Recherche:

* tokenization: <https://www.youtube.com/watch?v=D8j1c4NJRfo>
* batch: <https://www.youtube.com/watch?v=T0VSn-XkjGM>, <https://www.youtube.com/watch?v=cjBn4ATxZoQ>
* decoder: https://www.youtube.com/watch?v=Qo\_0PGusJfU
* pytorch: <https://www.youtube.com/watch?v=ORMx45xqWkA>, <https://www.youtube.com/watch?v=r2JKV4_zlmM>
* transformers: <https://www.youtube.com/watch?v=SZorAJ4I-sA>
* https://www.youtube.com/watch?v=QEaBAZQCtwE

Versuche, ob existierende KIs unser Problem lösen:

* ChatGPT, BingChat und Co: ohne Probleme (noch weiter ausführen)
* GPT4All:  
  - Mistral: klappt gut, auch auf DE (noch weiter ausführen)  
  - Falcon: Klappt kaum, lädt lange und stürzt ab, hängt sich oft auf (noch weiter ausführen)

Modell-Versuche:

* **Orca 2**:  
  - lädt erst lange (20Min) und ist dann abgebrochen  
  - im zweiten Versuch: Nach 72Min folgende Fehlermeldung:  
  „ValueError: The current `device\_map` had weights offloaded to the disk. Please provide an `offload\_folder` for them. Alternatively, make sure you have `safetensors` installed if the model you are using offers the weights in this format.”
* **OpenChat**: <https://huggingface.co/openchat/openchat_3.5>  
  - Funktioniert bei der Inatallation im Comand Prompt nicht
* Mistral-7B-v0.1: <https://huggingface.co/mistralai/Mistral-7B-v0.1>
* zephyr-7b-beta: <https://huggingface.co/HuggingFaceH4/zephyr-7b-beta>
* Llama-2-7b: <https://huggingface.co/meta-llama/Llama-2-7b>
* phi-1\_5: <https://huggingface.co/microsoft/phi-1_5>
* Llama-2-13b-chat-hf: <https://huggingface.co/meta-llama/Llama-2-13b-chat-hf>
* **GPT-2:** <https://huggingface.co/gpt2>  
  - funktioniert. Ladezeit: 1Min50Sek  
  - ERKLÄRUNG CODE:  
  *from transformers import pipeline, set\_seed:* Diese Zeile importiert die benötigten Funktionen und Module aus der Transformers-Bibliothek. Die pipeline-Funktion wird verwendet, um verschiedene NLP-Aufgaben wie Textgenerierung, Zusammenfassung usw. auszuführen. set\_seed wird genutzt, um einen bestimmten Zufallsgenerator festzulegen, was die Reproduzierbarkeit von Ergebnissen ermöglicht.

*generator = pipeline('text-generation', model='gpt2'):* Hier wird ein NLP-Pipeline-Objekt namens generator erstellt, das für die Textgenerierung verwendet wird. Es verwendet das Modell 'gpt2', ein vortrainiertes Modell für Textgenerierung, das Teil der Hugging Face Transformers-Bibliothek ist.

*set\_seed(42):* Diese Zeile setzt den Zufallsgenerator auf den Wert 42. Dies hilft, sicherzustellen, dass die generierten Texte bei wiederholter Ausführung des Codes dieselben bleiben, da der Zufallsgenerator auf denselben Startwert festgelegt ist.

*generator("Hello, I'm a language model,", max\_length=30, num\_return\_sequences=5)*: Hier wird die generator-Funktion aufgerufen, um Texte zu generieren. Der übergebene Starttext ist "Hello, I'm a language model,". Mit max\_length=30 wird die maximale Länge des generierten Textes auf 30 Token begrenzt. num\_return\_sequences=5 gibt an, dass fünf verschiedene Textsequenzen generiert werden sollen, basierend auf dem Starttext und den Modellparametern.

Zusammengefasst ruft dieser Code ein vortrainiertes GPT-2-Modell auf, um fünf verschiedene Textsequenzen mit einer maximalen Länge von 30 Tokens basierend auf dem Starttext "Hello, I'm a language model," zu generieren und verwendet dabei einen festgelegten Zufallsgenerator, um reproduzierbare Ergebnisse zu erhalten.