, Bd. 26, 2013.
- [23] M. Wornow, „Transformer Math (Part 1) - Counting Model Parameters“. Zugegriffen: 13. Februar 2025. Verfügbar unter: <https://michaelwornow.net/2024/01/18/counting-params-in-transformer>
- [24] H. Yao, D.-l. Zhu, B. Jiang, und P. Yu, „Negative log likelihood ratio loss for deep neural network classification“, in *Proceedings of the Future Technologies Conference (FTC) 2019: Volume 1*, 2020, S. 276–282.
- [25] P. Zhou, X. Xie, Z. Lin, und S. Yan, „Towards Understanding Convergence and Generalization of AdamW“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 46, Nr. 9, S. 6486–6493, 2024, doi: 10.1109/TPAMI.2024.3382294.
- [26] I. Loshchilov und F. Hutter, „Decoupled Weight Decay Regularization“, 2019, Verfügbar unter: <https://arxiv.org/abs/1711.05101>
- [27] D. P. Kingma, „Adam: A method for stochastic optimization“, *arXiv preprint arXiv:1412.6980*, 2014.
- [28] I. Loshchilov und F. Hutter, „SGDR: Stochastic Gradient Descent with Warm Restarts“, 2017, Verfügbar unter: <https://arxiv.org/abs/1608.03983>

- [29] Y. Bengio, P. Simard, und P. Frasconi, „Learning long-term dependencies with gradient descent is difficult“, *IEEE Transactions on Neural Networks*, Bd. 5, Nr. 2, S. 157–166, 1994, doi: 10.1109/72.279181.
- [30] R. Pascanu, „Understanding the exploding gradient problem“, *arXiv preprint arXiv:1211.5063*, 2012.
- [31] J. Zhang, T. He, S. Sra, und A. Jadbabaie, „Why gradient clipping accelerates training: A theoretical justification for adaptivity“, *arXiv preprint arXiv:1905.11881*, 2019.
- [32] A. Krogh und J. Hertz, „A simple weight decay can improve generalization“, *Advances in neural information processing systems*, Bd. 4, 1991.
- [33] G. Zhang, C. Wang, B. Xu, und R. Grosse, „Three Mechanisms of Weight Decay Regularization“, 2018, Verfügbar unter: <https://arxiv.org/abs/1810.12281>
- [34] J. Chen *u. a.*, „LLMArena: Assessing Capabilities of Large Language Models in Dynamic Multi-Agent Environments“, 2024, Verfügbar unter: <https://arxiv.org/abs/2402.16499>
- [35] G. Penedo *u. a.*, „FineWeb2: A sparkling update with 1000s of languages“. Verfügbar unter: <https://huggingface.co/datasets/HuggingFaceFW/fineweb-2>
- [36] J. Abadji, P. Ortiz Suarez, L. Romary, und B. Sagot, „Towards a Cleaner Document-Oriented Multilingual Crawled Corpus“, *arXiv e-prints*, S. arXiv:2201.06642, Jan. 2022.
- [37] Wikimedia Deutschland e. V., „Deutsche Wikipedia, Datenextraktion vom 1. November 2024“. Verfügbar unter: <https://dumps.wikimedia.org/dewiki/>
- [38] Y. Shibata *u. a.*, „Byte pair encoding: A text compression scheme that accelerates pattern matching“, 1999.
- [39] A. Macijauskas, „Tokenizers deep dive“, 2024, Zugegriffen: 20. Februar 2025. Verfügbar unter: <https://augustasmacijauskas.github.io/personal-website/posts/tokenizers-deep-dive/tokenizers-deep-dive.html>
- [40] A. Paszke *u. a.*, „Automatic differentiation in PyTorch“, in *NIPS-W*, 2017.
- [41] L.-K. Schulz, „Cloud Computing for Education: Design and Implementation of a Platform Solution for Dynamic Provisioning of Computing Power and Software with Kubernetes“.
- [42] S. Li *u. a.*, „PyTorch Distributed: Experiences on Accelerating Data Parallel Training“, 2020, Verfügbar unter: <https://arxiv.org/abs/2006.15704>
- [43] F. Chaubard, D. Eddy, und M. J. Kochenderfer, „Beyond Gradient Averaging in Parallel Optimization: Improved Robustness through Gradient Agreement Filtering“, 2024, Verfügbar unter: <https://arxiv.org/abs/2412.18052>

- [44] X. Li, Z.-Q. J. Xu, und Z. Zhang, „Loss Spike in Training Neural Networks“. Verfügbar unter: <https://arxiv.org/abs/2305.12133>
- [45] S. Takase, S. Kiyono, S. Kobayashi, und J. Suzuki, „Spike No More: Stabilizing the Pre-training of Large Language Models“, 2024, Verfügbar unter: <https://arxiv.org/abs/2312.16903>
- [46] K. Nishida, K. Nishida, und K. Saito, „Initialization of Large Language Models via Reparameterization to Mitigate Loss Spikes“. Verfügbar unter: <https://arxiv.org/abs/2410.05052>
- [47] P. Jaccard, „Étude comparative de la distribution florale dans une portion des Alpes et des Jura“, *Bull Soc Vaudoise Sci Nat*, Bd. 37, S. 547–579, 1901.
- [48] „Der t-Test“, in *Quantitative Methoden: Einführung in die Statistik*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, S. 43–117. doi: 10.1007/978-3-540-33308-1_3.
- [49] P. Virtanen *u. a.*, „SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python“, *Nature Methods*, Bd. 17, S. 261–272, 2020, doi: 10.1038/s41592-019-0686-2.
- [50] L. Kennedy-Shaffer, „Before $p < 0.05$ to Beyond $p < 0.05$: Using History to Contextualize p-Values and Significance Testing“, *The American Statistician*, Bd. 73, Nr. sup1, S. 82–90, 2019, doi: 10.1080/00031305.2018.1537891.
- [51] S. Goodman, „A Dirty Dozen: Twelve P-Value Misconceptions“, *Seminars in Hematology*, Bd. 45, Nr. 3, S. 135–140, 2008, doi: <https://doi.org/10.1053/j.seminhematol.2008.04.003>.
- [52] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2. Aufl. New York: Routledge, 1988. doi: 10.4324/9780203771587.
- [53] Deutscher Bundestag, „BundesGit Gesetze Tools“. Zugriffen: 16. Februar 2025. Verfügbar unter: <https://github.com/bundestag/gesetze>
- [54] Deutscher Bundestag, „Drucksachen der 1. - 19. Wahlperiode“, 2021. Zugriffen: 16. Februar 2025. Verfügbar unter: <https://www.bundestag.de/services/opendata>
- [55] A. G. Asuero, A. Sayago, und A. González, „The correlation coefficient: An overview“, *Critical reviews in analytical chemistry*, Bd. 36, Nr. 1, S. 41–59, 2006.
- [56] A. Clark, „Pillow (PIL Fork) Documentation“. Verfügbar unter: <https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf>
- [57] G. Van Rossum und F. L. Drake Jr, *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.
- [58] Y. Wang, H. Huang, C. Rudin, und Y. Shaposhnik, „Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMap,

- and PaCMAP for Data Visualization“, *Journal of Machine Learning Research*, Bd. 22, Nr. 201, S. 1–73, 2021, Verfügbar unter: <http://jmlr.org/papers/v22/20-1061.html>
- [59] „Hauptkomponentenanalyse (PCA)“, in *Multivariate Statistik in der Ökologie: Eine Einführung*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, S. 105–123. doi: 10.1007/978-3-540-37706-1_9.
- [60] L. v. d. Maaten und G. Hinton, „Visualizing data using t-SNE“, *Journal of machine learning research*, Bd. 9, Nr. Nov, S. 2579–2605, 2008.
- [61] L. McInnes, J. Healy, und J. Melville, „UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction“, 2020, Verfügbar unter: <https://arxiv.org/abs/1802.03426>
- [62] M. Gruber, „Why you should not rely on t-SNE, UMAP or TriMAP“. Zugegriffen: 8. Februar 2025. Verfügbar unter: <https://towardsdatascience.com/why-you-should-not-rely-on-t-sne-umap-or-trimap-f8f5dc333e59/>
- [63] J. D. Hunter, „Matplotlib: A 2D graphics environment“, *Computing in Science & Engineering*, Bd. 9, Nr. 3, S. 90–95, 2007, doi: 10.1109/MCSE.2007.55.
- [64] L. Blume, „DEversAI“. Zugegriffen: 9. Februar 2025. Verfügbar unter: <https://deversai.vercel.app/>
- [65] L. Blume, „Quelltext für das Jugend forscht-Projekt DEversAI“. Zugegriffen: 17. Februar 2025. Verfügbar unter: <https://github.com/leo848/deversai>
- [66] R. Harris, A. Faubert, T. L. Hau, B. McCann, und andere, „Svelte – cybernetically enhanced web apps“. Zugegriffen: 3. Januar 2024. Verfügbar unter: <https://svelte.dev/>
- [67] R. Harris, A. Faubert, T. L. Hau, B. McCann, und andere, „SvelteKit: Web development, streamlined.“. Zugegriffen: 2. Januar 2024. Verfügbar unter: <https://kit.svelte.dev/>
- [68] A. Hejlsberg, „TypeScript: JavaScript with types“. Zugegriffen: 2. Januar 2024. Verfügbar unter: <https://www.typescriptlang.org/>
- [69] G. Bierman, M. Abadi, und M. Torgersen, „Understanding TypeScript“, in *European Conference on Object-Oriented Programming*, 2014, S. 257–281.
- [70] S. Ramírez, „FastAPI“. Verfügbar unter: <https://github.com/fastapi/fastapi>
- [71] J. Pasche, „Uberspace - Hosting auf Asteroids“. Zugegriffen: 17. Februar 2025. Verfügbar unter: <https://uberspace.de/de/>
- [72] I. Fette und A. Melnikov, „The websocket protocol“, 2011.
- [73] M. Bostock, „D3.js - Data-Driven Documents“. Verfügbar unter: <http://d3js.org/>

- [74] Uber, „deck.gl: WebGL powered geospatial visualization layers“. Zugegriffen: 30. Dezember 2024. Verfügbar unter: <https://deck.gl/>
- [75] R. Singh und S. S. Gill, „Edge AI: A survey“, *Internet of Things and Cyber-Physical Systems*, Bd. 3, S. 71–92, 2023, doi: <https://doi.org/10.1016/j.iotcps.2023.02.004>.
- [76] Gesellschaft für Informatik, „Dezentrale KI: unabhängig und individuell“. Zugegriffen: 28. März 2025. Verfügbar unter: <https://gi.de/grand-challenges/edge-ai>
- [77] Gesellschaft für Informatik, „Die Grand Challenges der Informatik 2025“. Zugegriffen: 28. März 2025. Verfügbar unter: <https://gi.de/grand-challenges>
- [78] Gesellschaft für Informatik, „Grand Challenges 2025: Vor diesen Herausforderungen steht die Informatik“. Zugegriffen: 28. März 2025. Verfügbar unter: <https://gi.de/meldung/grand-challenges-2025>
- [79] H. Reuters, „Projekt Gutenberg“. Zugegriffen: 17. Februar 2025. Verfügbar unter: <https://www.projekt-gutenberg.org/>
- [80] A. Gu und T. Dao, „Mamba: Linear-Time Sequence Modeling with Selective State Spaces“, 2024, Verfügbar unter: <https://arxiv.org/abs/2312.00752>
- [81] A. Torres-Leguet, „mamba.py: A simple, hackable and efficient Mamba implementation in pure PyTorch and MLX.“. Verfügbar unter: <https://github.com/alexndrTL/mamba.py>
- [82] D. Banerjee, T. Suresh, S. Ugare, S. Misailovic, und G. Singh, „CRANE: Reasoning with constrained LLM generation“, 2025, Verfügbar unter: <https://arxiv.org/abs/2502.09061>
- [83] S. Wiseman und A. M. Rush, „Sequence-to-Sequence Learning as Beam-Search Optimization“, 2016, Verfügbar unter: <https://arxiv.org/abs/1606.02960>
- [84] J. Kaplan *u. a.*, „Scaling Laws for Neural Language Models“, 2020, Verfügbar unter: <https://arxiv.org/abs/2001.08361>
- [85] Y. Bahri, E. Dyer, J. Kaplan, J. Lee, und U. Sharma, „Explaining neural scaling laws“, *Proceedings of the National Academy of Sciences*, Bd. 121, Nr. 27, S. e2311878121, 2024, doi: 10.1073/pnas.2311878121.