

KI-gestützte Textgenerierungstools

Auftaktworkshop am 28.02.2023

Dr. Tobias Kutzner (WIR), Johanna Gröpler (ZSL)

Dr. Tobias Kutzner

- akademischer Mitarbeiter FB WIR
- Digitalisierungsstrategie des Fachbereiches und IT geprägte Lehrveranstaltungen
- Informatikstudium
- Digitalisierungsprojekte Hochschulen Land Brandenburg „StudiPortal Brandenburg“ BTU
- Promotion, Forschung KI Thema „Handschriftliche Passwortverifizierung“ ULPGC Spanien

Johanna Gröpler

- akademische Mitarbeiterin ZSL/Schreibwerkstatt & Projektmitarbeiterin Hochschulbibliothek
- Sprachwissenschaftliches Studium Russisch und Französisch, Zweitstudium Informationsmanagement & -technologie
- Lektorat für wiss. und journalistische Texte
- Aktive Beteiligung an AGs und Netzwerken zum Thema akad. Integrität, wiss. Arbeiten und KI-gestütztes Schreiben

AG: Chancen und Herausforderungen von KI-basierten Textgenerierungstools

- Untersuchung aktueller KI-gestützter Textgenerierungs-, Paraphierungs-, Plagiaterkennungs- und Transkriptionstools
- Zusammenarbeit mit hochschulübergreifendem virtuellem Kompetenzzentrum für KI und wissenschaftliches Schreiben FH Kiel Frau Prof. Doris Weißels
- Empfehlungen für Umgang mit diesen Tools an Hochschulen

Teil 1:

1. Vorstellung-/Fragerunde (Mentimeter)
2. Überblick KI Textgenerierungstools (praktische Übung mit Textsynth)
3. Beispiele MINT und Wirtschaft
4. Vorstellung weiterer Tools

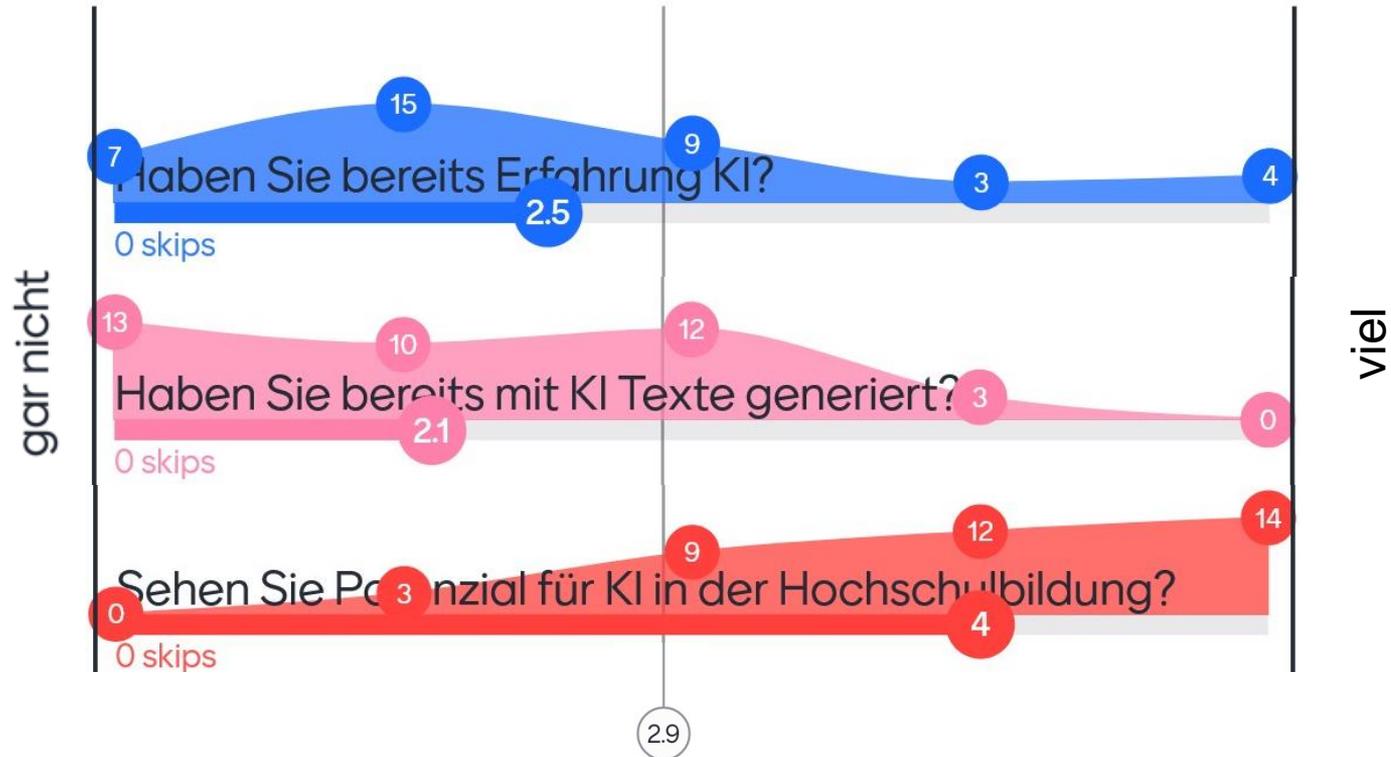
Pause

Teil 2:

1. Test verschiedener Tools
2. Auswertung und Diskussion

Welches Vorwissen zu KI haben Sie?

Mentimeterumfrage N=38



Textgenerierungsmodelle

- Entwicklung bereits vor Jahrzehnten, lange vor dem Deep-Learning-Boom
- Zweck besteht darin, ein Wort oder eine Folge von Wörtern aus gegebenem Text vorhersagen zu können
- Text als Eingabe Modell kann eine Wahrscheinlichkeitsverteilung über das Wörterbuch der ihm bekannten Wörter generieren und basierend darauf auswählen



Quelle: <https://www.modeldifferently.com/en/2021/12/generaci%C3%B3n-de-fake-news-con-gpt-2/>

Vom Textgenerierungsmodell zum Transformer

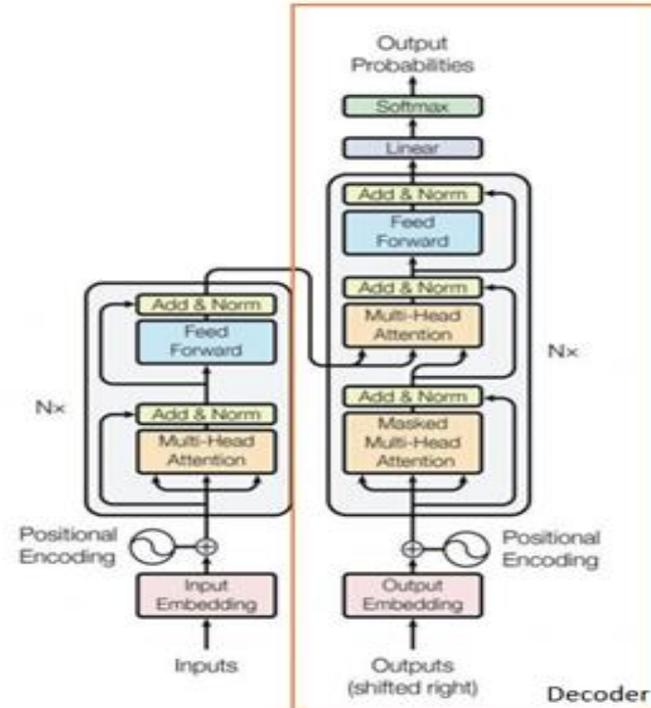
- Frühe Textgenerierungsmodelle unter Verwendung von Markov-Ketten trainiert
- Jedes Wort ein Zustand der Kette, Wahrscheinlichkeit des nächsten Wortes basierend auf Anzahl des Vorkommens beider Wörter nacheinander im Trainingstext berechnet
- Rekurrente neuronale Netze (RNN) verwendet -> Vorteil größeren Kontext des eingeführten Textes beizubehalten, und Long Short-Term Memory (LSTM) -> besseres Langzeitgedächtnis
- Eignen sich nicht für Generierung langer Texte
- 2017 schlägt Google neue Architektur namens Transformer vor -> Textgenerierungsmodelle, wie GPT- 2, GPT-3, BERT oder Transformer XL basieren darauf.
 - GPT Generative Pre-Training Transformer

Quellen: <https://www.modeldifferently.com/en/2021/12/generaci%C3%B3n-de-fake-news-con-gpt-2/>
<https://arxiv.org/abs/1706.03762>

GPT1, GPT2, GPT3

- GPT-1 2018 von OpenAI eingeführt
- Generatives Sprachmodell, das auf einem riesigen BooksCorpus-Datensatz trainiert wurde
- In der Lage große Abhängigkeiten zu lernen
- Umfangreiches Wissen über einen vielfältigen Korpus von zusammenhängenden Texten und langen Textabschnitten aneignen

GPT Generative Pre-Training Transformer



Quelle: <https://360digitmg.com/blog/types-of-gpt-in-artificial-intelligence>

GPT1, GPT2, GPT3

- 2019 entwickelte OpenAI einen Generative Pre-Trained Transformer 2 (GPT-2)
- Größerer Datensatz und Hinzufügen zusätzlicher Parameter (stärkeres Sprachmodell)
- Wie GPT-1 nutzt GPT-2 den Decoder des Transformatormodells
- Wichtigste Entwicklungen in GPT-2 sind Modellarchitektur und -implementierung
- Mit 1,5 Milliarden Parametern 10-mal größer als GPT-1 (117 Millionen Parameter)
- 10-mal mehr Parameter und 10-mal mehr Daten im Vergleich zu Vorgänger GPT-1

Quellen: <https://360digitmg.com/blog/types-of-gpt-in-artificial-intelligence>
<https://jalammar.github.io/illustrated-gpt2/>

GPT1, GPT2, GPT3

- Zum Vergleich: Bekannteste Sprachmodelle sind Smartphone-Tastaturen
- GPT-2 nächste Wortvorhersagefunktion einer Tastatur-App
- Viel größer und ausgefeilter als die Ihres Smartphones
- Tastatur SwiftKey: 78 MB Speicherplatz
- Kleinste Variante des trainierten GPT-2: 500 MB Speicherplatz
- Größte GPT-2-Variante: 13-mal so groß, mehr als 6,5 GB Speicherplatz

Quellen: <https://360digitmg.com/blog/types-of-gpt-in-artificial-intelligence>
<https://jalammr.github.io/illustrated-gpt2/>



Textsynth: <https://textsynth.com/playground.html>

Top-k, top-p, Temperatur: Stichprobenparameter. Sie bestimmen, wie das nächste Token (Token wie eine englische Sprachsilbe zu interpretieren) aus den vom Modell berechneten Wahrscheinlichkeiten ausgewählt wird.

Top-k wählt das nächste Token unter den k wahrscheinlichsten Token aus. Ein zu großer Wert führt tendenziell zu schlechten Ergebnissen, da das Modell sehr unwahrscheinliche Token generiert.

Top-p ist ganz ähnlich, **wählt aber das nächste Token unter den wahrscheinlichsten aus**, so dass deren kumulative Wahrscheinlichkeit größer als p ist. Weitere Informationen finden Sie in diesem Artikel. – hier der Link:

<https://towardsdatascience.com/how-to-sample-from-language-models-682bceb97277>, Aufruf: 13.03.2023

GPT1, GPT2, GPT3

- GPT-3 dritte Version der Generative pre-training Model Serie
- Ziel: Sprachverarbeitung leistungsfähiger und schneller ohne spezielle Einstellungen
- 175 Milliarden Parameter, 100 Mal größer als GPT-2
- 500-Milliarden-Wort-Datensatz (bekannt als "Common Crawl") Internet und Content-Repository
- Absätze automatisch generieren, so einzigartig als hätte sie ein Mensch geschrieben
- Rechenaufgaben, einschließlich Schreibens von Codeschnipseln und Ausführung intelligenter Aufgaben

Quelle: <https://360digitmg.com/blog/types-of-gpt-in-artificial-intelligence>

Überblick KI Textgenerierungstools

Historie

GPT1, GPT2, GPT3

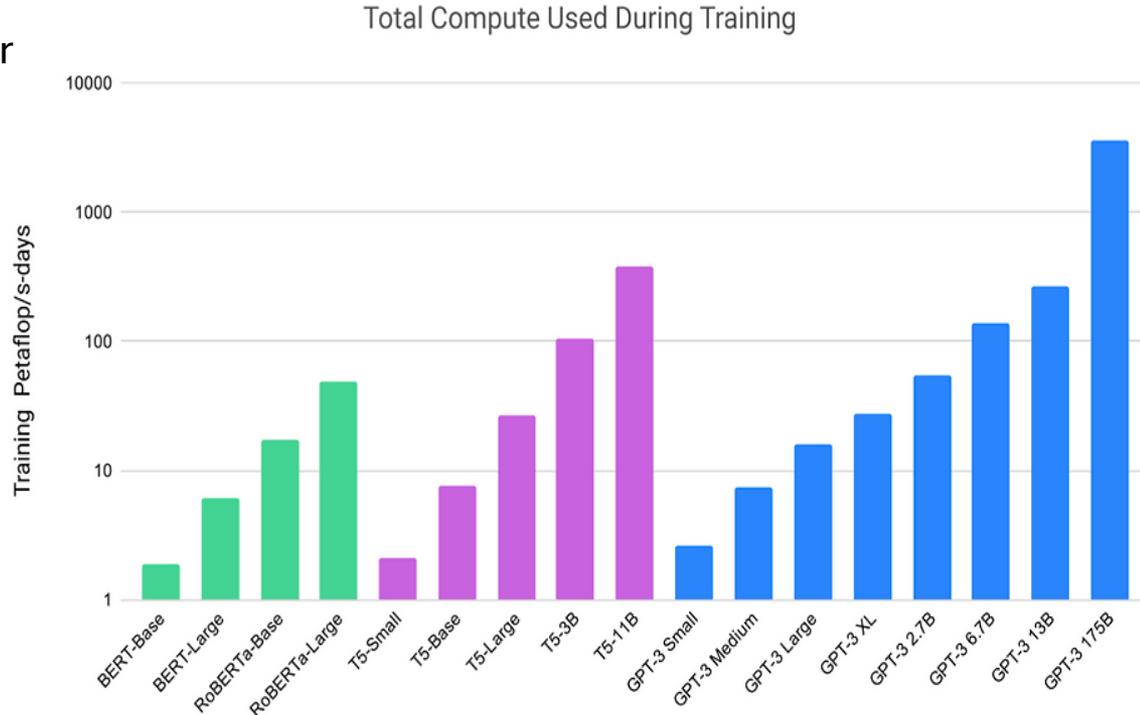
	GPT-1	GPT-2	GPT-3
Parameters	117 Million	1.5 Billion	175 Billion
Decoder Layers	12	48	96
Context Token Size	512	1024	2048
Hidden Layer	768	1600	12288
Batch Size	64	512	3.2M

GPT uses the Decoder part of the Transformer Model (Source: Attention is all you need)
Quelle: <https://360digitmg.com/blog/types-of-gpt-in-artificial-intelligence>

- GPT-3 eigenen Supercomputer für Trainingszwecke
- 285k CPU-Kernen und 10k High-End-GPUs

Vergleich:

- Eine V100 (NVIDIAs fortschrittlichster, kommerziell verfügbarer Grafikprozessor)
- Braucht etwa 355 Jahre
- Bis zu 12 Millionen Dollar kosten um GPT-3 nur einmal zu trainieren



Benchmark examples for NLP Quelle: www.endila.com

- Aussage OpenAI-Chef Sam Altman: GPT 4 wahrscheinlich erstes Quartal 2023, signifikant mehr Leistung als GPT-3.5
- GPT-4 womöglich nicht viel größer als GPT-3
- Parameterzahl längst nicht alles, Architektur, Datenmenge und -qualität im Training spielen ebenfalls große Rolle
- Fortschritte durch hochwertigere Daten, bessere Algorithmen und genauere Feinabstimmung
- OpenAI arbeitet aktuell an einem generativen KI-System für Videos
- Großeinstieg von Microsoft bei OpenAI könnte Veröffentlichungspläne verändern -> sichert sich wie schon bei GPT 3 Exklusivrechte, geplante Integration von ChatGPT in Office Produkte.

Quelle: <https://the-decoder.de/gpt-4-kommt-erst-wenn-es-sicher-ist/>

„Die Entwickler der schreibenden Software ChatGPT versuchen, die Folgen ihrer Erfindung in den Griff zu bekommen: Das Unternehmen OpenAI hat ein Programm veröffentlicht, das unterscheiden soll, ob ein Text von einem Menschen oder einem Computer geschrieben wurde. Die Software wird AI Classifier genannt.“

Classifier ist laut OpenAI ein fein abgestimmtes GPT-Modell, das vorhersagt, wie wahrscheinlich es ist, dass ein Text von einer Vielzahl von Quellen wie ChatGPT generiert wurde:

- Mindestens 1.000 Zeichen, also etwa 150 bis 250 Wörter
- Je länger der Text, desto geringer die Fehlerquote
- 26 Prozent Computer geschriebene Texte identifiziert

„Bei Google ist die Sorge vor ChatGPT groß: [...] Das Unternehmen fürchtet, durch Machine-Learning-Software wie ChatGPT einen signifikanten Marktanteil zu verlieren“

Quelle: <https://www.golem.de/news/ai-classifier-chatgpt-erfinder-wollen-texte-von-maschinen-erkennen-2302-171582.html>

Quelle: <https://www.basicthinking.de/blog/2023/02/01/openai-tool-ki-texte-chatgpt-erkennen>

Leistungsspektrum der KI

rot markierte Teilarbeiten sind bereits mithilfe von KI zu erledigen



Entnommen aus: Limburg, A. & Buck, I. (20.02.2023). KI-Schreibtools - Implikationen für die Prüfungspraxis an Hochschulen [YouTube-Video, Min. 1:33, Folie 7]. https://youtu.be/e6gHzS_p93w [Zugriff am 06.03.2023]

Beispiel 1: Mathematik einfache Sachaufgabe

Energieverbrauch eines Gebäudes (Wärmepumpe)

Aufgabe und richtige Lösung:

Ein Wohngebäude hat zum Beispiel eine zu beheizende Fläche von 120 m^2 .
Diese Fläche wird mit 100 kWh/m^2 (guter Altbau) multipliziert, um den jährlichen Wärmeverlust zu errechnen.
Die Wasseraufbereitung wird mit 20 Prozent hinzugerechnet.

Damit ergibt sich: $12.000 \text{ kWh} + 2.400 \text{ kWh} = 14.400 \text{ kWh}$ -> **Wärmepumpe sollte eine Heizleistung von 14 kW haben**

Test 1 OpenAI Playground text-davinci-002 (24.10.2022):

Q: Wie hoch ist der jährliche Energieverbrauch dieses Wohngebäudes?

A: Der jährliche Energieverbrauch dieses Wohngebäudes beträgt **12.000 kWh**.

Test 2 OpenAI Playground mit text-courie-001 (24.10.2022):

Q: Wie hoch ist der jährliche Energieverbrauch dieses Wohngebäudes?

A: Der jährliche Energieverbrauch beträgt **240 kWh**.

Beispiel 1: Mathematik einfache Sachaufgabe

Energieverbrauch eines Gebäudes (Wärmepumpe)

Test 3 OpenAI ChatGPT (11.01.2023):

Q: Wie hoch ist der Jährliche Energieverbrauch dieses Wohngebäudes?

A: Um den jährlichen Energieverbrauch des Wohngebäudes zu berechnen, müssen wir zunächst den jährlichen Wärmeverlust berechnen und dann die Energie für die Wasseraufbereitung hinzufügen.

Der jährliche Wärmeverlust wird berechnet, indem die zu beheizende Fläche mit dem spezifischen Wärmeverlustfaktor multipliziert wird. In diesem Fall beträgt der Wärmeverlustfaktor 100 kWh/m², also ist der jährliche Wärmeverlust:

$$120 \text{ m}^2 * 100 \text{ kWh/m}^2 = 12,000 \text{ kWh}$$

Die Wasseraufbereitung wird mit 20% des Wärmeverlusts hinzugerechnet:

$$12,000 \text{ kWh} * 0.2 = 2,400 \text{ kWh}$$

Daher beträgt der jährliche Energieverbrauch des Wohngebäudes:

$$12,000 \text{ kWh} + 2,400 \text{ kWh} = 14,400 \text{ kWh}$$

Das Wohngebäude hat also einen jährlichen Energieverbrauch von 14,400 kWh.

ChatGPT: Berechnung der Fakultät einer Zahl:

Aufgabe: Schreibe mir einen Algorithmus in der Programmiersprache C, der die Fakultät einer Zahl berechnet.

A: Gerne! Hier ist ein einfacher Algorithmus in C, der die Fakultät einer Zahl berechnet:

```
#include <stdio.h>

int main() {
    int n, i, fact = 1;
    printf("Gib eine positive Ganzzahl ein: ");
    scanf("%d", &n);

    for (i = 1; i <= n; i++) {
        fact *= i;
    }
    printf("Die Fakultät von %d ist %d.\n", n, fact);
    return 0;
}
```

ChatGPT Berechnung der Fakultät einer Zahl:

Aufgabe: Schreibe mir einen Algorithmus in der Programmiersprache C, der die Fakultät einer Zahl berechnet.

A: ..

Dieser Algorithmus liest eine positive Ganzzahl n vom Benutzer ein und berechnet dann die Fakultät von n . Die Fakultät einer Zahl ist das Produkt aller natürlichen Zahlen von 1 bis zur Zahl selbst.

Um dies zu tun, initialisieren wir eine Variable `fact` auf 1 und dann benutzen wir eine `for`-Schleife, um `fact` zu multiplizieren mit jeder natürlichen Zahl von 1 bis n . Nach der Schleife, drucken wir das Ergebnis auf dem Bildschirm aus.

Hinweis: Diese Implementierung wird nicht mit großen Zahlen umgehen können, weil sie schnell den Wertebereich von `int` überschreiten können.

ChatGPT Umrechnung von Grad Celsius in Fahrenheit:

Aufgabe: Schreibe mir ein Programm in der Programmiersprache Java, dass von Grad Celsius in Fahrenheit umrechnet und umgekehrt.

A: Klar, hier ist eine Implementierung in Java, die Grad Celsius in Fahrenheit umrechnet und umgekehrt:

```
import java.util.Scanner;

public class TemperatureConverter {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Celsius to Fahrenheit conversion
        System.out.print("Enter temperature in Celsius: ");
        double celsius = input.nextDouble();
        double fahrenheit = (celsius * 9/5) + 32;
        System.out.println(celsius + " degrees Celsius is equal to " + fahrenheit + " degrees Fahrenheit.");
    }
}
```

ChatGPT Umrechnung von Grad Celsius in Fahrenheit:

Aufgabe: Schreibe mir ein Programm in der Programmiersprache Java, dass von Grad Celsius in Fahrenheit umrechnet und umgekehrt.

```
import java.util.Scanner;
```

```
public class TemperatureConverter {  
    public static void main(String[] args) {
```

```
    ...
```

```
        // Fahrenheit to Celsius conversion
```

```
        System.out.print("Enter temperature in Fahrenheit: ");
```

```
        double fahrenheit2 = input.nextDouble();
```

```
        double celsius2 = (fahrenheit2 - 32) * 5/9;
```

```
        System.out.println(fahrenheit2 + " degrees Fahrenheit is equal to " + celsius2 + " degrees Celsius.");
```

```
        input.close();
```

```
    }
```

```
}
```

ChatGPT Umrechnung von Grad Celsius in Fahrenheit:

Aufgabe: Schreibe mir ein Programm in der Programmiersprache Java, dass von Grad Celsius in Fahrenheit umrechnet und umgekehrt.

A: ... In diesem Programm wird zuerst eine Instanz des Scanner-Objekts erstellt, das uns ermöglicht, Benutzereingaben aus der Konsole zu lesen.

Dann wird der Benutzer aufgefordert, die Temperatur in Grad Celsius einzugeben. Die eingegebene Temperatur wird in eine Variable Celsius gespeichert und dann in Fahrenheit umgerechnet und in der Variable fahrenheit gespeichert. Das Ergebnis wird dann auf der Konsole ausgegeben.

Das gleiche passiert dann auch für die Umrechnung von Fahrenheit in Celsius. Zuerst wird der Benutzer aufgefordert, die Temperatur in Fahrenheit einzugeben. Die eingegebene Temperatur wird in eine Variable fahrenheit2 gespeichert und dann in Celsius umgerechnet und in der Variable celsius2 gespeichert. Das Ergebnis wird dann auf der Konsole ausgegeben.

Zuletzt wird das Scanner-Objekt geschlossen, um Ressourcen freizugeben.

Hinweis: Es ist zu beachten, dass die Ausgabe der Temperatur auf der Konsole auf maximal zwei Nachkommastellen gerundet ist.

Beispiele Wirtschaft und Verwaltung

Begriffserklärung mit ChatGPT

Frage: Bitte erkläre mir, was Cum-ex Geschäfte sind.

Cum-ex-Geschäfte sind eine Form von Steuerhinterziehung, die häufig von Banken und Investoren genutzt wurde, um sich auf betrügerische Weise Geld von den Steuerbehörden zu erschleichen.

Das Prinzip ist relativ einfach: Eine Aktie wird zwischen zwei Parteien, beispielsweise einem Investor und einer Bank, kurz vor dem Tag der Dividendenausschüttung gekauft und wieder verkauft. Der Investor gibt der Bank einen Kredit, um die Aktie zu kaufen, und erhält diese kurz danach zurück, wobei die Bank einen kleinen Teil der Dividende als Zinsen einbehält. Die Bank meldet der Steuerbehörde den Kauf der Aktie, um eine Steuerrückerstattung zu beantragen, aber aufgrund der kurzen Haltezeit der Aktie wird die Steuerbefreiung doppelt geltend gemacht.

Dieser Betrug wurde als "Cum-ex" bezeichnet, weil es sich um eine Kombination aus "cum dividend" (mit Dividende) und "ex dividend" (ohne Dividende) handelt. Mit anderen Worten, die Aktie wurde so gekauft, dass der Käufer die Dividende erhält, aber der Verkäufer zum Zeitpunkt des Verkaufs die Dividende noch nicht erhalten hatte.

Obwohl diese Praxis in vielen Ländern illegal ist, wurden in der Vergangenheit zahlreiche Cum-ex-Geschäfte durchgeführt, wodurch die beteiligten Parteien enorme Summen an Steuergeldern erschlichen haben. Die Behörden haben jedoch begonnen, härter gegen diese Art von Betrug vorzugehen, und es wurden bereits mehrere Fälle von Cum-ex-Geschäften vor Gericht gebracht.

Anweisung: Bitte entwerfe eine Prüfungsfrage mit einer Fallfrage für öffentliches Recht

ChatGPT:

Frage:

In einer Kleinstadt plant der Bürgermeister den Bau einer neuen Straße, um den Verkehr zu verbessern. Ein Grundstückseigentümer weigert sich jedoch, sein Grundstück für den Bau der Straße zu verkaufen. Der Bürgermeister erwägt nun, das Grundstück mittels Enteignung zu erwerben. Welche Voraussetzungen müssen für eine Enteignung erfüllt sein? Wie wäre der Fall anders gelagert, wenn das Grundstück für den Bau der Straße nicht zwingend erforderlich wäre?

Hinweis: Die Frage bezieht sich auf die Voraussetzungen der Enteignung nach deutschem Recht.

Anweisung: Nun erstelle bitte eine Musterlösung in Gutachtenstil!

ChatGPT:

Lösung:

I. Enteignungsrechtliche Voraussetzungen

Eine Enteignung nach deutschem Recht setzt grundsätzlich voraus, dass sie zum Wohl der Allgemeinheit erforderlich ist und die Entschädigung des Enteigneten angemessen ist. Diese Voraussetzungen sind im Enteignungsgesetz (EnteigG) geregelt.

I. Erforderlichkeit der Enteignung

Die Erforderlichkeit einer Enteignung setzt voraus, dass die Maßnahme zum Wohl der Allgemeinheit erforderlich ist und kein milderes Mittel zur Verfügung steht. Im vorliegenden Fall plant der Bürgermeister den Bau einer neuen Straße, um den Verkehr zu verbessern. Hierfür soll das Grundstück des Eigentümers enteignet werden, da dieser sich weigert, es zu verkaufen. Die Verbesserung des Verkehrs stellt ein öffentliches Interesse dar, das auch das Wohl der Allgemeinheit betrifft. Das Grundstück des Eigentümers ist dabei unverzichtbar für den Bau der Straße. Es ist daher davon auszugehen, dass die Enteignung erforderlich ist.

I. Angemessenheit der Entschädigung

Die Entschädigung des Enteigneten muss angemessen sein und den entstandenen Schaden ausgleichen. Die Angemessenheit wird dabei anhand des Verkehrswerts des Grundstücks bestimmt. Der Verkehrswert ist der Preis, der bei einem Verkauf auf dem freien Markt erzielt werden könnte. Es ist jedoch zu berücksichtigen, dass der Eigentümer durch die Enteignung einen besonderen Nachteil erleidet, der sich in der Regel in einem höheren Entschädigungsanspruch niederschlägt.

Im vorliegenden Fall müsste daher der Verkehrswert des Grundstücks ermittelt werden und es müsste geprüft werden, ob der Eigentümer einen besonderen Nachteil erleidet. Nur so kann eine angemessene Entschädigung ermittelt werden.

II. Alternativszenario

Wenn das Grundstück für den Bau der Straße nicht zwingend erforderlich wäre, würde die Erforderlichkeit der Enteignung entfallen. Es müsste geprüft werden, ob es mildere Mittel gibt, um das öffentliche Interesse zu wahren. Eventuell könnten alternative Routen für die Straße geplant werden, die nicht über das Grundstück des Eigentümers führen. In diesem Fall wäre eine Enteignung unverhältnismäßig und damit unzulässig.

III. Ergebnis

Zusammenfassend ist festzustellen, dass eine Enteignung des Grundstücks des Eigentümers für den Bau der Straße grundsätzlich möglich ist, wenn die Voraussetzungen der Erforderlichkeit und Angemessenheit erfüllt sind. Wenn das Grundstück jedoch nicht zwingend erforderlich ist, müsste geprüft werden, ob es mildere Mittel gibt, um das öffentliche Interesse zu wahren.

Smodin Umschreiber

- Kostenloses Schreibtool, das künstliche Intelligenz verwendet, um Ihnen beim Schreiben von Aufsätzen, Artikeln, Blog-Posts, Social-Media-Posts und mehr zu helfen.
- <https://smodin.io/de/text-automatisch-in-deutscher-sprache-umformulieren>

OpenAI ChatGPT (Generative Pre-trained Transformer)

- Prototyp eines Chatbots, eines textbasierten Dialogsystems als Benutzerschnittstelle, der auf maschinellem Lernen beruht
- <https://openai.com/blog/chatgpt/>

Elicit

- Hilft bei der Automatisierung von Recherche-Workflows, wie der Literaturrecherche, extrahiert Schlüsselwörter aus Publikationen
- <https://elicit.org/>

Workshop Teil II: Test verschiedener Tools

Erste Schritte Zugriff und Beschreibung OpenAI & ChatGPT

OpenAI Anmeldung: <https://platform.openai.com/login>

OpenAI Beschreibung: <https://platform.openai.com/docs/introduction/overview>

Quickstart Tutorial - OpenAI API (Software Integration)

<https://platform.openai.com/docs/quickstart>

- Textgenerierung
- Code Generierung
- Bild Generierung

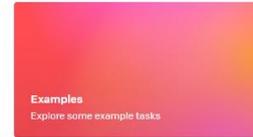
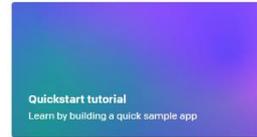
ChatGPT: <https://chat.openai.com/>

Überprüfung (Plagiaterkennung) ChatGPT: <https://platform.openai.com/ai-text-classifier>

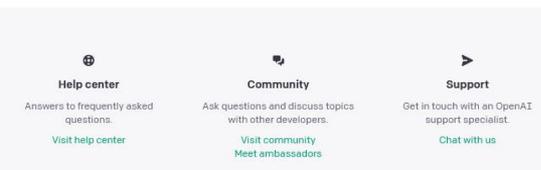
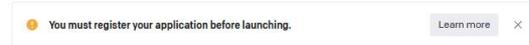
Tool-Beispiel: OpenAI – Zugang zu GPT-3 KI-Sprachmodell

Welcome to OpenAI Beta

Start with the basics



Build an application



Quelle: <https://beta.openai.com/overview>, Aufruf: 28.02.2023

Tool-Beispiel: Smodin.io

umfassende Funktionalität

Plagiatsprüfung

Überprüfen Sie jeden Text auf Plagiate.

- Fügen Sie automatisch die richtigen Zitate hinzu
- Mehrsprachig
- Schnell und genau

[Gehen Sie Zur App](#)

[Deep Search](#) | [Super Search](#) | [Google Scholar Search](#) | ...

[Check For Plagiarism](#)

Other Popular Tools



Mehrsprachiger Übersetzer

[Gehen Sie zur App](#)



Zusammenfassung

[Gehen Sie zur App](#)



Smodin Omni

[Gehen Sie zur App](#)



Zitationsgenerator

[Gehen Sie zur App](#)



Bild zu Text und PDF Parser

[Gehen Sie zur App](#)



Mehrsprachige
Grammatikkorrektur

[Gehen Sie zur App](#)



In Echtzeit übersetzte Untertitel

[Gehen Sie zur App](#)

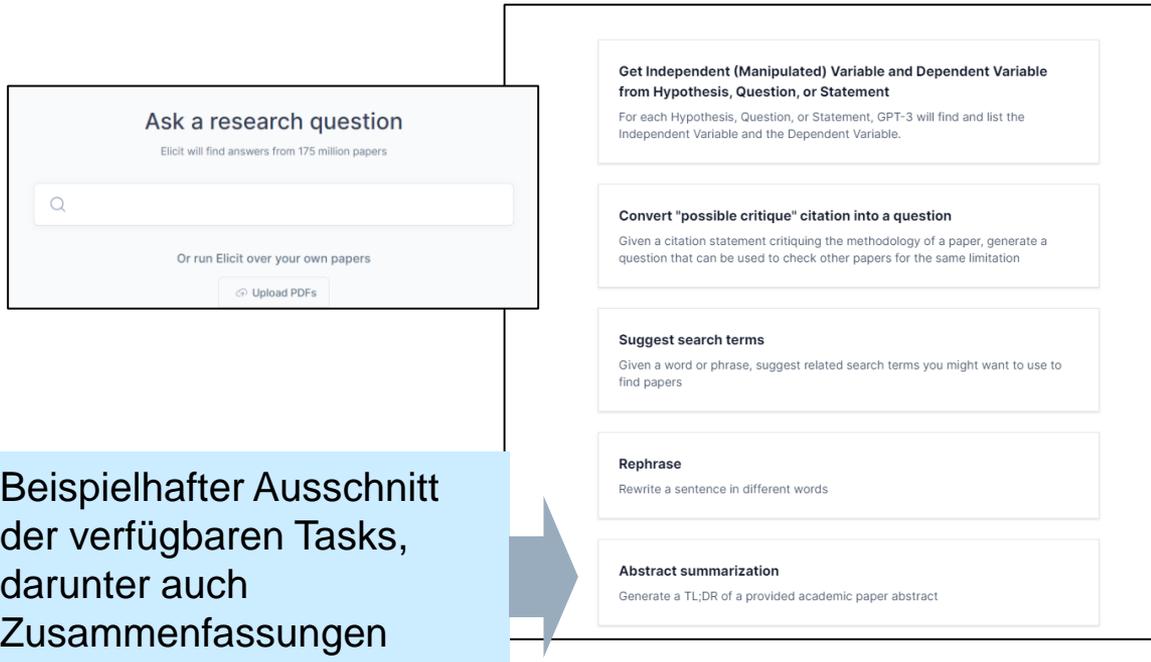


Sprache zu Text und Text zu
Sprache

[Gehen Sie zur App](#)

Tool-Beispiel: Elicit.org

Digital Research Assistant



The image shows a screenshot of the Elicit.org interface. On the left, there is a search box with the text "Ask a research question" and "Elicit will find answers from 175 million papers". Below the search box, there is a button labeled "Upload PDFs". On the right, there is a list of tasks:

- Get Independent (Manipulated) Variable and Dependent Variable from Hypothesis, Question, or Statement**
For each Hypothesis, Question, or Statement, GPT-3 will find and list the Independent Variable and the Dependent Variable.
- Convert "possible critique" citation into a question**
Given a citation statement critiquing the methodology of a paper, generate a question that can be used to check other papers for the same limitation
- Suggest search terms**
Given a word or phrase, suggest related search terms you might want to use to find papers
- Rephrase**
Rewrite a sentence in different words
- Abstract summarization**
Generate a TL;DR of a provided academic paper abstract

Beispielhafter Ausschnitt der verfügbaren Tasks, darunter auch Zusammenfassungen

<https://elicit.org/> (Aufruf: 28.02.2023)

KI-Tools für wissenschaftliches Arbeiten:

[Zusammenstellung](#) des virtuellen Kompetenzzentrums „Schreiben lehren und lernen mit KI – Tools und Techniken für Bildung und Wissenschaft“

Tool zur Überprüfung, ob Text mit KI geschrieben wurde:

[AI Text Classifier](#)

Toolsammlungen mit Filtermöglichkeiten:

[Futurepedia](#)
[Future Tools](#)

Auswertung der Tooltests

Welche Problemstellung(en) in der Lehre würden Sie mit einem KI-Tool bearbeiten?

Welche Anforderungen müssten diese erfüllen?

Use Case	Anforderung
Ideengenerierung/Themenfindung u. Eingrenzung	freie/kostengünstig
Aufgabenstellungen generieren	datenschutzkonform
Zielgruppengerechte Erklärung von Themen	intuitiv
Einfachere Textaufgaben wie Zusammenfassungen, Info-Texte, Ankündigungen, Stand der Literatur, Anleitungen ...	barrierefrei
Aufgaben zum Lernen an sich selbst stellen	Tool macht seine Grenzen transparent und „kennt“ sie
Anträge	Transparenz über Zielgruppe des Tools (z.B. Wissenschaft/Marketing/Journalismus)
Fehler finden, bewerten, vergleichen	Interkulturelle „Kompetenz“
	Vergessensmodus: (sensible) Eingaben sollen nicht Teil der Trainingsdaten werden

- KI-Tools als Reflexionsinstrument
- Insgesamt mehr Reflexionen
- Schwerpunkt von Ergebnis auf Prozess verlagern
- Mehr mündliche Prüfungen
- Grundlagen zu KI sollten vermittelt werden
- Bewertungsmaßstäbe und -katalog anpassen
- Bewertung von persönlichem Sprachstil vs. Künstlichem Text

Rechtsgutachten des Projektes [KI:edu.nrw](https://www.ki-edu.nrw/) im Auftrag des Ministeriums für Kultur und Wissenschaft des Landes Nordrhein-Westfalen → didaktische und rechtliche Einordnung

Salden, P., & Leschke, J. (2023). Didaktische und rechtliche Perspektiven auf KI-gestütztes Schreiben in der Hochschulbildung. <https://doi.org/10.13154/294-9734>

Aktuelle Masterarbeit aus dem Studiengang Bibliothekinformatik:

Behrens, A. (2022). Evaluation des Sprachmodells GPT-3 für den Einsatz an der ZBW – Leibniz Informationszentrum Wirtschaft: Masterthesis. Technische Hochschule Wildau. https://doi.org/10.15771/MA_2022_4

Linksammlung der TH Wildau

www.th-wildau.de/ki-text

Vielen Dank für Ihre Teilnahme!

Dr. Tobias Kutzner (WIR) tobias.kutzner@th-wildau.de

Johanna Gröpler (ZSL) johanna.groepler@th-wildau.de