

Sprachmodelle in



[KI in den MINT-Fächern]

Harald Schilly
März 2024

CoCalc

von Sagemath, Inc.



CoCalc

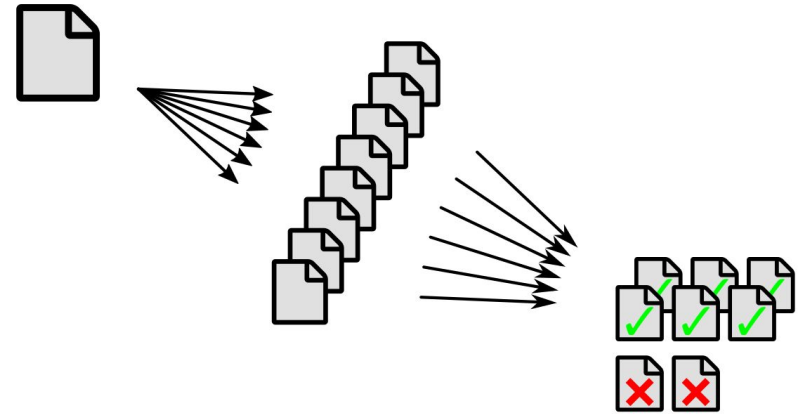
Online Plattform für wissenschaftliches Rechnen, echtzeit Kollaboration, und für den Unterricht.



- Im Kern: ein **einheitliches Interface** mit zahlreichen “Editoren”
- Zugriff auf eine **große Sammlung von open source Software**:
 - Python, R, Octave, Julia, C/C++, Jupyter Notebooks, X11 Emulator, etc.
- Alles läuft **“remote”** am Server, daher arbeiten alle in exakt derselben Umgebung:
 - Es gibt separate Arbeitsumgebungen, genannt “Projekte”.
 - Ein solches Projekt ist ähnlich einem Account auf einem Linux Server.
- Populär ist das **“Jupyter Notebook”**, welches sowohl als eigenständige tief integrierte Variante verfügbar ist, als auch über den klassischen oder “lab” Jupyter Server.
- Außerdem: “Sage Worksheets”, “LaTeX Editor”, “Linux Terminal”, etc.

CoCalc /2

- Unterricht über einen “Kurs Manager”
- Konzeptuell analog zu einem “PC Labor”
 - Jede/r Studierende hat ein **eigenes “Projekt”**
 - Aufgaben sind Verzeichnisse von Dateien
 - Der Unterrichtende verteilt Dateien zur Bearbeitung an alle → benotet → und sammelt sie später wieder ein



Homework

Assignment

Grading

User Interface

Hierarchisches Interface

Projekte

- isolierte Arbeitsbereiche
- Sammlung von Dateien
- Kollaboratoren
- Prozesse

Editoren (für jede unterstützte Datei)

- rechts: Jupyter Notebook
- links: Dateien

The screenshot displays a Jupyter Notebook environment. On the left, a file explorer shows a directory structure with files like 'task_output.log', 'collect-2024-03.csv', and 'tesla-analyze.ipynb'. The main area shows a Jupyter Notebook with the following code and output:

```
In [9]: fig, ax = plt.subplots(1)
t[recently].plot(ax = ax, x='timestamp', y='blevel', grid=True)
ax2 = t[recently].plot(ax = ax, x='timestamp', y=['brange', 'brangeest'],
secondary_y=True, grid=True)
_ = ax.set_ylabel('Battery [%]')
_ = ax2.set_ylabel('Range [km]')
_ = ax.xaxis.set_major_formatter(
matplotlib.dates.DateFormatter("%Y-%m-%d %H:%M"))
)
```

Out [9]:

The plot shows three data series: 'blevel' (blue), 'brange' (orange), and 'brangeest' (green). The x-axis is 'timestamp' ranging from 2024-03-11 00:00 to 2024-03-17 00:00. The left y-axis is 'Battery [%]' ranging from 50 to 90. The right y-axis is 'Range [km]' ranging from 200 to 450. All three series show a sharp increase around 2024-03-13 00:00.

```
In [10]: tripl = (t.timestamp >= '2021-05-16 08:00') & (t.timestamp <= '2021-05-16
14:00')
```

```
In [11]: trip_triest_home = (t.timestamp >= '2021-08-31 10:00') & (t.timestamp <=
'2021-08-31 18:00')
```

UI/Chat

Es gibt zwei Arten von Chats:

Eigenständig

Eine Datei in dem Projekt.

Zur Kommunikation mit anderen im selben Projekt.

“Side-Chat”

Neben jedem Editor kann ein Chat aufgeklappt werden.

Dieser Chat ist spezifisch für diese Datei

The screenshot shows the JupyterLab interface. The top bar includes navigation icons for Files, Log, Find, Settings, Processes, and Eclipse.sagews. Below this is a toolbar with icons for File, Save, TimeTravel, Run, Stop, Restart, in, out, and a refresh icon. A menu bar contains Help, Modes, #, Data, Control, and Program. A sidebar on the left shows a file explorer with categories like Plots, Calculus, Linear, Graphs, Numbers, and Rings. The main area is a code editor with Python code for finding the maximum obscuration of an eclipse and converting UTC to Chicago time. On the right, a chat panel is open, showing a message from Hal Snyder and a response from @Richard Miles. A red box highlights the chat panel, and red arrows point to the 'Chat' icon in the top right and the 'Chat' button in the chat panel.

```
105 42 |  
106 43 |  
107 44 |  
108 45 |  
109 46 |  
110 47 |  
111 48 |  
112 49 |  
113 50 |  
114 51 |  
115 52 |  
116 53 |  
117 |  
118 |  
119 1 |  
120 2 |  
121 3 |  
122 4 |  
123 5 |  
124 6 |  
125 7 |  
126 8 |  
127 9 |  
128 10 |
```

Click "Chat" at upper right to open a chat panel for this file.

Besonderheiten

- **Erwähnungen:** mittels @[name] kann jemand direkt angesprochen werden
- **Programmieren:** beliebiger Code kann im Chat direkt ausgeführt werden

Large Language Models /1



- Innovatives KI Produkt – aktuell in einer Hype-Phase
- Werden auf einer extrem großen Menge an Text (inkl. Code) trainiert
- Trainingsziel: Vorhersage des restlichen Textes aus einem Teil davon
 - Das klingt sehr allgemein, kann aber effektiv unbegrenzt komplex sein!
- Herausforderung: Modell muss ein grundsätzliches “abstraktes” Verständnis der Welt aufbauen.

Large Language Models /2



Anwendungsmethoden

- **Verarbeitung:** Einen längeren Text als “Datensatz” schnell durchlesen und daraus Schlussfolgerungen zu ziehen, Frage zu Beantworten, Korrekturen vorzunehmen, etc.
- **Generativ:** ein *kurzer* “Auftrag” (“Prompt”) soll einen *längeren* Text generieren.
- **Kreativität:** widersprüchliche bzw. sinnfreie Prompts führen schnell zu “kreativen” aber auch “sinnlosen” Antworten → oft nicht gewollt
- **Ungeeignet:** Wortspiele, klassische Datenverarbeitung (Rechnen, Sortieren, etc.) – klassische Methoden sind genauer und effizienter.



LLMs in CoCalc?

LLM ↔ CoCalc?



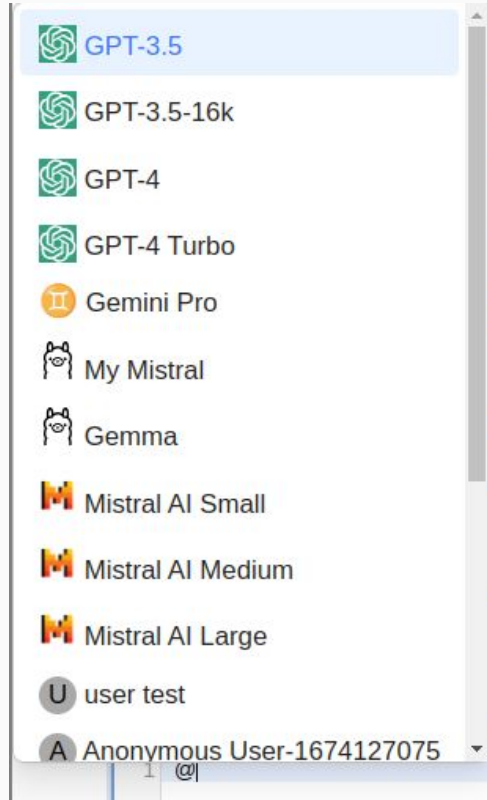
- **Nutzen:** Wie kann so ein LLM Tool eingebunden werden, sodass es auch wirklich einen **Mehrwert** bringt?
- **Transparenz:** Poweruser haben schon früh zurück gemeldet, dass automatische Anfragen Unbehagen auslösen.
- **Erkennen:** Wie muss das Interface aufbereitet werden, dass diese Tools auch erkennbar sind – bzw. klar genug kommunizieren, was passiert?
- **Annehmen:** Werden User das Tool überhaupt annehmen? Braucht es Einschränkungen?

LLM ↔ CoCalc















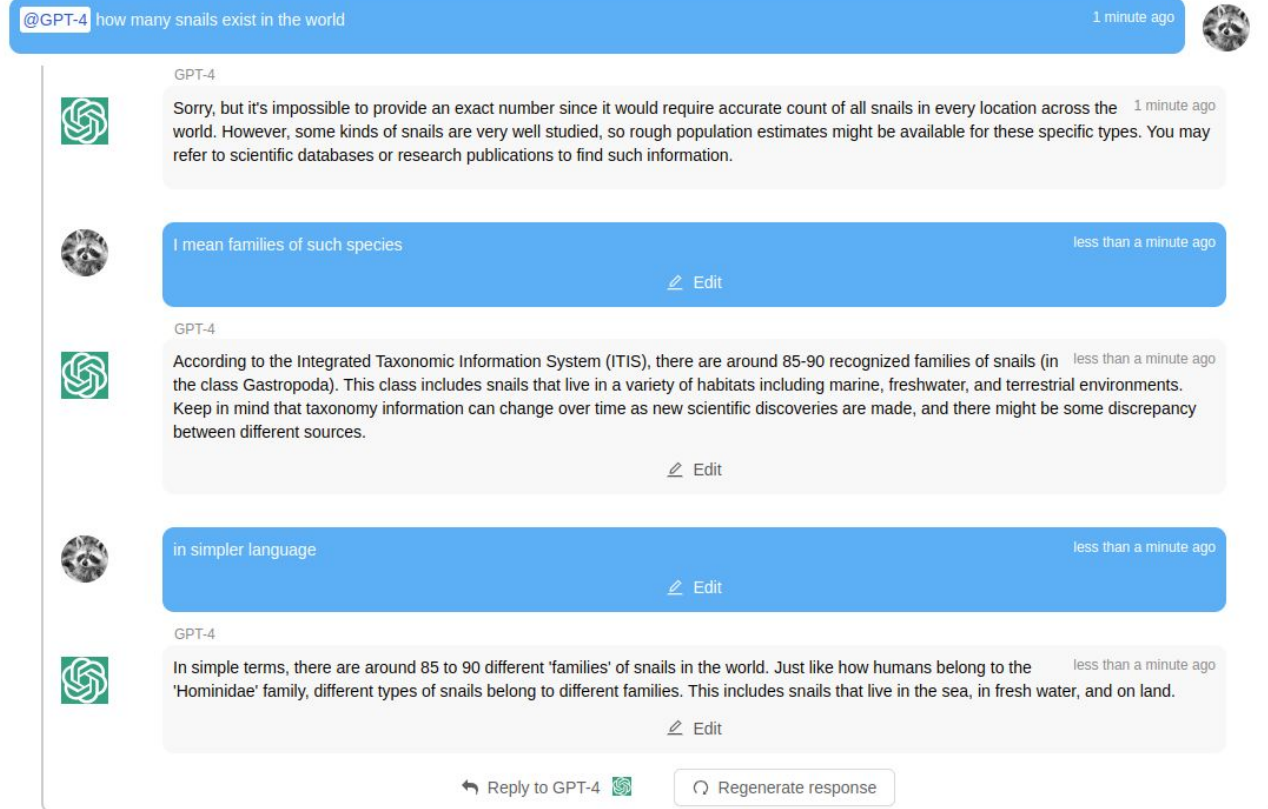
- Primäre Interaktion: **über Chats!**
- CoCalc sendet konstruierte Text-Prompts bzw. einen Teil der Kommunikation eines Chats an einen Server, welches ein LLMs bereitstellt.
- Der User hat **volle Kontrolle**, welches verwendet wird.
- Die Antwort des Servers wird in **Echtzeit** wiedergegeben.
- Die Anfrage und Kommunikation ist **transparent und vollständig nachvollziehbar**.
- Insbesondere gibt es **keine versteckten/impliziten Anfragen!**
- **Kurse:** AI Assistenten lassen sich partiell oder komplett für Studierende deaktivieren.

In Chats werden LLMs wie auch andere im Projekt mittels “@...” angesprochen.



A vertical list of chat models with their respective icons:

-  GPT-3.5
-  GPT-3.5-16k
-  GPT-4
-  GPT-4 Turbo
-  Gemini Pro
-  My Mistral
-  Gemma
-  Mistral AI Small
-  Mistral AI Medium
-  Mistral AI Large
-  user test
-  Anonymous User-1674127075



A chat conversation with GPT-4. The user asks: "@GPT-4 how many snails exist in the world" (1 minute ago). GPT-4 responds: "Sorry, but it's impossible to provide an exact number since it would require accurate count of all snails in every location across the world. However, some kinds of snails are very well studied, so rough population estimates might be available for these specific types. You may refer to scientific databases or research publications to find such information." (1 minute ago). The user replies: "I mean families of such species" (less than a minute ago). GPT-4 responds: "According to the Integrated Taxonomic Information System (ITIS), there are around 85-90 recognized families of snails (in the class Gastropoda). This class includes snails that live in a variety of habitats including marine, freshwater, and terrestrial environments. Keep in mind that taxonomy information can change over time as new scientific discoveries are made, and there might be some discrepancy between different sources." (less than a minute ago). The user replies: "in simpler language" (less than a minute ago). GPT-4 responds: "In simple terms, there are around 85 to 90 different 'families' of snails in the world. Just like how humans belong to the 'Hominidae' family, different types of snails belong to different families. This includes snails that live in the sea, in fresh water, and on land." (less than a minute ago). At the bottom, there are buttons for "Reply to GPT-4" and "Regenerate response".

Chat mit LLM: Code generieren und ausführen!



@Mistral AI Large plot 100 random points in 2D and make a linear fit

3 minutes ago



Mistral AI Large



Sure, here's a simple way to do it in Python using matplotlib for plotting and numpy for linear algebra:

3 minutes ago

Edit Copy Run Python3

```
import matplotlib.pyplot as plt
import numpy as np

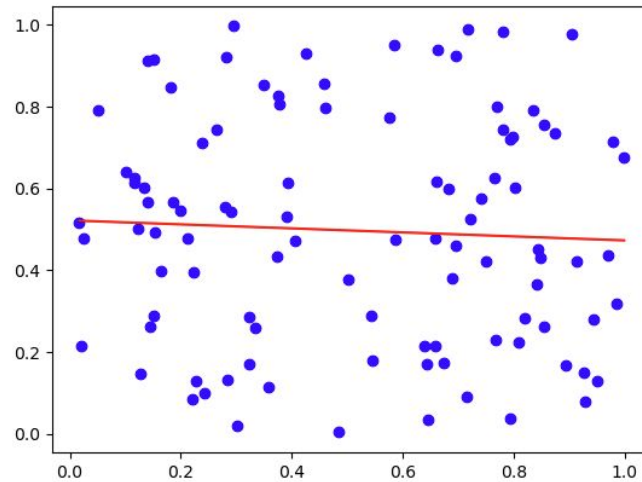
# Generate 100 random points
x = np.random.rand(100)
y = np.random.rand(100)

# Calculate linear regression
A = np.vstack([x, np.ones(len(x))]).T
m, c = np.linalg.lstsq(A, y, rcond=None)[0]

# Plot original points
plt.scatter(x, y, color='b')

# Plot linear fit
x_line = np.array([min(x), max(x)])
y_line = m*x_line + c
plt.plot(x_line, y_line, color='r')

plt.show()
```



This script first generates 100 random points in 2D. Then it calculates the linear regression of these points using the least squares method. Finally, it plots the original points and the linear fit. The linear fit is a line that best approximates the original points according to the least squares method.



CoCalc nützt LLMs, indem mittels komplexer, **automatisch generierter Prompts**, Probleme ausformuliert werden und das LLM Lösungen vorschlagen soll. Das läuft transparent ab, da die Anfrage + Antwort in dem Side-Chat zu sehen ist. Generierter Code wird explizit mittels Copy/Paste in das Notebook eingepflegt.

```
In [3]: import numpy as np
xx = np.vektor([1,25,2,31])
xx^2
```

Out[3]: 🗨️ Help me fix this...

```
AttributeError                                Traceback (most recent call last)
Cell In[3], line 2
      1 import numpy as np
----> 2 xx = np.vektor([1,25,2,31])
      3 xx^2
File ~/local/lib/python3.10/site-packages/numpy/_init_.py:333, in _getatt
    330     "Removed in NumPy 1.25.0"
    331     raise RuntimeError("Tester was removed in NumPy 1.25.")
--> 333 raise AttributeError("module {!r} has no attribute "
    334                        "{!r}".format(__name__, attr))
AttributeError: module 'numpy' has no attribute 'vektor'
```

Get Help from **Mistral AI Small** free – Fast, ...

The following will be sent to Mistral AI Small:

@Mistral AI Small help me fix my code.

▼ Context

I ran a cell in a Jupyter notebook.

I received the following error:

```
-----
AttributeError
Traceback (most recent call last)
Cell In[3], line 2
      1 import numpy as np
      2 xx = np.vektor([1,25,2,31])
      3 xx^2
```

Cancel ▶ Ask Mistral AI Small (enter)



@Mistral AI Small help me fix my code.

1 minute ago

▶ Context

Mistral AI Small

The error is due to a typo in your code. The correct function in numpy for creating a vector is `np.array()` , not `np.vektor()` . Also, the correct syntax for squaring a numpy array in python is `**2` , not `^2` . Here is the corrected code:

1 minute ago

 Edit  Copy  Run  Python 3 (ipy...)

```
import numpy as np
xx = np.array([1,25,2,31])
xx**2
```

Diese “Help-me-fix” Funktion ist die mit Abstand beliebteste in CoCalc!

Generiere ein komplettes Jupyter Notebook



Create Jupyter Notebook using AI

Create Notebook Using

GPT-3.5 free – Fast, great for ...

Generate a Jupyter Notebook using the following Jupyter kernel:

Python 3 (ipykernel)

Provide a detailed description of the notebook you want to create:

Explain me, how I can use scipy to fit a normal distribution to 100 random numbers

Example: "Fit a statistical model to this time series of monthly values: 72, 42, 63, 44, 46, 51, 47, 39, 21, 31, 19, 22. Then plot it with extrapolation."

[➤ Create Notebook using GPT-3.5 \(shift+enter\)](#)

The following will be sent to GPT-3.5:

Explain directly and to the point, how to do the following task in the programming language "Python 3 (ipykernel)", which I will be using in a Jupyter notebook. Performing the standard libraries on the following task: summary

GPT-3.5 generated notebook

This notebook was generated in CoCalc by GPT-3.5 using the prompt:

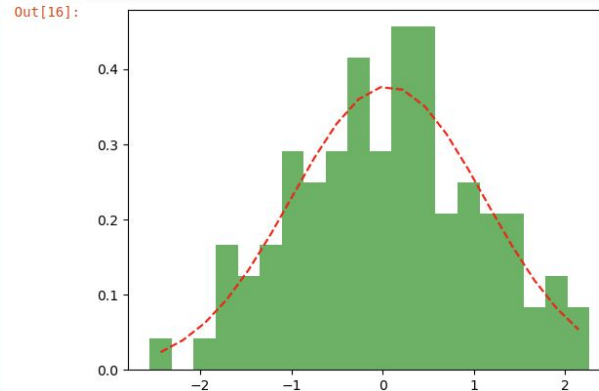
Explain me, how I can use scipy to fit a normal distribution to 100 random numbers

```
In [14]: def normal_distribution(x, mean, std_dev):  
         return norm.pdf(x, loc=mean, scale=std_dev)
```

Plot the histogram of the data and the fitted normal distribution curve.

```
In [15]: np.random.seed(0)  
         data = np.random.randn(100)
```

```
In [16]: plt.hist(data, bins=20, density=True, alpha=0.6, color='g')  
         plt.plot(bin_centers, normal_distribution(bin_centers, *params), 'r--')  
         plt.show()
```



Generiere ein neue Zelle in einem Jupyter Notebook



```
In [4]: A = np.random.randn(10, 10)
x = np.random.randn(10)
```

<> Code Text Paste Generate...

Generate code cell using Mistral AI Large paid – Most ...

Describe what the new cell should do.

erkläre mir, wie ich die beiden multiplizieren kann

The following will be sent to Mistral AI Large:

Create a new code cell for a Jupyter Notebook.

Kernel: "Python 3 (ipykernel)".

Programming language: "python". The entire code cell must be in a single code block. Enclose this block in triple backticks. Do not say what the output will be. Add comments as code comments. The previous code cell

Cancel ▶ Generate Using Mistral AI Large (shift+enter)

The code below was generated by Mistral AI Large using this prompt:

erkläre mir, wie ich die beiden multiplizieren kann

Sure, here's a Python code block that explains how to multiply the two numpy arrays A and x :

```
In [5]: # Import the necessary library
import numpy as np

# Since A is a 2D array and x is a 1D array, we can't
directly multiply them.
# Instead, we can perform a dot product which is a specific
type of matrix multiplication.
# In Python, we use the np.dot() function for this.

# The dot product of A and x is a 1D array where each
element is the sum of the product of each row in A and x.
result = np.dot(A, x)



# Print the result
print(result)
```

Out[5]: [-0.35398204 1.46208126 1.05058476 0.71689691 -2.10546553
 2.68928537
-0.13447704 1.8199953 -0.73311295 2.16318215]

“erkläre mir, wie ich die beiden multiplizieren kann” wird verstanden, weil als Kontext die letzte Code-Zelle mitgeschickt wird!

Zellen in einem Jupyter Notebook erklären!



 Get explanation of this code from  **Mistral AI Large** paid – Most ... ▾

The following will be sent to Mistral AI Large:

@Mistral AI Large Explain the following Python 3 (ipykernel) code that is in a Jupyter notebook:

▼ Details

Edit Copy Run Python 3 (ipy...

```
def f(name: str):
    n1 = sum(ord(_) for _ in name)
    n2 = 1
    for c in name:
        n2 = n1 * ord(c)
    return n1 + n2

f("CoCalc")
```

Cancel Ask Mistral AI Large (enter)

@Mistral AI Large Explain the following Python 3 (ipykernel) code that is in a Jupyter notebook: less than a minute ago

► Details Edit


Mistral AI Large

This Python function `f(name: str)` takes a string `name` as input. less than a minute ago

- `n1 = sum(ord(_) for _ in name)`: This line calculates the sum of the ASCII values of all characters in the input string `name` and assigns it to `n1`.
- `n2 = 1`: Initializes `n2` to 1.
- The for loop `for c in name:` iterates over each character in the string `name`.
- `n2 = n1 * ord(c)`: In each iteration, `n2` is updated to be the product of the current `n1` and the ASCII value of the current character `c`.
- `return n1 + n2`: The function returns the sum of `n1` and the final value of `n2`.

When `f("CoCalc")` is called, it calculates and returns the sum of the total ASCII values of all characters in "CoCalc" and the product of the total ASCII values and the ASCII value of the last character "c".

Edit

Reply to Mistral AI Large  Regenerate response

... und mit "Reply..." können follow-up Fragen gestellt werden.

0.002 seconds Ⓜ Explain Format Copy 21

```
In [9]: def f(name: str):
        n1 = sum(ord(_) for _ in name)
        n2 = 1
        for c in name:
            n2 = n1 * ord(c)
        return n1 + n2

        f("CoCalc")

Out[9]: 54900
```



AI Generate LaTeX Formula using GPT-3.5

Select language model:  **GPT-3.5** free – Fast, great for ...

The selected AI language model will generate a LaTeX formula the description. [?](#)

[?](#) integral x from 0 to infinity of e^{1-x^2} / x dx

Generate

LaTeX

```
\int_{0}^{\infty} \frac{e^{1-x^2}}{x} \, dx
```

Preview

$$\int_0^{\infty} \frac{e^{1-x^2}}{x} dx$$

Note: You might have to ensure that `\usepackage{am`

```
latex/tex.tex
1 \documentclass{article}
2 \title{Title}
3 \author{Author}
4 \usepackage{amsmath}
5 \begin{document}
6
7  $\int_{0}^{\infty} \frac{e^{1-x^2}}{x} \, dx$ 
8
9
10 \end{document}
```

PDF - Preview

Print Downlo... Bigger Smaller Zoom t... Z

$$\int_0^{\infty} \frac{e^{1-x^2}}{x} dx$$



Generate LaTeX Formula using Mistral AI Medium

Select language model:  Mistral AI Medium free - Inte... ▼

The selected AI language model will generate a LaTeX formula the description. [?](#)

[?](#) maxwell equations, explain each variable

Generate

```


$$\begin{aligned}
 \nabla \cdot \mathbf{E} &= \frac{\rho}{\varepsilon_0} \\
 \nabla \cdot \mathbf{B} &= 0 \\
 \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\
 \nabla \times \mathbf{B} &= \mu_0 \mathbf{J} + \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t}
 \end{aligned}$$


```

Note: The variables in Maxwell's equations are defined as follows:

```

\begin{itemize}
\item  $\mathbf{E}$  is the electric field
\item  $\mathbf{B}$  is the magnetic field
\item  $\rho$  is the charge density
\item  $\mathbf{J}$  is the current density
\item  $\varepsilon_0$  is the permittivity of free space
\item  $\mu_0$  is the permeability of free space
\item  $\nabla$  is the del operator, representing spatial derivatives
\item  $\frac{\partial}{\partial t}$  is the partial derivative with respect to time.
\end{itemize}

```

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}$$

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t}$$

Note: The variables in Maxwell's equations are defined as

- \mathbf{E} is the electric field
- \mathbf{B} is the magnetic field
- ρ is the charge density
- \mathbf{J} is the current density
- ε_0 is the permittivity of free space
- μ_0 is the permeability of free space
- ∇ is the del operator, representing spatial derivatives
- $\frac{\partial}{\partial t}$ is the partial derivative with respect to time.

Note: You might have to ensure that `\usepackage{amsmath}`



latex/tex.tex

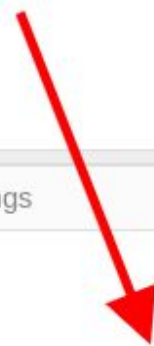
```
1 \documentclass{article}
2 \title{Title}
3 \author{Author}
4 \usepackage{amsmath}
5 \begin{document}
6
7 \int_{0}^{\infty} \frac{e^{1-x^2}}{x} \, dx
8
9 The \bold{entire sum} goes to zero!
10
11
12 \end{document}
```

Errors and Warnings

Errors (Latex)

Undefined control sequence. [Line 9 of tex.tex](#) Help me fix this...

- 1.9 The `\bold`



@GPT-3.5 help me fix my code. 2 minutes ago

► Context

GPT-3.5

The correct command to make text bold in LaTeX is `\textbf`, not `\bold`. Update line 10 to use `\textbf` instead. 2 minutes ...

Edit Copy Run Kernel...

The `\textbf{entire sum}` goes to zero!

and how to make it italic? 1 minute ago


GPT-3.5


To make text italic in LaTeX, you can use the command `\textit`. Update line 10 to make the text italic: 1 minute ago

Edit Copy Run Kernel...

The `\textit{entire sum}` goes to zero!



 **What would you like to do using GPT-4?** Tour X

Switch model:  **GPT-4** paid – Can follow com... ▼

Write a short summary about the development of AI since the 1960 to now. ✕

or 🔧 Fix Errors ✍️ Autocomplete 🗨️ Explain 👁️ Review 💬 Add Comments ⚡ Summarize

GPT-4 will see: Selection All None ↻ Update NOT Truncated (100% included)

No context from your file will be included.

You can try anything that you can possibly imagine: translate from one programming language to another, explain why code is slow, show the steps to solve an equation, etc.

➤ Ask GPT-4 (shift+enter)

Die Anfrage wird im Side-chat beantwortet und kann in das Dokument eingepflegt werden.

Ausgewählte Teile des Dokuments können als Anfrage abgeschickt werden. Prompts für “Fix Errors”, “Autocomplete”, “Summarize”, etc. sind mögliche Aufgaben.

AI has undergone significant growth since its inception in the 1950s. The earliest stages of AI, characterized by the 1960s, saw the development of problem-solving systems, linguistic, and perceptual processing. During the 1970s and 1980s, AI researchers focused on more formal knowledge representation and reasoning techniques. This was the beginning of expert systems, which sought to imitate the decision-making abilities of human experts.

In the 1990s, computational intelligence and machine learning began to take center stage, with algorithms such as neural networks, fuzzy systems, evolutionary computation, and later, deep learning. The introduction of these learning systems marked an important shift in AI research, with machines learning directly from data.

Since the 2000s to the present day, we’ve witnessed an explosion in AI due to advances in computational power, large-scale digitization of information, and improvements in machine learning algorithms. Today, AI applications span various domains, from automated driving to digital personal assistants and facial recognition systems.



Ganz generell können LLMs in *jedem* Editor aufgerufen werden. Das heißt, auch in einem Markdown Dokument kann es behilflich sein, es kann “Tasks” in einer Task-List zusammenfassen, in einem “*.c” file kann es Code generieren, usw.

Ausblick (Roadmap):

- Mehr Modelle, flexiblere Integration
- Flexiblere und anpassbare “System Prompts”
- Unterstützung von eigens gehosteten Modellen
- Detailverbesserungen

Zukünftige Ideen:

- Unterstützung im Unterricht: z.B. Assistent zum Korrigieren von Arbeiten (Notebooks, etc.)
- Generieren von Aufgaben
- Dokumentensuche

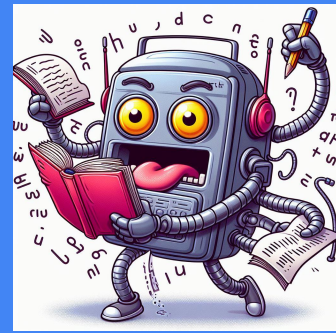
Wie gut werden LLMs Angenommen?



In der letzten Woche:

- ca. 17% aller aktiven Accounts verwenden ein LLM
- Es gab über 15000 Aufrufe
- Von denen verwenden
 - 67% eines der “Help me fix” Tools
 - 13% generieren Jupyter Zellen in Notebooks
 - 8% fragen in Chats genauer nach
 - 7% lassen sich Jupyter Zellen erklären
 - <1% generieren LaTeX Formeln
 - <1% generieren Jupyter Notebooks

Abrechnung



Die Firma hinter CoCalc ist sehr klein, kein VC funding. Daher muss genau auf das Geld geschaut werden:

- **Sub-cent Bereich:** Abfragen werden einzeln abgerechnet. Da es sich um Kleinstbeträge handelt, muss die Abrechnung robust für Beträge kleiner als \$0.01 sein.
- **Korrektheit:** Es darf nicht zu wenig verrechnet werden – sei es durch einen technischen Fehler, oder dass User die Abrechnung irgendwie versuchen zu umgehen.

FIN!

Harald Schilly

hsy@sagemath.com

cocalc.com



COCALC