[Künstliche Intelligenz]

Erstellen von Sprachmodellen

Wir bauen ein Mini-Sprachmodell aus Märchensätzen

Es war einmal ein König …

Ein **Large Language Model**, also ein großes Sprachmodell, entsteht nicht auf einmal. Es durchläuft mehrere Schritte, die wir im Unterricht stark vereinfacht nachbauen.

1. ## Trainingsdaten sammeln

   Zuerst werden Trainingsdaten gesammelt. Das sind viele Texte, aus denen das Modell Sprachmuster lernen kann. In unserem Fall nutzen wir kurze, lustige Märchensätze.

2. ## Tokens bilden `Tokenization`

   Danach werden die Texte in kleine Einheiten zerlegt, die Tokens. Häufig sind das Wörter oder Wortteile. Aus dem Satz „Es war einmal ein König“ entstehen zum Beispiel die Tokens „Es“, „war“, „einmal“, „ein“, „König“.

3. ## Worteinbettung `Word Embedding`

   Die Tokens werden in Zahlen umgewandelt. Jedes Wort bekommt eine Art Zahlen-Steckbrief. Wörter mit ähnlicher Bedeutung liegen im Modell näher beieinander, zum Beispiel „König“, „Königin“ und „Prinzessin“.

4. ## Wortvorhersage `Word Prediction`

   Das Modell lernt, welches Token wahrscheinlich als nächstes folgt. Wenn im Trainingsmaterial oft „Es war einmal …“ vorkommt, wird das Modell nach „Es war“ sehr wahrscheinlich „einmal“ erwarten.

5. ## Aufmerksamkeit `Attention`

   Damit das Modell nicht nur auf das letzte Wort schaut, prüft es, welche Wörter im bisherigen Satz besonders wichtig sind. In „Die Prinzessin brachte das Gold zur Bank“ ist „Gold“ wichtig, damit „Bank“ als Geldinstitut verstanden wird.

6. ## Neue Texte erzeugen `Text Generation`

   Aus den gelernten Mustern erzeugt das Modell neue Sätze. Es wählt Schritt für Schritt das wahrscheinlich nächste Token aus. So entstehen Texte, die manchmal sinnvoll, manchmal lustig und manchmal auch unsinnig sind.

Inhalt: 01 Trainingsdaten · 02 Tokenization · 03 Worteinbettung · 04 Wortbeziehungen · 05 Wortvorhersage · 06 Attention · 07 Textgenerierung · 08 Evaluation · 09 Grenzen von LLMs · Anhang: Trainingsdaten, Glossar

© 2026 E. Bechstädt · **CC BY-SA 4.0** · creativecommons.org/licenses/by-sa/4.0/deed.de

[Stufe 01]

Trainingsdaten sammeln

Texte → Modell

HINTERGRUND

Ein Large Language Model benötigt große Mengen an Texten, um Sprache zu lernen. Diese Texte nennt man **Trainingsdaten**.

Für unser Mini-Sprachmodell müssen wir glücklicherweise keine eigenen Trainingsdaten sammeln. Stattdessen nutzt ihr vorbereitete Märchensätze.

Jeder Trainingsdatensatz enthält andere Figuren, Orte und Satzmuster. Dadurch lernt jedes Sprachmodell etwas unterschiedlich und erzeugt später andere Ergebnisse.

PROMPT

Selbst bei der gleichen Eingabe, dem sogenannten **Prompt**, können verschiedene Texte entstehen.

1 **Wähle einen der vier Trainingsdatensätze aus dem Anhang aus.**

☐ Set A ☐ Set B ☐ Set C ☐ Set D

2 **Lies die enthaltenen Märchensätze aufmerksam durch.**

3 **Überlege:**

a) Welche Wörter und Figuren kommen besonders häufig vor?

b) Welche Stimmung entsteht durch die Texte?

c) Welche Arten von Sätzen könnte ein Sprachmodell daraus später erzeugen?

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

[Stufe 02]

Tokenization

Text → Bausteine

HINTERGRUND

Computer können Texte nicht direkt verstehen. Deshalb werden Sätze zunächst in kleine Einheiten zerlegt. Diese Einheiten nennt man **Tokens**. In modernen Large Language Models können Tokens ganze Wörter, Wortteile, Zahlen oder Satzzeichen sein.

Dadurch kann das Modell später lernen, welche Wörter häufig gemeinsam auftreten und welches Wort wahrscheinlich als nächstes folgt.

REGEL

Für unser Sprachmodell gilt:

1 Wort = 1 Token

1 Nutzt euren gewählten Trainingsdatensatz. **Zerlegt alle Märchensätze in einzelne Tokens.** Dabei gilt: Jedes Wort ist ein eigenes Token. Satzzeichen zählen ebenfalls als eigene Tokens. Schreibt jedes Token auf eine einzelne Karte oder schneidet die Wörter aus.

Beispiel: „Es war einmal ein König.“ → `Es` `war` `einmal` `ein` `König` `.`

= 6 Tokens

2 **Sortiert eure Tokens.** Findet heraus:

Welche Tokens kommen **besonders häufig** vor?	Welche Tokens kommen nur **einmal** vor?	Welche Wörter sind **typisch für Märchen**?

3 Warum könnte es für ein Sprachmodell hilfreich sein, Texte in kleine Einheiten zu zerlegen?

+ **Erweiterung (optional):** Moderne Sprachmodelle nutzen oft keine ganzen Wörter, sondern Wortteile. Beispiel:

`Königsschloss → Königs / schloss` `Drachenfeuer → Drachen / feuer`

`Zaubertrank → Zauber / trank`

Überlegt: Warum könnte diese Zerlegung für ein Sprachmodell nützlich sein?

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

[Stufe 03]

Worteinbettung

Word Embedding

HINTERGRUND

Ein Computer kann mit Wörtern allein noch nicht arbeiten. Deshalb werden Tokens in Zahlen umgewandelt. Diesen Schritt nennt man **Worteinbettung** oder **Word Embedding**.

Dabei erhält jedes Wort eine Art Zahlenbeschreibung. Wörter mit ähnlicher Bedeutung liegen im Sprachmodell später „nah beieinander".

Ein echtes Large Language Model berechnet dafür große Zahlenräume mit hunderten Eigenschaften. Wir vereinfachen diesen Prozess und ordnen Wörter selbst nach ihren Gemeinsamkeiten.

BEISPIEL

Thematisch gehören zusammen:

`König` `Königin` `Prinz` `Prinzessin`

1 Nutzt eure Tokens aus Stufe 2. **Sortiert die Wörter nun in sinnvolle Gruppen.** Wörter mit ähnlicher Bedeutung oder ähnlichem Einsatz sollen zusammengelegt werden.

Gruppe	mögliche Tokens
Figuren	König, Hexe, Ritter, Drache, ...
Orte	Schloss, Wald, Turm, ...

2 **Zusatzaufgabe:** Welche Wörter könnten zwischen zwei Gruppen passen?

? **Reflexionsfrage:** Warum könnte es für ein Sprachmodell hilfreich sein, ähnliche Wörter nah beieinander zu speichern?

+ **Erweiterung – Das Wörterfeld visualisieren:** Zeichnet eure Wortgruppen auf ein großes Blatt. Wörter mit ähnlicher Bedeutung sollen nah beieinander liegen. Verbindet oder umkreist Wörter, die gut zusammenpassen, oft gemeinsam auftreten oder ähnliche Rollen besitzen.

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

[Stufe 04]

Wortbeziehungen lernen

Wort → Wort

HINTERGRUND

Ein Sprachmodell lernt nicht nur einzelne Wörter, sondern auch deren Beziehungen. Es erkennt, welche Wörter häufig zusammen auftreten, welche Wörter oft aufeinander folgen und welche Satzmuster typisch sind.

Dadurch kann das Modell später vorhersagen, welches Wort wahrscheinlich als nächstes erscheint.

MERKE

Ein LLM speichert kein Wörterbuch, sondern ein großes **Netzwerk aus Wortbeziehungen**.

1 Nutzt eure Tokens und Wortgruppen aus den vorherigen Aufgaben. **Baut nun ein Sprachmodell auf. Zeichnet die Wortbeziehungen als Netzwerk.** Verbindet mit Pfeilen oder Linien jedes Wort mit dem Wort, das im Satz danach folgt. Wenn eine Verbindung mehrfach vorkommt, markiert sie dicker oder notiert die Anzahl daneben.

Ausschnitt eures Netzwerks (das vollständige Netzwerk auf einem großen Blatt)



2 **Zusatzaufgabe:** Sucht nach Wörtern, die mehrere mögliche Fortsetzungen besitzen. Beispiel:

Wort	mögliche Fortsetzungen
Der	König, Ritter, Drache
ging	in, durch, zum

? **Reflexionsfrage:** Warum reichen einzelne Wörter nicht aus, um Sprache sinnvoll vorherzusagen?

© 2026 E. Bechstädt · **CC BY-SA 4.0** · creativecommons.org/licenses/by-sa/4.0/deed.de

[Stufe 05]

Wortvorhersage

Word Prediction

HINTERGRUND

Das Sprachmodell kennt nun Wörter, Wortgruppen und Beziehungen zwischen ihnen. Jetzt kann das Modell beginnen, Sprache vorherzusagen. Diesen Prozess nennt man **Wortvorhersage** oder **Word Prediction**.

Ein Large Language Model erzeugt Texte genau auf diese Weise: Es berechnet immer wieder, welche Tokens als Nächstes wahrscheinlich sind, und wählt daraus ein Token aus.

DAS MODELL BERECHNET

Welches Wort folgt wahrscheinlich als nächstes?

1 Nutzt euer aufgebautes Sprachmodell. **Versucht nun, neue Märchensätze zu erzeugen.** Geht dabei Schritt für Schritt vor:

1. Wählt ein Startwort.
2. Folgt einer Verbindung zum nächsten Wort.
3. Wiederholt den Vorgang.
4. Stoppt, wenn kein Wort mehr folgt.

2 Einige Wörter besitzen mehrere mögliche Fortsetzungen. **Berechnet**, welche Fortsetzung am wahrscheinlichsten ist und welche nur selten vorkommt.

Wort	Fortsetzung	Häufigkeit	Wahrscheinlichkeit
Der	König		
Der	Ritter		
Der	Drache		

Tipp: Wahrscheinlichkeit = Häufigkeit dieser Fortsetzung ÷ Anzahl aller Fortsetzungen des Wortes

? **Reflexionsfrage:** Warum entstehen manchmal lustige oder unsinnige Sätze?

Bedeutet eine hohe Wahrscheinlichkeit automatisch, dass ein Satz sinnvoll ist?

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

[Stufe 06]

Aufmerksamkeit

Attention

1 Daten — 2 Tokens — 3 Einbettung — 4 Beziehungen — 5 Vorhersage — 6 Attention — 7 Generierung — 8 Evaluation

HINTERGRUND

Bisher betrachtet unser Sprachmodell hauptsächlich das vorherige Wort oder kurze Wortfolgen. Dadurch entstehen jedoch Probleme: Manche Wörter besitzen mehrere Bedeutungen. Das Sprachmodell muss deshalb erkennen, welche Wörter im Satz besonders wichtig sind. Diesen Mechanismus nennt man **Aufmerksamkeit** oder **Attention**.

DAS MODELL PRÜFT

Auf welche Wörter muss besonders geachtet werden, um den Satz richtig zu verstehen?

BEISPIEL

Satz 1 Der König saß auf der **Bank** im Garten.

Satz 2 Die Königin brachte das Gold zur **Bank**.

Das Wort „Bank“ besitzt hier zwei Bedeutungen: **Sitzgelegenheit** und **Geldinstitut**. Das Sprachmodell muss deshalb wichtige Kontextwörter erkennen.

1 Untersucht die Beispielsätze. **Markiert die Wörter, die helfen, die Bedeutung von „Bank“ zu erkennen.** Nutzt zwei Farben:

`Grün = Sitzgelegenheit` `Blau = Geldinstitut`

2 Untersucht: **Welche Wörter sind für das Verständnis eher unwichtig?**

Beispiel: `alte` `gleiche` `schöne`

+ **Erweiterung – Attention sichtbar machen:** Markiert im Satz mit Hilfe von Pfeilen, welche Wörter zum Erschließen der Bedeutung eines Wortes helfen.

Die Königin brachte das Gold zur Bank.

? **Reflexionsfragen:** Warum reicht das vorherige Wort allein oft nicht aus, um Sprache richtig zu verstehen? Warum ist Attention für Chatbots besonders wichtig?

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

[Stufe 07]

Textgenerierung

Prompt → Text

1 Daten · 2 Tokens · 3 Einbettung · 4 Beziehungen · 5 Vorhersage · 6 Attention · **7 Generierung** · 8 Evaluation

HINTERGRUND

Ein Large Language Model erzeugt Texte nicht auf einmal. Stattdessen arbeitet es Schritt für Schritt. So entstehen Geschichten, Antworten, Dialoge und sogar ganze Bücher.

Dies ist der Prozess einer **Generativen KI**, welche eine Textgenerierung, aber auch Bild-, Audio- oder Videogenerierung ermöglicht.

1. Das Modell erhält einen Satzanfang (Prompt).
2. Es sagt das wahrscheinlich nächste Wort voraus.
3. Danach wird erneut das nächste Wort vorhergesagt.
4. Dieser Vorgang wiederholt sich immer weiter.

1 **Erzeugt mit Hilfe eures Sprachmodells eigene (lustige) Märchen.** Die Entscheidungen, unter Beachtung der Wahrscheinlichkeiten, könnt ihr mit Hilfe eines Zufallsgenerators bestimmen lassen. Dabei dürft ihr:

- zufällige Entscheidungen treffen,
- verschiedene Wortpfade nutzen,
- oder Wahrscheinlichkeiten berücksichtigen.

2 **Generiert mindestens drei neue Märchenanfänge.** Beginnt jeweils mit:

„Es war einmal …“

„Ein König …“

„Vor Zeiten …“

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

[Stufe 08]

Evaluation

Text → Bewertung

1 Daten · 2 Tokens · 3 Einbettung · 4 Beziehungen · 5 Vorhersage · 6 Attention · 7 Generierung · 8 Evaluation

1 **Bewertet eure in Stufe 7 erzeugten Sätze.** Schreibt drei eurer Sätze auf:

S1 .....

S2 .....

S3 .....

	Bewertung von 1 (sehr schlecht) bis 4 (sehr gut)		
Frage	**S1**	**S2**	**S3**
sinnvoll?			
typisch Märchen?			
eher unsinnig?			
grammatikalisch richtig?			
lustig?			

? **Reflexionsfrage:** Warum kann ein Sprachmodell kreative, aber manchmal auch unsinnige Texte erzeugen?

.....

.....

.....

.....

? **Reflexionsfrage:** Worin unterscheiden sich menschliches Schreiben und KI-generierte Texte?

.....

.....

.....

.....

© 2026 E. Bechstädt · **CC BY-SA 4.0** · creativecommons.org/licenses/by-sa/4.0/deed.de

[Abschlussphase]

Grenzen und Probleme von LLMs

1 Daten — 2 Tokens — 3 Einbettung — 4 Beziehungen — 5 Vorhersage — 6 Attention — 7 Generierung — 8 Evaluation

1 **Kurze Reflexion:** Erklärt die Begriffe und findet jeweils ein Beispiel aus eurem eigenen Märchen-Sprachmodell.

Begriff	Was bedeutet das?	Beispiel aus unserem Modell
Bias		
Halluzinationen		
Trainingsdaten		
stereotype Antworten		
Unsinn trotz hoher Wahrscheinlichkeit		

? Was bedeuten diese Grenzen für euren Umgang mit Chatbots wie ChatGPT, Claude oder Gemini?

© 2026 E. Bechstädt · **CC BY-SA 4.0** · creativecommons.org/licenses/by-sa/4.0/deed.de ✓

[Anhang]

Trainingsdaten

Lustige Beispielsätze · Set A bis D

Set A

8 Sätze

1. Es war einmal ein Königssohn, der lieber Einrad fuhr als reiten lernte.
2. Es war einmal eine Königin, die jeden Morgen ihre Drachen mit Haferbrei fütterte.
3. Es war einmal ein Spielmann, der im Wald laut Karaoke sang und alle Tiere vertrieb.
4. Ein armer Müller hatte drei Söhne und einen Hahn, der ständig Opern sang.
5. Vor Zeiten war ein Schneider, der nur Hosen für Riesen nähte.
6. Es war ein kleiner Drache, der Angst vor gerösteten Marshmallows hatte.
7. Es war einmal ein König, der heimlich nachts Pfannkuchen buk.
8. Es war ein armer Ritter, dessen Pferd ständig einschlief.

Set B

8 Sätze

1. Es war einmal eine Prinzessin, die lieber in Gummistiefeln als in goldenen Schuhen tanzte.
2. Ein König hatte zwölf Kinder, aber keines räumte sein Zimmer auf.
3. Es war einmal ein Zauberer, der dauernd seine eigenen Zaubersprüche vergaß.
4. Ein alter Fischer fing einen sprechenden Karpfen, der nur über Wetter berichtete.
5. Es war ein Drache, der jeden Abend Angst vor dem Dunkeln bekam.
6. Vor Zeiten lebte ein Müller, dessen Esel unglaublich schlecht sang.
7. Es war einmal eine Hexe, die nur mit Erdbeerkuchen zaubern konnte.
8. Ein Königssohn zog durch die Welt und suchte die größte Portion Pommes.

Set C

8 Sätze

1. Es war einmal ein Ritter, der seinen Helm ständig falsch herum trug.
2. Ein armer Bauer besaß eine Ziege, die alle Schuhe im Dorf fraß.
3. Es war einmal ein König, der nur rückwärts auf seinem Thron sitzen wollte.
4. Eine Hexe wohnte tief im Wald und verkaufte dort verhexte Kekse.
5. Es war ein Prinz, der lieber Schnecken sammelte als Drachen bekämpfte.
6. Vor Zeiten lebte ein Drache, der nur vegetarische Suppe kochte.
7. Ein Müller wanderte durch den Wald und sprach die ganze Zeit mit Eichhörnchen.
8. Es war einmal eine Königin, die heimlich beim Ritterturnier mitmachte.

Set D

besonders verrückt

1. Es war einmal ein Drache, der mit einer Zahnbürste kämpfte.
2. Ein König hatte einen goldenen Kühlschrank im Thronsaal stehen.
3. Es war einmal eine Prinzessin, die auf einem fliegenden Teppich Pizza auslieferte.
4. Ein Ritter verlor im Wald ständig seine Hose.
5. Es war eine Hexe, die ihre Zaubertränke mit Senf würzte.
6. Vor Zeiten lebte ein Kater, der jeden Morgen den König weckte.
7. Es war einmal ein Bauer, der seinem Schwein das Tanzen beibrachte.
8. Ein Drache versteckte heimlich Marshmallows unter seinem Schatz.

Die Sätze von Set D erzeugen später besonders absurde Kombinationen. Tipp: Seite kopieren und die Sets für die Gruppen auseinanderschneiden.

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

[Anhang]

Fachvokabular

Large Language Models verständlich erklärt · 1/2

Dieses Glossar enthält die wichtigsten Fachbegriffe des Unterrichtsmaterials in vereinfachter Form. Die deutschen Bezeichnungen stehen dabei bewusst im Vordergrund, damit die Begriffe auch für jüngere Lernende zugänglich bleiben.

Fachbegriff	Deutsche Bezeichnung	Kurzbeschreibung
Large Language Model (LLM)	Großes Sprachmodell	Ein KI-System, das mit sehr vielen Texten trainiert wurde und neue Sprache erzeugen kann.
Trainingsdaten	Trainingsdaten	Texte, mit denen ein Sprachmodell lernt.
Token	Spracheinheit	Kleine Bestandteile eines Textes, z. B. Wörter, Wortteile oder Satzzeichen.
Tokenisierung	Zerlegung in Tokens	Der Prozess, bei dem Texte in einzelne Tokens aufgeteilt werden.
Word Embedding	Worteinbettung	Wörter werden als Zahlen dargestellt. Ähnliche Wörter liegen mathematisch nah beieinander.
Wortvektor	Zahlenbeschreibung eines Wortes	Die mathematische Darstellung eines Wortes innerhalb eines Sprachmodells.
Wortbeziehungen	Beziehungen zwischen Tokens	Verbindungen zwischen Wörtern, die häufig gemeinsam auftreten.
Next Token Prediction	Wortvorhersage	Das Modell berechnet, welcher Wortteil/welches Wort wahrscheinlich als nächstes folgt.
Wahrscheinlichkeit	Wahrscheinlichkeit	Gibt an, wie wahrscheinlich eine bestimmte Fortsetzung ist.
Prompt	Eingabeaufforderung	Der Anfang eines Textes oder eine Frage, die an das Sprachmodell gegeben wird.
Attention	Aufmerksamkeit	Mechanismus, mit dem wichtige Wörter im Kontext erkannt werden.
Kontext	Zusammenhang	Die Umgebung eines Wortes innerhalb eines Satzes oder Textes.

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

Fachvokabular (Fortsetzung)

Fachbegriff	Deutsche Bezeichnung	Kurzbeschreibung
Transformer	Transformer-Modell	Moderne Architektur von Sprachmodellen, die mit Attention arbeitet.
Textgenerierung	Erzeugen neuer Texte	Das schrittweise Schreiben neuer Texte durch Wortvorhersagen.
Generative KI	Generative künstliche Intelligenz	KI-Systeme, die neue Inhalte erzeugen können, z. B. Texte oder Bilder.
Bias	Verzerrung / Voreingenommenheit	Fehler oder Vorurteile, die aus den Trainingsdaten übernommen werden.
Halluzination	Halluzination	Eine KI erzeugt falsche oder erfundene Inhalte, die glaubwürdig wirken.
Sprachmodell	Sprachmodell	Ein Modell, das Wahrscheinlichkeiten zwischen Wörtern lernt.
Datensatz	Datensammlung	Sammlung von Texten oder Informationen zum Trainieren einer KI.
Modelltraining	Training des Modells	Der Vorgang, bei dem ein Modell aus den Trainingsdaten Muster und Wahrscheinlichkeiten lernt.
Parameter	Stellwerte des Modells	Innere Werte eines Sprachmodells, die beim Lernen angepasst werden.
KI-Modell	Modell künstlicher Intelligenz	Mathematisches System zur Verarbeitung von Informationen.

Vereinfachte Merksätze für den Unterricht

Begriff	Merksatz
Token	„Ein Computer zerlegt Sprache in kleine Bausteine.“
Word Embedding	„Ähnliche Wörter liegen im Modell nah beieinander.“
Word Prediction	„Das Modell sagt das wahrscheinlich nächste Wort voraus.“
Attention	„Das Modell achtet auf wichtige Wörter.“
Generative KI	„Die KI erzeugt neue Texte aus Wahrscheinlichkeiten.“
Bias	„Die KI übernimmt Muster aus ihren Trainingsdaten.“
Halluzination	„Die KI kann überzeugend klingende Fehler erzeugen.“

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

[Einführung]

Hinweise zur Unterrichtsreihe

EINLEITUNG

Künstliche Intelligenz begegnet Schülerinnen und Schülern mittlerweile täglich, in Chatbots, Suchmaschinen, Übersetzungsprogrammen oder sozialen Netzwerken. Doch wie funktioniert ein Large Language Model eigentlich? Wie kann ein Computer scheinbar „verstehen", was wir schreiben, und selbst neue Texte erzeugen?

Dieses Unterrichtsmaterial verfolgt einen bewusst **handlungsorientierten Ansatz**: Die Lernenden bauen Schritt für Schritt ihr eigenes vereinfachtes Sprachmodell auf und durchlaufen dabei zentrale Entwicklungsphasen moderner Large Language Models selbstständig und analog. Statt fertige Erklärungen zu konsumieren, arbeiten sie mit Tokens, erzeugen Wortnetzwerke, untersuchen Wahrscheinlichkeiten, analysieren Aufmerksamkeit („Attention") und generieren schließlich eigene KI-Märchen.

Geeignet für informatische Bildung, Medienbildung, KI-Einführungen, fächerverbindenden Unterricht und projektorientiertes Lernen. **Einsatz** vollständig als Unterrichtsreihe, modular als Einzelstunden oder auszugsweise.

ZEITLICHE EINSCHÄTZUNG DER UNTERRICHTSPHASEN

Stufe	Inhalt	geschätzte Dauer
1	Trainingsdaten auswählen	5 min
2	Tokenisierung	10–15 min
3	Worteinbettung / Word Embedding	10–15 min
4	Wortbeziehungen lernen	15–20 min
5	Wortvorhersage / Word Prediction	10–15 min
6	Aufmerksamkeit / Attention	10–15 min
7	Textgenerierung	10–15 min
8 + AB 09	Evaluation, Grenzen und Probleme von LLMs (gemeinsame Auswertung)	ca. 30 min
	Gesamt	**ca. 120 min**

Praxiserfahrung: Im Grundkurs (Dreiergruppen) dauerte die Gruppenarbeit an Stufe 1–7 insgesamt 90 Minuten, danach folgten etwa 30 Minuten gemeinsame Zusammenfassung und Auswertung. Die Zeiten je Stufe sind Richtwerte.

DIDAKTISCHE HINWEISE

Das Material folgt dem Prinzip **vom Konkreten zum Abstrakten** sowie dem **EIS-Prinzip**:

- **enaktiv**: Karten legen, Pfeile verbinden
- **ikonisch**: Netzwerke darstellen
- **symbolisch**: Wahrscheinlichkeiten und Modelle

Die Lernenden erfahren so, dass KI nicht „magisch" arbeitet, sondern auf Wahrscheinlichkeiten basiert und Sprache als Muster verarbeitet. Die absurden Märchentexte fördern Motivation, Gesprächsanlässe, Fehlersuche und kritische Reflexion.

ZU DEN TRAININGSDATEN

Die Sätze enthalten typische Märchenanfänge, häufige Tokens („Es", „war", „einmal", „ein"), ähnliche Satzstrukturen, wiederkehrende Rollen und klare Wortbeziehungen. So entstehen beim Generieren sinnvolle, halbsinnvolle und völlig absurde KI-Märchen. Das eignet sich gut für Diskussionen über Wahrscheinlichkeiten, Kreativität und Grenzen von KI sowie „stochastische Papageien".

FACHBEGRIFFE

Englische Fachbegriffe einführen, im Unterrichtsgespräch aber bevorzugt die deutschen verwenden. Die englischen Begriffe gelten dann als „Fachsprache der KI-Forschung".

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

LÖSUNG · für Lehrkräfte

[Lösung Stufe 01]

Trainingsdaten sammeln

Erwartungshorizont

a *Welche Wörter und Figuren kommen besonders häufig vor?*

Set	Figuren	Auffällige Themen
A	König, Königssohn, Königin, Müller, Ritter, Drache, Spielmann, Schneider	Essen (Haferbrei, Pfannkuchen, Marshmallows), Musik (Karaoke, Opern)
B	Prinzessin, König, Königssohn, Zauberer, Hexe, Drache, Müller, Fischer	Essen (Erdbeerkuchen, Pommes), Kleidung (Gummistiefel, Schuhe)
C	Ritter, König, Königin, Prinz, Hexe, Drache, Müller, Bauer	Tiere (Ziege, Schnecken, Eichhörnchen), Wald, Essen
D	Drache (2×), König (2×), Prinzessin, Ritter, Hexe, Kater, Bauer	Alltagsdinge (Zahnbürste, Kühlschrank, Pizza, Senf)

In **allen** Sets am häufigsten: **Es, war, einmal, ein/eine, der/die** sowie Komma und Punkt.

b *Welche Stimmung entsteht durch die Texte?*

- **Märchenhaft** durch typische Anfänge („Es war einmal“, „Vor Zeiten“) und bekannte Figuren (König, Hexe, Drache).
- **Humorvoll bis absurd**, weil die Figuren etwas Unerwartetes tun: ein Drache mit Zahnbürste, ein Hahn, der Opern singt.
- Der Witz entsteht durch den **Bruch mit der Erwartung**. Das hilft später beim Erkennen von Unsinn.

c *Welche Arten von Sätzen könnte ein Sprachmodell daraus später erzeugen?*

Die Sätze folgen wenigen Mustern. Genau diese Muster wird das Modell lernen und neu kombinieren:

```
Es war einmal ein/eine [Figur], der/die [Handlung].
```

```
Ein [Figur] hatte / besaß [Ding] …
```

```
Vor Zeiten lebte / war ein [Figur], der [Handlung].
```

Erwartbar sind also **Märchenanfänge mit Nebensatz**, bei denen Figuren und Handlungen aus verschiedenen Sätzen vermischt werden, z. B. „Es war einmal ein König, der ständig Opern sang.“ (Set A)

Hinweis: Jedes Set erzeugt ein anderes Modell: gleicher Prompt, andere Texte. Das lässt sich in Stufe 7 gut vergleichen.

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

[Lösung Stufe 02]

Tokenization

Erwartungshorizont

1 *Zerlegt alle Märchensätze in einzelne Tokens (Wörter und Satzzeichen).*

Set	Sätze	Tokens gesamt	davon Satzzeichen	verschiedene Tokens
A	8	110	16	67
B	8	108	15	75
C	8	106	14	72
D	8	92	13	60

Gezählt mit Groß-/Kleinschreibung („Ein“ ≠ „ein“, wie bei echten LLMs). Kleine Abweichungen sind unproblematisch.

2 *Sortiert eure Tokens: häufig, nur einmal, typisch für Märchen.*

Set	Häufigste Tokens (Anzahl)	nur 1× vorkommend
A	, 8 · . 8 · war 7 · Es 6 · ein 6 · der 6 · einmal 4	55 von 67
B	. 8 · , 7 · Es 4 · war 4 · die 4 · einmal 3 · Ein 3 · ein 3	63 von 75
C	. 8 · , 6 · Es 4 · war 4 · ein 4 · der 4 · einmal 3	58 von 72
D	. 8 · , 5 · Es 4 · war 4 · einmal 3 · ein 3 · Ein 3 · der 3	45 von 60

- **Häufig** sind kleine Funktionswörter und Satzzeichen, also das Satzgerüst.
- **Nur einmal** kommen die meisten Inhaltswörter vor (Zahnbürste, Karaoke, Pommes …).
- **Typisch Märchen:** Es war einmal, Vor Zeiten, König, Königin, Prinzessin, Ritter, Hexe, Drache, Müller, Zauberer, Wald, Thron.

3 *Warum ist es hilfreich, Texte in kleine Einheiten zu zerlegen?*

- Kleine Einheiten lassen sich **zählen, vergleichen und in Zahlen umwandeln**. Das ist die Grundlage für alle weiteren Stufen.
- Aus wenigen Bausteinen können **beliebig viele neue Sätze** zusammengesetzt werden.
- Das Modell kann lernen, **welches Token wie oft auf ein anderes folgt**.

+ *Erweiterung: Warum ist die Zerlegung in Wortteile nützlich?*

`Zaubertrank → Zauber | trank` `Königsschloss → Königs | schloss`

`Drachenfeuer → Drachen | feuer`

Neue oder seltene Wörter lassen sich aus **bekannten Teilen** verstehen, der Wortschatz bleibt kleiner. Wichtig bei deutschen Komposita und Wortformen (König / Königs).

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

[Lösung Stufe 03]

Worteinbettung

Erwartungshorizont

1 *Sortiert die Wörter in sinnvolle Gruppen (Beispiel aus allen vier Sets).*

Gruppe	mögliche Tokens
Figuren	König, Königin, Prinzessin, Prinz, Königssohn, Ritter, Hexe, Zauberer, Müller, Bauer, Fischer, Schneider, Spielmann
Tiere	Drache, Hahn, Esel, Ziege, Kater, Schwein, Karpfen, Pferd, Schnecken, Eichhörnchen
Orte	Wald, Thronsaal, Thron, Dorf, Zimmer, Welt
Essen	Haferbrei, Pfannkuchen, Marshmallows, Pizza, Erdbeerkuchen, Pommes, Kekse, Suppe, Senf
Dinge / Kleidung	Zahnbürste, Kühlschrank, Teppich, Helm, Hose, Gummistiefel, Schuhe, Schatz
Tätigkeiten	sang, tanzte, kämpfte, zaubern, kochte, fraß, buk, fütterte, nähte
Eigenschaften	armer, alter, kleiner, goldenen, fliegenden, verhexte, vegetarische
Funktionswörter	Es, war, einmal, ein, eine, der, die, und, mit, im · Satzzeichen , .

Andere begründete Gruppen (z. B. „Familie“, „Zauberei“) sind ebenso richtig.

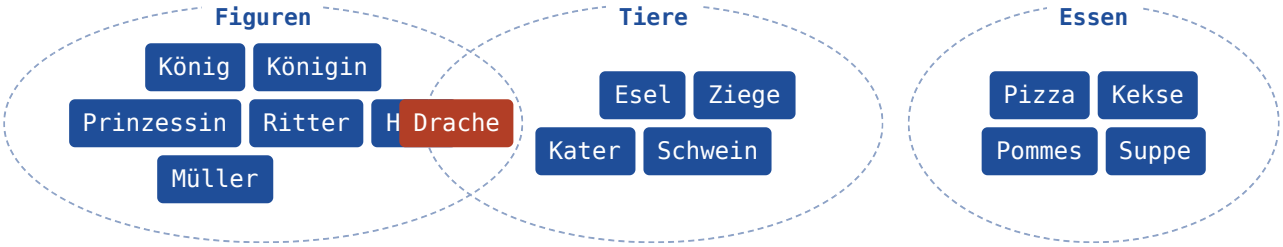
2 *Zusatzaufgabe: Welche Wörter könnten zwischen zwei Gruppen passen?*

- **Drache, Kater**: Tier und Figur, denn beide handeln wie Personen.
- **Ritterturnier**: Figur und Ereignis · **Zaubertränke**: Ding und Zauberei · **Schatz**: Ding und Eigenschaft.

? *Reflexion: Warum ist es hilfreich, ähnliche Wörter nah beieinander zu speichern?*

- Ähnliche Wörter sind **austauschbar**: Was für „König“ gilt, passt oft auch für „Königin“.
- Das Modell kann Gelerntes auf **seltene Wörter übertragen**, statt jedes Wort einzeln zu lernen.
- Bedeutung wird **messbar** als Abstand zwischen Zahlen. Das hilft bei Synonymen und bei der Vorhersage.

+ *Erweiterung: Wörterfeld (Beispiel). „Drache“ liegt in der Überschneidung zweier Felder.*



© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

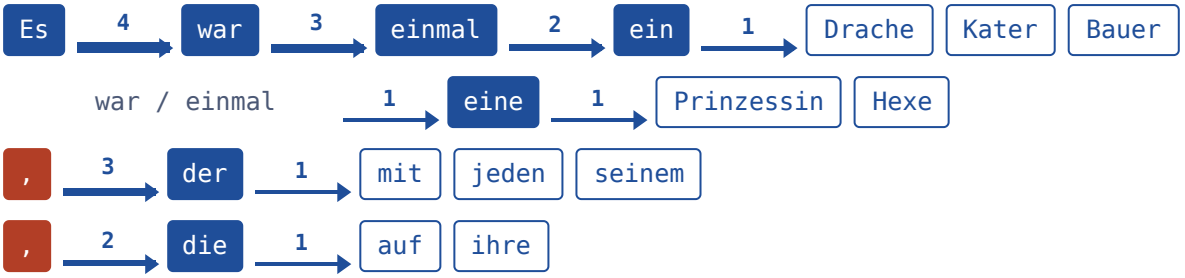
LÖSUNG · für Lehrkräfte

[Lösung Stufe 04]

Wortbeziehungen lernen

Erwartungshorizont

1 *Zeichnet die Wortbeziehungen als Netzwerk, mehrfache Verbindungen dicker. (Ausschnitt Set D)*



Stärkste Verbindungen je Set (diese Pfeile sollten am dicksten sein):

Set	Verbindung (Anzahl)
A	Es→war 6 · ,→der 6 · war→einmal 4 · war→ein 3 · einmal→ein 3
B	Es→war 4 · war→einmal 3 · ,→der 3 · einmal→eine 2 · ,→die 2
C	Es→war 4 · ,→der 4 · war→einmal 3 · einmal→ein 2 · der→nur 2 · Wald→und 2
D	Es→war 4 · war→einmal 3 · ,→der 3 · einmal→ein 2 · ,→die 2

2 *Zusatzaufgabe: Sucht Wörter mit mehreren möglichen Fortsetzungen.*

Set	Wort → mögliche Fortsetzungen
A	ein → Königssohn | Spielmann | Schneider | kleiner | König | armer , → der | die | dessen · armer → Müller | Ritter
B	Ein → König | alter | Königssohn · die → lieber | nur | Welt | größte , → der | die | aber | dessen
C	ein → Ritter | König | Prinz | Drache · der → nur | seinen | lieber im → Dorf | Wald · und → verkaufte | sprach
D	Ein → König | Ritter | Drache · Drache → , | versteckte mit → einer | Senf · im → Thronsaal | Wald · seinem → Schwein | Schatz

? *Reflexion: Warum reichen einzelne Wörter nicht aus, um Sprache sinnvoll vorherzusagen?*

- Ein Wort allein hat **viele mögliche Nachfolger**. Erst Wortfolgen und Satzmuster zeigen, was passt.
- Grammatik hängt von **früheren Wörtern** ab: nach „eine Hexe ,“ muss „die“ folgen, nicht „der“.
- Bedeutung entsteht erst im **Zusammenhang**. Dieses Problem löst später die Attention (Stufe 6).

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

[Lösung Stufe 05]

Wortvorhersage

Erwartungshorizont

1 *Erzeugt neue Märchensätze: Startwort wählen, Verbindung folgen, wiederholen, beim Punkt stoppen.*

Beispielpfad im Netzwerk von Set D

`Es` → `war` → `einmal` → `ein` → `Drache` → `,` → `der` → `mit` → `Senf` → `würzte` → `.`

Ergebnis: „Es war einmal ein Drache, der mit Senf würzte.“ Jeder Übergang steht so im Trainingsdatensatz, der ganze Satz aber nirgends.

Korrekt ist jeder Satz, dessen Übergänge alle im eigenen Netzwerk vorkommen.

2 *Berechnet, welche Fortsetzung am wahrscheinlichsten ist und welche nur selten vorkommt.*

FORMEL P(Fortsetzung) = **Anzahl dieser Fortsetzung ÷ Anzahl aller Fortsetzungen** des Wortes

Set	Wort	Fortsetzungen (Anzahl)	Wahrscheinlichkeit (fett = am häufigsten)
A	war (7×)	einmal 4 · ein 3	**57 %** · 43 %
A	, (8×)	der 6 · die 1 · dessen 1	**75 %** · 12,5 % · 12,5 %
B	war (4×)	einmal 3 · ein 1	**75 %** · 25 %
B	, (7×)	der 3 · die 2 · aber 1 · dessen 1	**43 %** · 29 % · 14 % · 14 %
C	, (6×)	der 4 · die 2	**67 %** · 33 %
C	der (4×)	nur 2 · seinen 1 · lieber 1	**50 %** · 25 % · 25 %
D	war (4×)	einmal 3 · eine 1	**75 %** · 25 %
D	, (5×)	der 3 · die 2	**60 %** · 40 %
D	Ein (3×)	König 1 · Ritter 1 · Drache 1	je 33 %

? *Reflexion: Warum entstehen lustige oder unsinnige Sätze? Bedeutet hohe Wahrscheinlichkeit „sinnvoll“?*

- Das Modell kennt nur, **welches Wort auf welches folgt**, nicht deren Bedeutung.
- **Nein.** Hohe Wahrscheinlichkeit heißt nur: **kam in den Trainingsdaten oft vor**. Sie sagt nichts über Wahrheit, Logik oder Grammatik.
- Beispiel (Set D): „Es war einmal eine Hexe, der mit Senf würzte.“ Jeder Schritt ist erlaubt, der Satz ist trotzdem falsch.

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

[Lösung Stufe 06]

Aufmerksamkeit (Attention)

Erwartungshorizont

1 *Markiert die Wörter, die helfen, die Bedeutung von „Bank“ zu erkennen (Grün = Sitzgelegenheit, Blau = Geldinstitut).*

Satz 1 · Sitzgelegenheit

Der | König | saß | auf | der | **Bank** | im | Garten | .

Satz 2 · Geldinstitut

Die | Königin | brachte | das | Gold | zur | **Bank** | .

Bedeutung	Hinweiswörter und Begründung
Grün · Sitz	**saß, auf, Garten**: Auf einer Bank im Garten sitzt man.
Blau · Geld	**Gold, brachte**: Gold bringt man zur Bank, um es aufzubewahren. Das wichtigste Wort ist „Gold“.

2 *Welche Wörter sind für das Verständnis eher unwichtig?*

Der | die | das | der | im | König | Königin | alte | gleiche | schöne

Artikel und **schmückende Adjektive** verändern die Bedeutung von „Bank“ nicht. Auch **wer** handelt (König oder Königin), ist hier unwichtig. Achtung: „zur“ steht direkt vor „Bank“, passt aber zu beiden Bedeutungen und hilft deshalb nicht.

+ *Erweiterung: Attention mit Pfeilen sichtbar machen*

saß → Bank

Garten → Bank

Satz 1: Bezug auch über mehrere Wörter hinweg

Gold ⟹ Bank (stark)

brachte → Bank

Satz 2: dicker Pfeil = mehr Aufmerksamkeit

Weitere Beispiele aus den Märchensätzen: In „Ein Drache versteckte Marshmallows unter **seinem** Schatz“ zeigt „seinem“ auf **Drache**. „**Schloss**“ kann ein Gebäude oder ein Türschloss sein.

? *Reflexion: Warum reicht das vorherige Wort oft nicht? Warum ist Attention für Chatbots wichtig?*

- Das vorherige Wort („zur“) passt oft zu **mehreren Bedeutungen**. Die entscheidenden Hinweise stehen weiter vorn im Satz.
- Wörter wie „er“, „sie“, „seinem“ beziehen sich auf **weit entfernte Wörter**.
- Chatbots müssen **lange Fragen und ganze Gespräche** im Blick behalten und die richtige Bedeutung wählen. Ohne Attention gäbe es Missverständnisse und unpassende Antworten.

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

[Lösung Stufe 07]

Textgenerierung

Erwartungshorizont

1 *Erzeugt eigene Märchen. Die Entscheidungen könnt ihr unter Beachtung der Wahrscheinlichkeiten per Zufallsgenerator treffen.*

So wird der Zufall fair: Jede Fortsetzung bekommt so viele Würfelaugen (oder Lose), wie sie im Netzwerk vorkommt. Beispiel Set D:

nach **war**: einmal 3× · eine 1×

1 2 3 → einmal

4 → eine 5 6 → neu würfeln

ergibt genau 75 % · 25 %

nach **,**: der 3× · die 2×

1 2 3 → der

4 5 → die 6 → neu würfeln

ergibt genau 60 % · 40 %

Alternative Lostopf: Pro Vorkommen ein Zettel, ziehen und zurücklegen. So funktioniert auch das „Sampling" echter Sprachmodelle.

2 *Generiert mindestens drei neue Märchenanfänge mit den vorgegebenen Prompts.*

Prompt	Set	möglicher generierter Satz (aus den Netzwerken geprüft)
Es war einmal …	**A**	Es war einmal ein König, der ständig Opern sang.
	D	Es war einmal ein Bauer, der mit Senf würzte.
Ein König …	**B**	Ein König hatte zwölf Kinder, dessen Esel unglaublich schlecht sang.
	D	Ein König hatte einen goldenen Kühlschrank im Wald ständig seine Hose.
Vor Zeiten …	**A**	Vor Zeiten war ein Spielmann, der ständig Opern sang.
	C	Vor Zeiten lebte ein König, der nur vegetarische Suppe kochte.

Bewertungskriterium: Jeder Übergang muss im eigenen Netzwerk vorkommen und der Satz endet mit einem Punkt. Inhaltlich sind alle Ergebnisse individuell.

Gesprächsanlass: In Set A kommt „Ein König" nie vor. Das Modell kann nur ab „König" weitermachen, z. B. „Ein König, der ständig einschlief." Ein Satz ohne Hauptverb entsteht. Echte LLMs haben ein ähnliches Problem, wenn ein Prompt kaum zu ihren Trainingsdaten passt.

? *Wo steckt die „generative KI"?*

- Der Text entsteht **Token für Token**: Jede Vorhersage wird an den Text angehängt und ist dann Grundlage für die nächste.
- Durch den **Zufall** entstehen beim gleichen Prompt jedes Mal andere Texte.

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

[Lösung Stufe 08]

Evaluation

Erwartungshorizont

1 *Bewertet eure in Stufe 7 erzeugten Sätze (1 = sehr schlecht bis 4 = sehr gut / trifft voll zu).*

S1 „Es war einmal ein König, der ständig Opern sang.“ (Set A)

S2 „Es war einmal eine Hexe, der mit Senf würzte.“ (Set D)

S3 „Ein König hatte einen goldenen Kühlschrank im Wald ständig seine Hose.“ (Set D)

Frage	S1	S2	S3	Begründung (Beispiel)
sinnvoll?	**4**	**1**	**1**	S1 ist eine vollständige, verständliche Aussage.
typisch Märchen?	**4**	**3**	**2**	Märchenanfang und Figuren vorhanden, S3 mit Kühlschrank kaum.
eher unsinnig?	**1**	**3**	**4**	S3 verbindet zwei Sätze ohne Zusammenhang.
grammatikalisch richtig?	**4**	**1**	**1**	S2: „eine Hexe, **der**“ ist falsch. S3: zwei Satzteile ohne Verbindung.
lustig?	**3**	**3**	**4**	Absurde Kombinationen wirken komisch.

Die Bewertung ist teilweise subjektiv. Entscheidend ist eine nachvollziehbare Begründung. Vergleicht die Bewertungen zwischen Gruppen.

? *Warum kann ein Sprachmodell kreativ, aber manchmal auch unsinnige Texte erzeugen?*

- **Kreativ:** Es kombiniert gelernte Bausteine neu. So entstehen Sätze, die es in den Trainingsdaten nie gab.
- **Unsinnig:** Es wählt nach **Wahrscheinlichkeit**, nicht nach Bedeutung. Logik, Grammatik über mehrere Wörter und Wissen über die Welt prüft es nicht.
- Deshalb spricht man von einem **„stochastischen Papagei“**: Er plappert statistisch nach, ohne zu verstehen.

? *Worin unterscheiden sich menschliches Schreiben und KI-generierte Texte?*

Mensch	Sprachmodell
schreibt mit **Absicht** und Ziel	setzt fort, was **wahrscheinlich** ist
nutzt **Erfahrung, Wissen und Gefühle**	nutzt **Muster aus Trainingsdaten**
versteht die **Bedeutung** der Wörter	rechnet mit Tokens als Zahlen
kann den eigenen Text prüfen und Fehler bemerken	merkt Unsinn nicht selbst, kann überzeugend falsch klingen
langsamer, übernimmt **Verantwortung**	sehr schnell, große Textmengen, ohne Verantwortung

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de

[Lösung Abschlussphase]

Grenzen und Probleme von LLMs

Erwartungshorizont

1 Daten · 2 Tokens · 3 Einbettung · 4 Beziehungen · 5 Vorhersage · 6 Attention · 7 Generierung · 8 Evaluation

! *Kurze Reflexion: Welche Probleme zeigt schon unser Mini-Sprachmodell?*

Problem	Was ist das?	Beispiel aus unserem Märchenmodell
Bias	Fehler oder Vorurteile, die aus den Trainingsdaten übernommen werden	In jedem Set sind nur **1–2 von 8 Hauptfiguren weiblich**. Das Modell erzeugt also fast nur männliche Helden.
Halluzinationen	Falsche oder erfundene Inhalte, die glaubwürdig wirken	„Ein Drache verlor seine Zahnbürste im Thronsaal." Das klingt nach einem echten Märchen, steht aber in keinem Text.
Trainingsdaten	Das Modell kann nur, was in seinen Daten steckt	Das Set-D-Modell kennt kein „Schloss" und in Set A gibt es kein „Ein König". Jedes Set ergibt ein anderes Modell.
Stereotype Antworten	Immer gleiche Rollenbilder und Muster	Die Hexe zaubert immer, der König ist reich (golden), die Prinzessin wird nie Ritterin. Neue Sätze wiederholen diese Muster.
Unsinn trotz hoher Wahrscheinlichkeit	Wahrscheinlich heißt nicht richtig	„Es war einmal eine Hexe, der mit Senf würzte." Jeder Schritt ist erlaubt, der Satz ist trotzdem falsch.

→ *Übertrag auf echte Sprachmodelle (ChatGPT, Claude, Gemini …)*

- Echte LLMs funktionieren im Prinzip **genauso**, nur mit Milliarden Texten, Wortteilen als Tokens und sehr vielen Parametern.
- Die gleichen Probleme treten auf: Bias aus Internettexten, erfundene Quellen und Fakten, veraltetes Wissen, Stereotype.

✓ *Konsequenzen für den Umgang mit KI*

Prüfen
KI-Texte mit verlässlichen Quellen gegenchecken

Hinterfragen
Wer oder was fehlt? Welche Rollenbilder entstehen?

Verantworten
Die Verantwortung für den Text bleibt beim Menschen

© 2026 E. Bechstädt · CC BY-SA 4.0 · creativecommons.org/licenses/by-sa/4.0/deed.de