Informatik Spektrum
https://doi.org/10.1007/s00287-024-01571-1

HAUPTBEITRAG

Prompt Engineering zur Generierung von Sprachlerninhalten

Sylvio Rüdian[1,2] · **Niels Pinkwart**[1]

Angenommen: 22. August 2024

Zusammenfassung
Spätestens seit der Veröffentlichung von ChatGPT sind generative Modelle ein integraler Bestandteil aktueller Innovationen. Durch die geschickte Kombination und Einbettung solcher Modelle können Tools entstehen, welche den Arbeitsalltag unterstützen. In diesem Beitrag werden generative Verfahren in den Bildungskontext überführt. Konkret wurde die Domäne des Sprachenlernens gewählt. Anhand des Szenarios eines Sprachlernkurses für Sprachlernanfänger*innen wird die Kombination generativer Modelle verdeutlicht, um die praktische Anwendung zu demonstrieren. Dazu wird das „Exploratory" und „Confirmatory" Prompt Engineering eingeführt, welches die Erstellung und Evaluation von Anfragen an sprachbasierte generative Modelle ermöglicht. Es werden Möglichkeiten zur Integration in Sprachlerntools erörtert, aber auch deren Eigenheiten und Grenzen. Darüber hinaus zeigt der Beitrag anhand einer Sequenzierungsstrategie, dass sprachbasierte generative Modelle nicht jede Instruktion befolgen und Alternativen notwendig sind.

Einleitung

Die Erstellung von Lernmaterialien ist ein aufwendiger Prozess, bei dem Lehrende nicht nur das zu vermittelnde Wissen beherrschen, sondern es auch kreativ und didaktisch angemessen aufbereiten müssen [1]. Sollen Lerninhalte digital umgesetzt werden, sind zudem Medienkompetenzen erforderlich [2]. Dabei ist nicht nur das Lernen ein hochkomplexer kognitiver Prozess [3], sondern auch die Erstellung von Lernmaterialien [4]. Technologien können Lehrende hierbei unterstützen.

In den letzten Jahren sind große Sprachmodelle populär geworden [5]. Sogenannte „Generative-Pretrained-Transformer-Modelle" (GPT) [6] erlauben es den Anwender*innen, Texte fortzuführen. In Kombination mit Instruktionen wurde ein instruktionsbasiertes Modell, bekannt als InstructGPT, trainiert [7]. Daraus entstandene Modelle wie GPT 3.5, besser bekannt als ChatGPT [8], ermöglichen die Erledigung textbezogener Aufgaben [7]. Auf den ersten Blick sind solche generativen Modelle eine großartige Grundlage für die Erstellung von Lerninhalten.

Gegenwärtig sind Ausgaben von Sprachmodellen textbasiert. Generative Verfahren sind jedoch nicht zwingend auf Texte beschränkt. Bildgeneratoren erlauben die Generierung von Bildern anhand von textuellen Eingaben [9]. Basierend auf neuronalen Netzen wurden Modelle trainiert, um Bilder zu generieren, die einer vorgegebenen Beschreibung entsprechen [10]. Die Technologie, die dahintersteckt, ist eine autoregressive Transformer-Architektur, die Bilder und Texte kombiniert, wodurch das Modell die Bildgenerierung ermöglicht [11]. Da generative Verfahren bislang auf eine Ausgabeart beschränkt sind (z. B. Texte oder Bilder), lohnt sich ein Blick auf Tools, welche generierten Output kombinieren.

Bei der Anwendung der Bildgenerierung im Bildungsbereich argumentieren Ringvold et al. [12], dass die Technologie einen Kompromiss zwischen den Möglichkeiten und Grenzen darstellt. Generative Verfahren können fehlerhafte Ergebnisse generieren oder Inhalte halluzinieren [13]. Daher besteht die Notwendigkeit der Evaluation der Ausgaben generativer Modelle, um zu überprüfen, ob daraus entstehende Tools nachhaltigen Einsatz finden können. Trotz der Flexibilität der Sprachmodelle (Large Language Models, LLMs), ist deren Evaluation eine Herausforderung [14]. Die Forschung ist hierbei noch jung und wenig fortgeschritten. Aufgrund einer fehlenden Systematik zur Evaluation von Ein- und Ausgaben aus LLMs wird das „Exploratory" und „Confirmatory" Prompt Engineering eingeführt, adaptiert von Tukeys Nomenklatur aus dem Bereich der Statistik [15]. Anschließend wird der Einsatz eines LLMs und die

✉ Sylvio Rüdian
ruediasy@informatik.hu-berlin.de

[1] Deutsches Forschungszentrum für künstliche Intelligenz (DFKI) – Berlin, Berlin, Deutschland

[2] Humboldt-Universität zu Berlin, Berlin, Deutschland

Published online: 24 September 2024

Evaluation anhand eines Fallbeispiels aus dem Sprachenlernen vorgestellt. Zuletzt werden anhand eines weiteren Beispiels potenzielle Grenzen von LLMs aufgezeigt und eine Alternative wird vorgeschlagen, welche symbolische KI-Verfahren in subsymbolische KI-Verfahren überführt, welche wiederum in komplexeren Szenarien Anwendung finden können.

Exploratory & Confirmatory Prompt Engineering

Zunächst muss ein Tool so konzipiert werden, dass einzelne Teilaufgaben entstehen, die von generativen Modellen übernommen werden können. Dazu wird ein gesamtheitliches Problem in Teilprobleme zerlegt. Dieses Prinzip ist in der Informatik als Modularisierung oder Dekomposition bekannt [16]. Die Teilprobleme, die sich durch generative Verfahren grundsätzlich lösen lassen könnten, müssen mithilfe von Prompts vorbereitet werden. „Prompts sind Anweisungen, die einem LLM gegeben werden, um Regeln durchzusetzen, Prozesse zu automatisieren und bestimmte Qualitäten (und Quantitäten) des generierten Outputs sicherzustellen“ [17]. Der Prozess der Erstellung solcher Prompts kann als Exploratory Prompt Engineering bezeichnet werden, da die Vorgehensweise explorativer Natur ist. Anhand eines Soll-Outputs kann ein Prompt Engineer zunächst die erwünschten Ausgaben definieren. Anschließend werden Instruktionen erstellt, mit deren Hilfe die gewünschten Ausgaben generiert werden können [17]. Die Instruktionen enthalten dabei die durchzuführenden Anweisungen, potenzielle Einschränkungen und die erwünschte Zielformatierung des Ausgabeformats. Prompts können parametrisiert werden, indem die Anweisung einen Platzhalter enthält, der sich beliebig austauschen lässt. Dadurch entsteht ein Prompt Template. Abb. 1 illustriert ein so konzipiertes Prompt Template, wobei die Teilanweisungen durch ein gewöhnliches Trennzeichen (z. B. Komma) getrennt werden. Alternativ können Einschränkungen, die Zielgruppe oder Formatierungen einem LLM durch Systemnachrichten übergeben werden (Few-Shot Prompting [18]).

Erfüllen die Ausgaben des generativen Modells die Soll-Anforderungen des Prompt Engineers für ausgewählte Beispiele, kann der Prozess des Exploratory Prompt Engineerings abgeschlossen und in das Confirmatory Prompt Engineering überführt werden. Während beim Exploratory Prompt Engineering der vielversprechendste Prompt identifiziert wird, werden beim Confirmatory Prompt Engineering die generierten Ausgaben für verschiedene Inhalte quantitativ evaluiert, welche den/die Platzhalter ersetzen [19].

Um LLMs zu evaluieren, werden normalerweise standardisierte Datensätze mit konkreten Metriken verwendet [20]. GSM8K, BoolQ, MMLU, WinoGrande, Codex HumanEval, BBQ oder ToxiGen sind lediglich eine Auswahl möglicher Benchmarks, welche in Evaluationen Anwendung finden [20]. Diese zielen jedoch nicht zwingend auf Evaluationskriterien ab, welche konkrete Prompts zielführend evaluieren, da Ein- und Ausgaben und Gütekriterien vordefiniert sind, um LLMs gänzlich zu evaluieren. Beispielsweise handelt es sich bei „ToxiGen“ um einen Datensatz, mit welchem volkverhetzende Aussagen getestet werden können [21]. Für den Einsatz von Prompts im Bildungskontext konnte bislang keine angemessene Metrik identifiziert werden [22].

Daher wird vorgeschlagen, eine Liste z. B. von Themen vorzubereiten, mit welchen der vorbereitete Prompt getestet wird. Damit kann ein Prompt Engineer die Hypothese überprüfen, ob die Ausgaben dem erwarteten Soll auch dann entsprechen, wenn sich der Inhalt der auszuführenden Anweisung ändert. Dazu muss eine Metrik definiert werden, anhand derer der generalisierbare Einsatz des Prompts quantitativ überprüft wird. Je nach Anweisung können unterschiedliche Metriken zum Einsatz kommen, z. B. die Korrektheit, die Angemessenheit oder weitere. Ist die Anweisung explorativer oder kreativer Natur, ist eine händische Evaluation notwendig. Nur wenn die Evaluation trivial ist (z. B. bei der Überprüfung von Rechtschreibfehlern) kann auf die Evaluation durch mehrere Evaluator*innen verzichtet werden. Tools können bei der Evaluation unterstützten. Sind hingegen konkrete Soll-Ausgaben vordefiniert, welche als Ausgabe bei Verwendung der Prompts zu erwarten sind, können diese auch automatisiert auf Überdeckung getestet werden. Anhand der Evaluation kann ein Prompt Engineer schlussfolgern, ob die Prompts überarbeitet werden müssen oder ob die Prompt Templates in einem komplexeren Tool Anwendung finden können. Im Fall der notwendigen Überarbeitung geht der Prozess in das Exploratory Prompt Engineering zurück. Damit kann ein Prompt iterativ optimiert werden. Entsprechen die Ausgaben des Prompts jedoch überwiegend den Erwartungen des Prompt Engineers, ist die Erprobung in einem komplexen Szenario möglich.

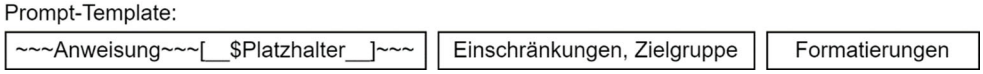

Abb. 1 Schematische Darstellung eines Prompt Templates

Fallbeispiel Sprachlernkurse

In diesem Abschnitt wird die Konzeption eines Generators erörtert, der die Generierung von Lernmaterialien für Sprachlernkurse erlaubt. Im Kontext des Sprachenlernens beschränkte sich der Einsatz von Technologien bis zur Veröffentlichung von LLMs hauptsächlich auf den Einsatz von „Natural Language Processing“ (NLP) [23]. Hierbei können syntaktische Abweichungen von einer Norm identifiziert werden, z. B., um Grammatikfehler zu finden [24], aber auch semantische Informationen lassen sich zur Weiterverarbeitung ableiten [25]. Basis waren bislang Textkorpora, welche mittels NLP die Bedeutung und Identifikation von Relationen erlauben [26]. Exemplarisch kann der COBUILD-Korpus verwendet werden, welcher 1980 für die englische Sprache konzipiert wurde [27]. Mittels einer Vielzahl von Beispielen können Lernende eine neue Sprache lernen [28]. Dennoch gleicht ein Korpus eher einem Nachschlagewerk [28] und ist nicht mit interaktiven Lernmaterialien gleichzusetzen. Die Erstellung von digitalen Sprachlernmaterialien ist hingegen ein aufwendiger Prozess, der Wissen, Kreativität und technische Fähigkeiten zur Umsetzung erfordert [29]. Sprachmodelle könnten Texte flexibel generieren und beschränken sich nicht auf das Finden bestehender Werke.

In diesem Paper werden Sprachlernkurse nach dem Vorbild von Rosetta Stone erstellt, deren Kurse anhand eines übermittelten Themas automatisch generiert werden. Lernende sollen eine Sprache durch visuelle Beziehungen erfahren, ohne dass sie textuelle Übersetzungen von einer Sprache in eine andere auswendig lernen müssen. Es handelt sich dabei um eine Umsetzung des generellen Nativismus [30], für den im Kontext des Spracherwerbs die Annahme besteht, dass das Sprachenlernen im Kindesalter durch die kognitive Entwicklung ermöglicht wird, während „nicht-sprachspezifische Prozesse und Kategorien bereits vorhanden sind“ [30]. Demnach kann eine Sprache potenziell durch Mustererkennung und logische Verknüpfungen im Kindesalter erlernt werden [31]. Das Lernen durch logische Verknüpfungen wird auch Abduktion genannt, wodurch Erkenntnisse erweitert werden, was beim logischen Schließen nicht immer der Fall sein muss [32]. Hierzu wird die wahrscheinlichste Lösung für eine unvollständige Menge von Beobachtungen gesucht [33]. Wenn Bilder als Grundlage für logische Schlussfolgerungen dienen, wird das angewandte Prinzip als visuelles abduktives Schlussfolgern bezeichnet [34]. Die Idee kann als Grundlage verwendet werden, um Sprachlernmaterialien zu generieren [35]. Die Aufgaben der Lernenden bestehen darin, Verknüpfungen zwischen Bildmaterialien und zugehörigen Wörtern und Sätzen zu identifizieren. Das Prinzip basiert darauf, dass die Lernenden aus der realen Welt die in den Bildmaterialien abgebildeten Objekte und Szenarien kennen und die Verknüpfung mit der zu lernenden Sprache herstellen können. Damit können Situationen imitiert werden, wobei sich Lernende in einer vereinfachten fremdsprachlichen Umgebung befinden und somit eine Sprache durch logische Verknüpfungen erlernen können.

Angenommen, die Lernenden kennen die visuelle Darstellung zu den Substantiven „Banane“, „Mango“ und „Zitrone“. Dann könnten die Lernenden eine Single-Choice-Frage, in welcher ein Bild einer Grapefruit zuzuordnen ist, bei der die Antwortmöglichkeiten aus den drei Optionen zzgl. dem Wort „Grapefruit“ bestehen, durch logisches Schließen beantworten, da dies das einzige unbekannte Wort ist (Ausschlussverfahren). Demzufolge können die Lernenden schlussfolgern, dass aufgrund des Wissens über die bereits bekannten Zuordnungen nur die verbliebene Variante korrekt sein kann. Werden die Sätze komplexer, z. B. „Ich muss die Mango und Grapefruit schälen.“, kann dieser allein aus der Kenntnis über Bilder von Mangos und/oder Grapefruits erkannt werden. Anhand dieses Prinzips kön-

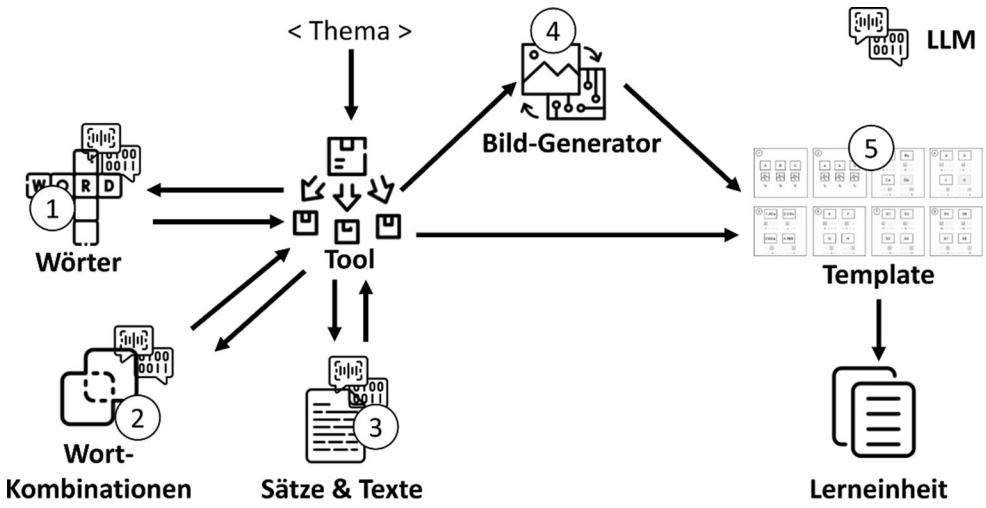

Abb. 2 Prozess zum Generieren einer Sprachlerneinheit mit vordefiniertem Template gemäß Rüdian & Pinkwart [36]. Icons stammen von Flaticon

nen Templates definiert werden, welche solch einer Logik im Rahmen einer Lernprogression folgen [36].

Die Herausforderung besteht nun darin, Lerninhalte zu finden, mit denen diese Logik umgesetzt werden kann, wobei LLMs unterstützen können. Gemäß dem Template von Rüdian & Pinkwart [36] kann ein LLM zum Identifizieren von Wörtern, deren Kombinationen und zum Formulieren von Sätzen und Texten mit vorgegebenen Wörtern verwendet werden. Zudem können Bildgenerierungsverfahren für die konkreten Wörter, Sätze und Texte neue Bilder generieren, welche anschließend als Illustrationen innerhalb des Templates verwendet werden können. Abb. 2 illustriert den Prozess zum Generieren einer Sprachlerneinheit.

Hierbei können die Schritte 1–3 von LLMs übernommen werden. Die beim explorativen Prompt Engineering entstandenen Prompt Templates sind in der Veröffentlichung von Rüdian & Pinkwart zu finden [36].

Das Confirmatory Prompt Engineering, durchgeführt mit 200 generierten Lerneinheiten kommt zu dem Ergebnis, dass die meisten generierten und ausgewählten Wörter sprachlich korrekt (98,5 %) und angemessen (97 %) waren. Wurden einzelne Sätze auf der Basis von zwei oder drei Wörtern formuliert, die mit einer maximalen Satzlänge von 12 Wörtern versehen waren, waren 80,5 % korrekt und 83,5 % angemessen. Texte waren überwiegend korrekt (90 %), und die meisten waren angemessen (93,5 %). Das Ergebnis zeigt, dass LLMs durchaus in der Lage sind, Teilaufgaben zu übernehmen. Die größte Schwäche wurde bei der Formulierung von Sätzen aus vorgegebenen Wörtern identifiziert. Es ist möglich, dass ein angepasster Prompt zu einem besseren Ergebnis führt. Allerdings ist nicht auszuschließen, dass ein LLM durch die Beschränkung der maximalen Wortanzahl je Satz nicht in der Lage ist, einen korrekten und angemessenen Satz zu formulieren, da diese Aufgabe ein hohes Maß an Kreativität erfordert.

Eine Herausforderung beim Einsatz von LLMs, deren Ausgaben weiterverarbeitet werden, ist deren Varianz. Zwar können Einschränkungen und Formatierungsvorgaben innerhalb eines Prompts definiert werden, z. B., dass die Ausgabe in einem vorgegebenen JSON-Format erfolgt. Dennoch ist dies kein Garant dafür, dass das durch ein LLM generierte Resultat den erwarteten Vorgaben immer entspricht. Es besteht demnach die Möglichkeit, dass eine Antwort nicht dem erwarteten Format entspricht, sodass sie nicht immer wie erwartet verarbeitet werden kann. Solche möglichen Fehler müssen abgefangen werden. Während des Experiments wurden in den Antworten Aufzählungen verwendet, obwohl die Abfrage ausdrücklich gefordert hat, keine Zahlen zu verwenden. In dem konkreten Tool generierte das LLM anstelle eines Satzes, wie gefordert, mehrere Sätze oder die Antwort begann mit „Ich werde drei Varianten ausprobieren:“, bevor die eigentliche Antwort begann. Selbst wenn die Anweisungen vorsahen, dass die Antwort einen Satz pro Zeile enthalten sollte, wurden häufig ganze Texte ohne Zeilenumbrüche generiert. Die Struktur und Formatierung der Antworten müssen demnach stets überprüft werden, um Abweichungen von der erwarteten Formatierung zu identifizieren, diese zur weiteren Verwendung automatisiert anzupassen oder die Generierung zu wiederholen.

Der generierte sprachliche Lerninhalt wurde bisher evaluiert. Die Evaluation generierter Bildmaterialien stand jedoch noch aus. Zu diesem Zweck wurden Bilder mit dem Bildgenerator DALL · E mini generiert [37]. Der Generator wurde exemplarisch ausgewählt, da dieser als Open Source zur Verfügung stand. Für jedes Wort (Substantive und Verben), für jeden Satz, der aus vorgegebenen Wörtern formuliert wurde, und für jeden Satz aus einem generierten zusammenhängenden Text wurden Bilder generiert, wobei die Wörter oder Sätze als direkte Prompts verwendet wurden. Im Folgenden wurde die Evaluation auf 113 Lerneinheiten beschränkt, die beim Confirmatory Prompt Engineering in [36] als angemessen und korrekt klassifiziert wurden.

Ziel der Evaluation ist die Überprüfung, ob die Bilder den Wörtern oder Sätzen eindeutig zugeordnet werden können. Nur wenn die Evaluator*innen eine eindeutige Lösung finden können, sind die generierten Bildmaterialien nützlich, um damit das Template zu füllen. Zu diesem Zweck wurde ein testähnliches Experiment vorbereitet, um festzustellen, ob die erzeugten Bilder mit den vorgegebenen Wörtern und Sätzen in Zusammenhang stehen. Für das Studiendesign mussten Evaluator*innen die richtige Beziehung zwischen Wörtern/Sätzen und Bildern identifizieren. Zugehörige Wörter jeder Wortklasse wurden gruppiert, da die Auswahl in den Aufgaben des Templates auf jene beschränkt ist. Ein Beispiel ist in Abb. 3 zu finden, wobei die durch Pfeile hervorgehobene Zuordnung von den Evaluator*innen identifiziert werden muss. Formulierte Sätze wurden zudem so gruppiert, dass sie dem Vorkommen in den Aufgaben entsprechen.

Die Evaluation der 113 Lerneinheiten benötigte im Durchschnitt 4 h pro Evaluator*in. Drei Evaluator*innen haben für die betrachteten Lerneinheiten die Bilder den

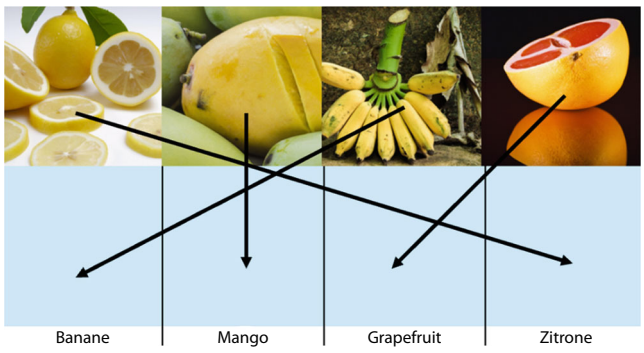

Abb. 3 Aufgabe zur Zuordnung von Bildern zu Wörtern

Tab. 1 Ergebnisse der Evaluation generierter Bilder: Korrekt identifizierte Relationen zwischen Bildern und Wörtern/Sätzen

Model	Klasse	Evaluator*in 1	Evaluator*in 2	Evaluator*in 3	∅
DALL·E mini	Substantive	0,831	0,792	0,788	0,804
	Verben	0,723	0,704	0,726	0,718
	Sätze	0,703	0,627	0,667	0,666
	Texte	0,799	0,739	0,759	0,766
	Lerneinheit	0,08	0,028	0,066	0,058

potenziell zugehörigen Wörtern und Sätzen zugeordnet. Da sich das Confirmatory Prompt Engineering in diesem Szenario auf die Eindeutigkeit der Zuordnungen beschränkt, wurde zur vereinfachten Darstellung auf die Auswertung weiterer Metriken verzichtet. Tab. 1 fasst die Ergebnisse der Evaluation zusammen. Die Krippendorffs Alpha Reliabilität (α) [38] wurde zwischen den Korrektheitsquoten gemessen, um festzustellen, ob die Evaluator*innen ähnliche Ergebnisse erzielen. Mit $\alpha = 0{,}991$ erzielt dieser einen sehr hohen Wert, sodass von einer hohen Übereinstimmung der Evaluator*innen ausgegangen werden kann. Insgesamt sind die erzielten Werte akzeptabel. Dennoch konnten 20–30 % der Bilder nicht eindeutig zugeordnet werden, was eine manuelle Nacharbeit erfordert, z. B. durch die erneute Generierung von Bildern. Insgesamt zeigt sich, dass von den 113 Lerneinheiten im Mittel 5–6 ohne weitere Anpassung weiterverwendet werden können. Die Auswertung zeigt, dass das Modell DALL·E mini zwar eine gute Grundlage bildet, um die visuellen Komponenten der Lernmaterialien zu generieren. Dennoch kann und sollte das Verfahren nur unterstützend eingesetzt werden, da eine vollständig automatisiert generierte Sprachlerneinheit sonst möglicherweise fehlerhaft oder unangemessen sein kann und Aufgaben nicht immer eindeutig lösbar sind.

Sequenzierungsstrategien

Nach dem Ansatz des aufgabenbasierten Sprachunterrichts (Task-Based Language Teaching, TBLT) [39] können verschiedene Aufgabentypen eingesetzt werden, um eine Lerneinheit zu entwickeln. Diese können von „einer kurzen Übung“ bis zu „einem komplexeren Szenario“ [39] reichen, um bestimmte Lernziele abzudecken [40]. Sie beinhalten, dass die Lernenden die Lerninhalte „verstehen, erstellen, verändern oder mit ihnen interagieren“ [41]. Die TBLT-Prinzipien basieren auf einem pädagogischen Fundament, wobei die Technologie eine entscheidende Rolle bei der Umsetzung dieser Prinzipien spielt [42]. Erfahrene Lehrende können Aufgaben entwerfen, die mit den zur Verfügung stehenden technischen Möglichkeiten umgesetzt werden können [43]. Die Auswahl von Aufgaben kann abhängig vom Wissensstand oder Interessen der Lernenden variieren. Daher sollten die Aufgaben für die Lernenden optimal ausgewählt und in eine ideale Reihenfolge gebracht werden, die zum bestmöglichen Lernergebnis führt.

Lerneinheiten, die basierend auf Templates generiert wurden, enthalten eine vordefinierte Mikrolernprogression. Werden Lerneinheiten wie beschrieben generiert, unterscheiden sie sich in ihren Inhalten, doch das angewandte Schema und die daraus resultierende Aufgabensequenz bleiben identisch. Lehrende neigen häufig dazu, in Kursen angewandte Lehrmethoden anzuwenden, die sie selbst präferieren [43]. Diese Einschränkung führt dazu, dass eine Einheitslösung angeboten wird, die als „Optimum“ für alle Lernenden propagiert wird. Die Lernenden müssen sich an die Gegebenheiten anpassen [44]. Dabei sind die Lernenden unterschiedlich und es gibt keine Einheitslösung, die für alle zum optimalen Lernerfolg führt [45]. Brusilovsky et al. [46] stellten fest, dass ein Onlinekurs, der für eine bestimmte Zielgruppe konzipiert wurde, „möglicherweise nicht für andere Nutzer*innen geeignet ist“. Daher ist es ratsam, Onlinekurse an die Lernenden anzupassen, um hohe Abbrecherquoten zu verringern [47]. Ein gutes Lehrkonzept berücksichtigt dabei die Bedürfnisse, Lernziele und Präferenzen der Lernenden [48]. Das führt jedoch da-

Tab. 2 Prompt zur Umsetzung des Karteikartenlernens (Flashcard Learning)

Typ	(Teil-)Prompt
Anweisung (1. Teil)	Angenommen, es gibt 10 Aufgaben, für welche angegeben ist, ob die Aufgabe richtig oder falsch gelöst wurde. Falls eine Aufgabe nicht richtig gelöst wurde, wiederhole sie am Ende:
Platzhalter	Aufgabe 1: richtig, Aufgabe 2: richtig, Aufgabe 3: falsch, Aufgabe 4: falsch Aufgabe 5: richtig, Aufgabe 6: richtig, Aufgabe 7: richtig, Aufgabe 8: falsch Aufgabe 9: richtig, Aufgabe 10: richtig
Anweisung (2. Teil)	Gib die gesamte Sequenz an, bis alle Aufgaben richtig gelöst wurden
Einschränkungen	Keine Erklärung
Formatierung	Trenne die Aufgaben durch Kommas

Tab. 3 Erwartete und generierte Ergebnisse, die bei der Anwendung des Prompts von Tab. 2 entstehen (Auswahl)

Erwartetes Ergebnis (Soll)	**Korrektheit**
Aufgabe 1, Aufgabe 2, Aufgabe 3, Aufgabe 4, Aufgabe 5, Aufgabe 6, Aufgabe 7 Aufgabe 8, Aufgabe 9, Aufgabe 10, *Aufgabe 3, Aufgabe 4, Aufgabe 8*	✓
Generiertes Ergebnis	**Korrektheit**
Aufgabe 1, Aufgabe 2, Aufgabe 3, Aufgabe 4, Aufgabe 5, Aufgabe 6 Aufgabe 7, Aufgabe 8, *Aufgabe 3, Aufgabe 4, Aufgabe 9, Aufgabe 10*	✗
Aufgabe 1, Aufgabe 2, Aufgabe 3, Aufgabe 4, Aufgabe 5, Aufgabe 6 Aufgabe 7, Aufgabe 8, Aufgabe 3, Aufgabe 4, *Aufgabe 9, Aufgabe 10,* Aufgabe 8	✗
Die gesamte Sequenz lautet: 1, 2, 3, 4, 5, 6, 7, 8, *3, 4, 9, 10*	✗
Aufgabe 1, Aufgabe 2, Aufgabe 3, Aufgabe 4, Aufgabe 5, Aufgabe 6 Aufgabe 7, Aufgabe 8, *Aufgabe 9, Aufgabe 3*, Aufgabe 4, Aufgabe 8	✗

zu, dass Aufgabensequenzen individuell angepasst werden müssen. Der Schlüssel hierzu sind sogenannte Sequenzierungsstrategien [49].

In diesem Abschnitt des Beitrags werden LLMs zunächst verwendet, um Sequenzen nach vordefinierten Strategien zu generieren. Die Demonstration beschränkt sich auf das Exploratory Prompt Engineering, zu welchem ein Beispiel illustriert wird, welches die Grenzen von LLMs verdeutlicht. Exemplarisch wurde das Karteikartenlernen als Strategie zur Ableitung einer optimalen Aufgabensequenz ausgewählt. Das Prinzip besteht daraus, dass Aufgaben, die nicht richtig gelöst wurden, wiederverwendet und zur Wiederholung an das Ende des Kurses gestellt werden [50]. Somit können Lernende beispielsweise effizient Vokabeln lernen und sie wiederholen, falls sie noch nicht gefestigt wurden.

Um das Karteikartenlernen für eine Sequenz von Aufgaben anzuwenden, wurde ein Prompt vorbereitet, welcher die Anweisung formuliert (Tab. 2). Der Prompt wurde um eine konkrete Sequenz ergänzt, auf welche die Strategie anzuwenden ist. Konkret handelt es sich um 10 exemplarische Aufgaben, von denen alle richtig gelöst wurden, bis auf die Aufgaben 3, 4 und 8. Nach der Beschreibung des Karteikartenlernens müssten diese Aufgaben (3, 4 und 8) die Sequenz ergänzen, da sie nicht richtig gelöst wurden und demnach wiederholt werden müssen.

Tab. 3 listet eine Auswahl von Ergebnissen auf, die bei der Anwendung des Prompts aus Tab. 2, angewandt auf das Modell GPT 3.5 (bekannt als eine der ersten Versionen von ChatGPT [8]) entstanden sind. Fehlerhafte Teilsequenzen wurden unterstrichen. Die Instruktion wurde nicht vollständig umgesetzt. Die letzte Ausgabe ist zwar vielversprechend, lässt jedoch die „Aufgabe 10“ aus.

Das Beispiel verdeutlicht, dass es Problemstellungen gibt, welche durch klassische LLMs zum Status quo nicht vollständig übernommen werden können, da ihre Fehlerrate zu hoch ist. Die Fähigkeit zur Beschreibung und Wiedergabe eines algorithmischen Verfahrens bedeutet nicht, dass ein Verfahren auch in der Lage ist, beschriebene Algorithmen gänzlich anzuwenden. Selbst wenn das Schlüsselwort „Karteikartenlernen“ in den Prompt integriert wird, führt dies zu keiner signifikanten Änderung der Ergebnisse. Wird hingegen ausschließlich die Anweisung zur Anwendung des Karteikartenlernens ohne Einschränkungen oder Formatierungsvorgaben verwendet, liefert das Modell ab und an eine korrekte Liste als Teil einer ausführlichen Erklärung. Auf Basis der bloßen Beschreibung innerhalb der Anweisung kann das LLM keine eindeutigen Resultate liefern, obwohl das erwartete Soll-Ergebnis eindeutig ist.

Im Folgenden wird demonstriert, dass die Generierung von Sequenzen, die konkrete Sequenzierungsstrategien verfolgen, mithilfe generativer Machine-Learning-Modelle möglich ist, als sogenannte subsymbolische KI. Zur Vereinfachung der Darstellung und zur Reduktion notwendiger Ressourcen wurde eine adaptierte Version des Karteikartenlernens umgesetzt. Diese unterscheidet sich im Vergleich zum traditionellen Karteikartenlernen darin, dass Aufgaben im Falle eines Fehlers von der Position i zur Wiederholung an die Position $i + 2$ verschoben werden. Durch diese Änderung ist eine lokale Sicht auf eine Teilsequenz ausreichend, ohne dass vollständige Sequenzen betrachtet werden müssen. Es ist jedoch davon auszugehen, dass, wenn die Generierung der adaptierten Version mithilfe eines generativen Modells möglich ist, auch die Imitation des traditionellen Karteikartenlernens funktioniert.

Grundsätzlich besteht die Frage, weshalb ein regelbasierter Ansatz in ein subsymbolisches KI-Verfahren überführt werden sollte. Während sich eine Sequenzierungsstrategie häufig durch triviale Regelwerke realisieren lässt, sind Kombinationen unterschiedlicher Strategien eine Herausforderung. Symbolische KI-Verfahren könnten Regelwerke kombinieren, doch wie sich im Kontext der Entwicklung von LLMs gezeigt hat, kann es sich lohnen, komplexe Modelle anhand einer Vielzahl von Beispieldaten zu trainieren, um neue Ausgaben zu generieren, die kombinierten Anweisungen folgen. Im Folgenden wird aufgezeigt, dass die Überführung in subsymbolische KI-Verfahren funktioniert. Die weitere Verarbeitung und Kombination von Regelwerken ist Teil künftiger Forschungsarbeiten.

Tab. 4 Zwei Sequenzen, bestehend aus 10 Aufgaben, in denen unterschiedliche Methoden angewandt werden, inkl. Performanz der Lernenden (R … Aufgabe richtig gelöst, F … Aufgabe nicht richtig gelöst, nicht näher bezeichnete Methoden: SCE, SCD, MC, B, H, W, SP)

Sequenz	Positionen	1	2	3	4	5	6	7	8	9	10
1	*Methode 1*	SCE	B	SCE	**MC**	W	**MC**	W	SP	SCE	B
	Antwort 1	R	R	R	**F**	F	R	R	R	R	R
2	*Methode 2*	**MC**	MC	**MC**	**SCE**	**H**	**SCE**	SCD	**B**	SP	**B**
	Antwort 2	**F**	R	R	**F**	R	R	R	**F**	R	F

Die Basis zur Erstellung eines generativen Verfahrens besteht aus Beispielen, die anhand der vorgegebenen Beschreibung synthetisch generiert werden. Tab. 4 illustriert zwei Sequenzen, in denen die adaptierte Version des Karteikartenlernens exemplarisch umgesetzt wurde.

Anstelle der algorithmischen Anwendung der Strategie wurde ein generatives Modell trainiert, das Sequenzen für die beschriebene Strategie generiert. Als Trainingsdaten wurden 3000 Beispielsequenzen generiert, welche der Beschreibung der Sequenzierungsstrategie folgen. Um mithilfe der synthetisch erzeugten Sequenzen ein Modell zu trainieren, wurden diese in Zeitseriendaten transformiert, mit der Idee, dass für eine eingehende Sequenz stets das nächste Element vorhergesagt wird.

Technisch wurde eine LSTM-Architektur (Long Short-Term Memory) umgesetzt. Diese können in praktischen Anwendungen häufig Muster aus Zeitreihendaten lernen [51]. Damit könnte eine Teilmenge einer Aufgabensequenz betrachtet werden und das Modell sagt das nächste Element voraus. Alternativ können rekurrente neuronale Netze (RNNs) verwendet werden, die jedoch nur lokale Abhängigkeiten berücksichtigen [52]. Um eine möglichst generische Architektur vorzustellen, die auch komplexere Sequenzierungsstrategien umsetzen kann, wurden LSTMs für die Umsetzung gewählt.

Das konkrete Modell wurde in Python3 mit TensorFlow implementiert. Es besteht aus einem bidirektionalen LSTM-Layer, der gewählt wurde, da er beim Training des Modells die Eingaben in beide Richtungen, vorwärts und rückwärts, verarbeitet [53]. 256 Knoten lieferten ein optimales Ergebnis. Dann folgte eine Dropout-Schicht, um ein potenzielles Overfitting [54] zu reduzieren, gefolgt von einem Dense Layer, der die finale Entscheidung ableitet. Abb. 4 illustriert den Loss während des Trainings zur Validierung, sowohl mit den Trainingsdaten als Testdaten (blau) als auch mit Testdaten, welche nicht zum Training verwendet wurden (orange). Es wird deutlich, dass das Modell ab Epoche ~1750 beginnt, zu lernen, da sich der Loss verringert. Das beste Modell ($loss$ = 1,55) wurde final ausgewählt. Anschließend wurde eine Kreuzvalidierung durchgeführt, um zu überprüfen, ob neue Sequenzen, die mithilfe des Modells generiert wurden, der Idee des adaptierten Karteikartenlernens entsprechen. Hierzu wurden keine Sequenzen verwendet, die Teil der Trainingsdaten waren. Die Genauigkeit (Precision) erreicht in der Kreuzvalidierung einen Wert von 100 %. Dies zeigt, dass generative Verfahren Sequenzierungsstrategien umsetzen können, wenn sie für diesen Zweck trainiert wurden.

Fazit

In diesem Beitrag wurden mithilfe von subsymbolischen KI-Verfahren strukturierte Lerninhalte abgeleitet, die wiederum in symbolischen KI-Verfahren Einsatz finden können. Hierzu wurde die Generierung von Lerninhalten für Sprachlernkurse fokussiert. Es wurde demonstriert, wie generative Modelle eingesetzt werden können, um komplexe Aufgaben wie die Erstellung von Sprachlernmaterialien zu unterstützen. Der Einsatz von LLMs bedarf einer gründlichen Vorbereitung und Überprüfung der generierten Ergeb-

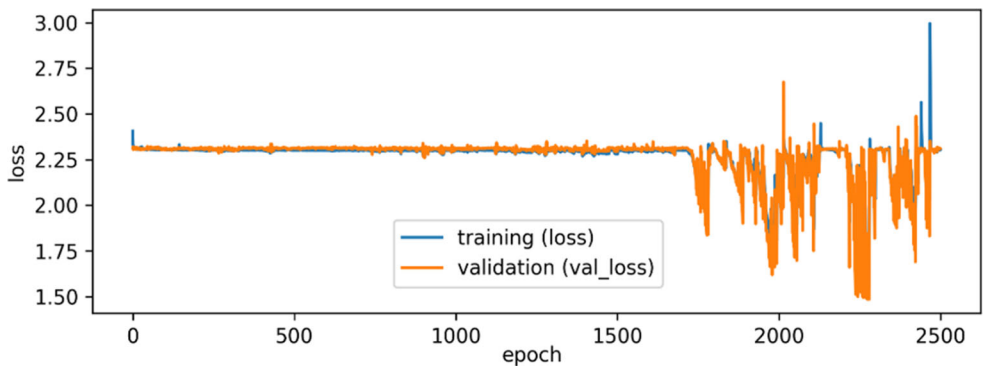

Abb. 4 Loss (Categorical Cross-Entropy) und Validierungs-Loss in 2500 Trainingsepochen

nisse, bevor diese als Teilkomponenten Einzug in Software finden können.

Allerdings hat das Beispiel zur Anwendung von Sequenzierungsstrategien auch verdeutlicht, dass Ergebnisse aus komplexen Anweisungen nicht immer den erwarteten Resultaten entsprechen müssen. Ein LLM ist in der Lage, sprachliche Komponenten wiederzugeben oder sie umzuformulieren. Beinhaltet mindestens einer der Texte, die zum Trainieren des LLM verwendet wurden, ein Beispiel zur Anwendung des Karteikartenlernens, kann dieses grundsätzlich auch adaptiv wiedergegeben werden. Ist die Antwort jedoch strukturell oder im Format limitiert, kann durch das betrachtete LLM zum Status quo nicht immer ein kausaler Zusammenhang aus der Beschreibung und dem zu erwartenden Ergebnis konstruiert werden.

Während die in diesem Beitrag verwendeten Templates vordefiniert und statisch waren, kann deren Anordnung mittels gelernter Sequenzierungsstrategien in ein subsymbolisches Verfahren überführt werden. Eine Sequenzierungsstrategie wurde exemplarisch umgesetzt. Die Kombination von symbolischen und subsymbolischen Verfahren wurde von Hochreiter [55] als einer der vielversprechendsten Ansätze für komplexe KI-Systeme klassifiziert. Der Ansatz zur Generierung von Lerninhalten ist vielversprechend. Je nach Zielsetzung können LLMs kombiniert mit symbolischer KI Anwendung finden, dennoch sind auch Verfahren nötig, welche für konkrete Zielsetzungen trainiert wurden, die durch LLMs zum Status quo nicht umsetzbar sind.

Das Lernen ist ein hochdimensionaler kognitiver Prozess [3]. Daher ist auch das personalisierte Lernen deutlich komplexer und bedarf der Betrachtung vieler Komponenten. Demnach ist weitere interdisziplinäre Forschung notwendig, welche die Bildungswissenschaften, Psychologie und Informatik kombiniert. Künftige Forschung wird zeigen, ob automatisiert generierte Lernmaterialien in Kombination mit individuellen Sequenzierungsstrategien die Lernenden dabei unterstützen können, bestmögliche Lernergebnisse zu erzielen.

Danksagung Diese Arbeit wurde vom Bundesministerium für Bildung und Forschung (BMBF) gefördert, Förderkennzeichen 16DHBQP058 (KI Campus 2.0) und 16DHBKI045 (IMPACT).

Funding Open Access funding enabled and organized by Projekt DEAL.

Literatur

1. Shermukhammadov B (2022) Creativity of a Teacher in an innovative educational environment. J High Educ Theory Pract 22(12)
2. Spector JM, De la Teja I (2001) Competencies for online teaching. ERIC Clearinghouse on Information & Technology, Syracuse
3. Estes W (2022) Handbook of learning and cognitive processes. Psychology Press
4. Freire MM (2013) Complex educational design: a course design model based on complexity. Campus-wide Inf Syst 30(3):174–185
5. Deng J, Lin Y (2022) The benefits and challenges of ChatGPT: An overview. Front Comput Intell Syst 2(2):81–83
6. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019) Language models are unsupervised multitask learners. Openai Blog 1(8)
7. Ouyang L, Wu J, Jiang X, Almeida D, Wainwright C, Mishkin P, Lowe R, al e (2022) Training language models to follow instructions with human feedback. Adv Neural Inf Process Syst 35:27730–27744
8. Koubaa A (2023) GPT-4 vs. GPT-3.5: A Concise Showdown
9. OpenAI Image generation. https://platform.openai.com/docs/guides/images/usage (Erstellt: 31. Jan. 2023). Zugegriffen: 3. Mai 2023
10. Fernandez P (2022) Technology behind text to image generators. library Hi Tech News 39(10):1–4
11. Ramesh A, Pavlov M, Goh G, Gray S, Voss C, Radford A, Sutskever I (2021) Zero-shot text-to-image generation. Int Conf Mach Learn: 8821–8831
12. Ringvold TA, Strand I, Haakonsen P, Strand KS (2023) AI text-to-image generation in Art and design teacher education: a creative tool or a hindrance to future creativity? In: International pupils' attitudes towards technology conference proceedings, Bd. 40
13. Rawte V, Sheth A, Das A (2023) A survey of hallucination in large foundation models. arXiv preprint
14. Chang Y, Wang X, Wang J, Wu Y, Zhu K, Chen H, Xie X (2023) A survey on evaluation of large language models. ACM
15. Tukey JW (1980) We need both exploratory and confirmatory. Am Stat 34(1):23–25
16. Miller TD, Elgard P (1998) Defining modules, modularity and modularization. In: IPS research seminar, Bd. 13. Aalborg Universiy Fuglsoe, Fuglsoe
17. White J, Fu Q, Hays S, Sandborn M, Olea C, Gilbert H, Schmidt DC (2023) A prompt pattern catalog to enhance prompt engineering with chatgpt. in arXiv:2302.11382
18. Logan RL IV, Balažević I, Wallace E, Petroni F, Singh S, Riedel S (2021) Cutting down on prompts and parameters: simple few-shot learning with language models. preprint:2106.13353, arXiv. " in
19. Rüdian S (2024) Exploratory and confirmatory prompt engineering. Educ Prompt Eng. https://doi.org/10.5281/zenodo.12549309
20. Chowdhery A, Narang S, Devlin J, Bosma M, Mishra G, Roberts A, Fiedel N (2023) Palm: scaling language modeling with pathways. J Mach Learn Res 24(240):1–113
21. Hartvigsen T, Gabriel S, Palangi H, Sap M, Ray D, Kamar E (2022) Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. arXiv:2203.09509

22. Laskar MTR, Bari MS, Rahman M, Bhuiyan MAH, Joty S, Huang JX (2023) A systematic study and comprehensive evaluation of chatgpt on benchmark datasets. arXiv:2305.18486
23. Meurers D (2012) Natural language processing and language learning. Encycl Appl Linguist: 4193–4205
24. Faltin AV (2003) Natural language processing tools for computer assisted language learning. Linguist Online 17(5)
25. Siddiqui Z (2018) English language teaching through NLP: techniques and methods. Res J Engl Lang Lit 6(2):181–184
26. Brunner A, Steyer K (2009) A model for corpus-driven exploration and presentation of multi-word expressions. In: NLP, Corpus Linguistics, Corpus Based Grammar Research
27. Krishnamurthy R (1996) Change and continuity at COBUILD (1986–1996). Eger J Engl Stud 1:61–79
28. Mukherjee J (2008) Anglistische Korpuslinguistik und Fremdsprachenforschung: Entwicklungslinien und Perspektiven. Z Fremdsprachenforsch 19:31–60
29. Tomlinson B (2023) Developing materials for language teaching. Bloomsbury Publishing
30. Höhle B (2002) Der Einstieg in die Grammatik: Die Rolle der Phonologie/Syntax-Schnittstelle für Sprachverarbeitung und Spracherwerb. Freie Universitat Berlin (Dissertation)
31. Bates E, MacWhinney B (1989) Functionalism and the competition model. Crosslinguistic Study Sentence Process 3:73–112
32. Paul G (1993) Approaches to abductive reasoning: an overview. Artif Intell Rev 7(2):109–152
33. Peirce CS (2014) Charles Sanders Peirce. Inf Theory (181)
34. Liang C, Wang W, Zhou T, Yang Y (2022) Visual abductive reasoning. In: Conference on Computer Vision and Pattern Recognition. IEEE/CVF, S 15565–15575
35. Dazzani V (2005) Learning and abduction. Semiotica (153):73–84
36. Rüdian S, Pinkwart N (2023) Auto-generated language learning online courses using generative AI models like ChatGPT. In: 21. Fachtagung Bildungstechnologien (DELFI) Aachen, S 65–76
37. Dayma B, Patil S, Cuenca P, Saifullah K, Abraham T, Khac PL, Melas L, Ghosh R (2021) Dall· e mini
38. Krippendorff K (2011) Computing Krippendorff's alpha-reliability. Departmental Papers (ASC), Pennsylvania
39. Ellis R (2003) Task-based language learning and teaching. Oxford University Press
40. Crookes G (1986) Task classification: a cross-disciplinary review
41. Nunan D (1989) Designing tasks for the communicative classroom. Cambridge University Press
42. Doughty CJ, Long MH (2003) Optimal psycholinguistic environments for distance foreign language learning. Lang Learn Technol 7(3):50–80
43. Gokalp M (2013) The effect of students' learning styles to their academic success. CE 4(10):627–632
44. Muir DJ (2001) Adapting online education to different learning styles. In: Building on the future. NECC 2001: national educational computing conference proceedings Chicaco. Bd. 22
45. Murphy RJ, Gray SA, Straja SR, Bogert MC (2004) Student learning preferences and teaching implications. J Dent Educ 68(8):859–866
46. Brusilovsky P, Eklund J, Schwarz E (1998) Web-based education for all: a tool for development adaptive courseware. In: Computer networks and ISDN systems, proceedings of seventh international world wide web conference, S 291–300
47. Aldowah H, Al-Samarraie H, Alzahrani AI, Alalwan N (2020) Factors affecting student dropout in MOOCs: a cause and effect decision-making model. J Comput High Educ 32:429–454
48. Hartnett M (2016) The importance of motivation in online learning. In: Motivation in online education. Springer, Singapore, S 5–32
49. Thomas S (2016) Future ready learning: peimagining the role of technology in education. In: National education technology plan. US Department of Education, Office of Educational Technology,
50. Teske K (2017) Duolingo. Calico J 34(3):393–401
51. Yu Y, Si X, Hu C, Zhang J (2019) A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput 31(7):1235–1270
52. Medsker L, Jain LC (1999) Recurrent neural networks: design and applications. CRC press
53. Schmidhuber J, Hochreiter S (1997) Long short-term memory. Neural Comput 9(8):1735–1780
54. Hawkins DM (2004) The problem of overfitting. J Chem Inf Comput Sci 44(1):1–12
55. Hochreiter S (2022) Toward a broad AI. Commun ACM 65(4):56–57