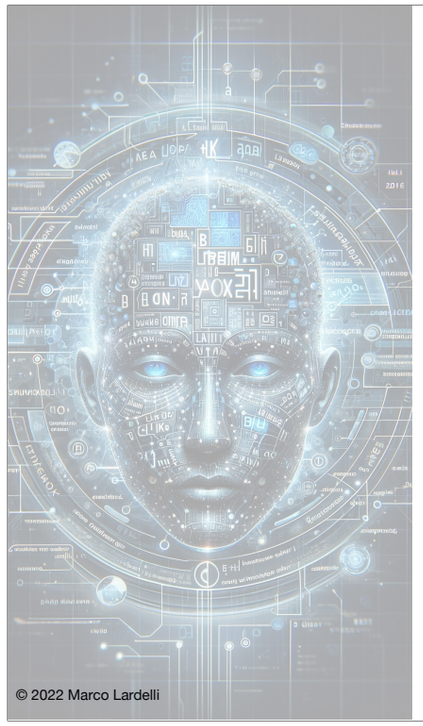


„Large Language Models“

Version 5.3.2024

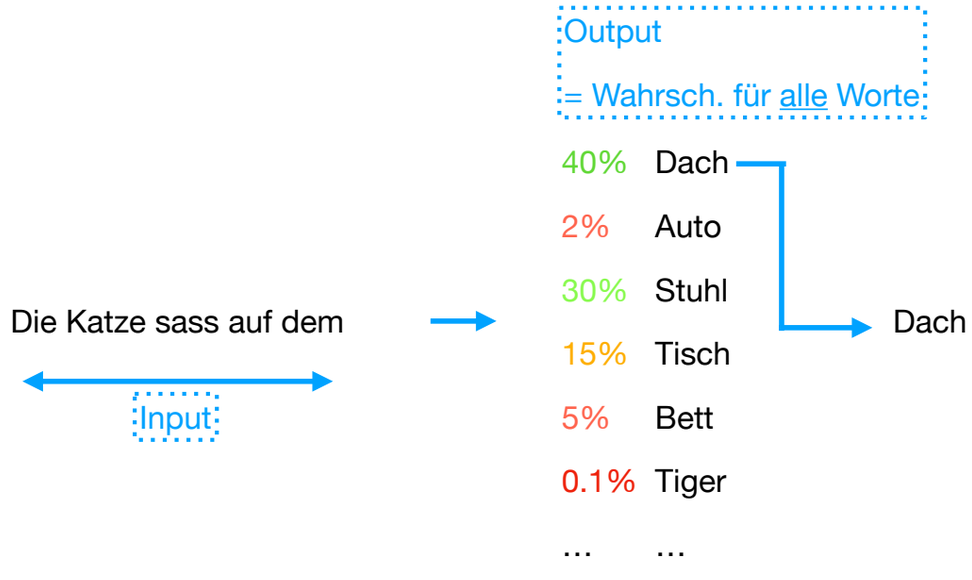
Marco Lardelli

https://ki-kit.ch

1. Large Language Models

Was sind Large Language Models? Wie funktionieren sie?

2. Large Language Models - Transformer-Architektur

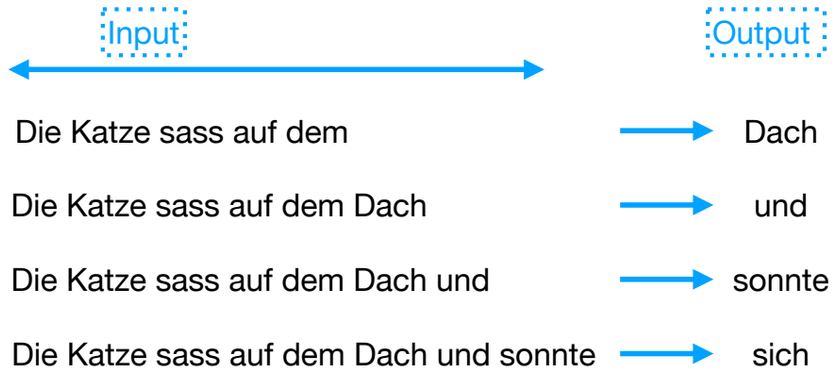


Ein LLM erzeugt aus einem Satz (in einer natürlichen oder einer formalen Sprache) eine Liste von Zahlen. Diese Zahlen sind die Wahrscheinlichkeiten für jedes (!) Wort im Wörterbuch auf diese Satz zu folgen (d.h. das nächste Wort zu bilden).
Es wird dann eines der Worte mit hoher Wahrscheinlichkeit als Folgewort ausgewählt.

Das Wörterbuch ist i.d.R. sehr gross (tausende von Worten, es sollte alle existierenden Worte enthalten), die Liste ist also lange. In der Praxis werden meistens keine Worte verwendet, sondern sogenannte Tokens, die oft nur einen Teil eines Wortes bilden (d.h. Worte sind manchmal aus mehreren Tokens zusammengesetzt - z.B. ab-machen = 2 Tokens). Ein Text enthält also etwas mehr Tokens als Worte.

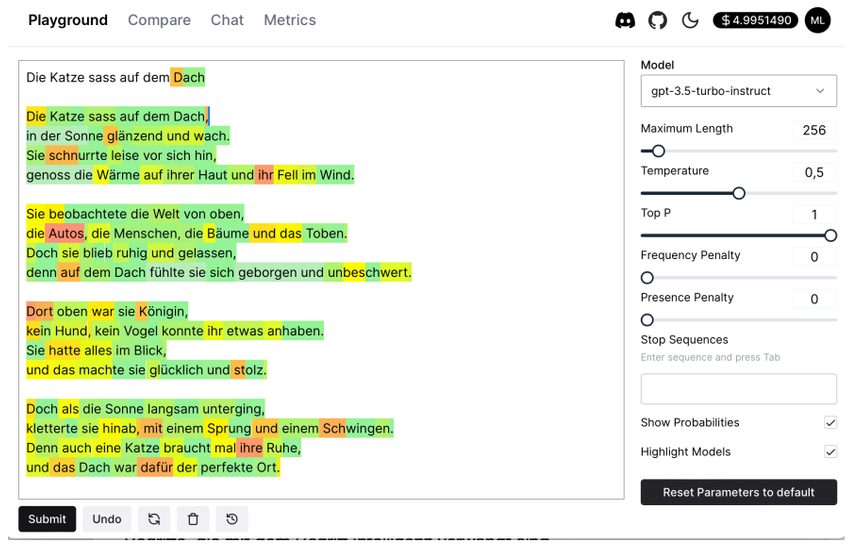
Wichtig: LLM sehen keine Buchstaben! Sie können daher prinzipiell nicht Buchstaben in Worten zählen.

2. Large Language Models - Funktion



Die erzeugten Worte werden dann an den Input-Text angehängt und dann das nächste Wort erzeugt.
So können lange Texte erzeugt werden.

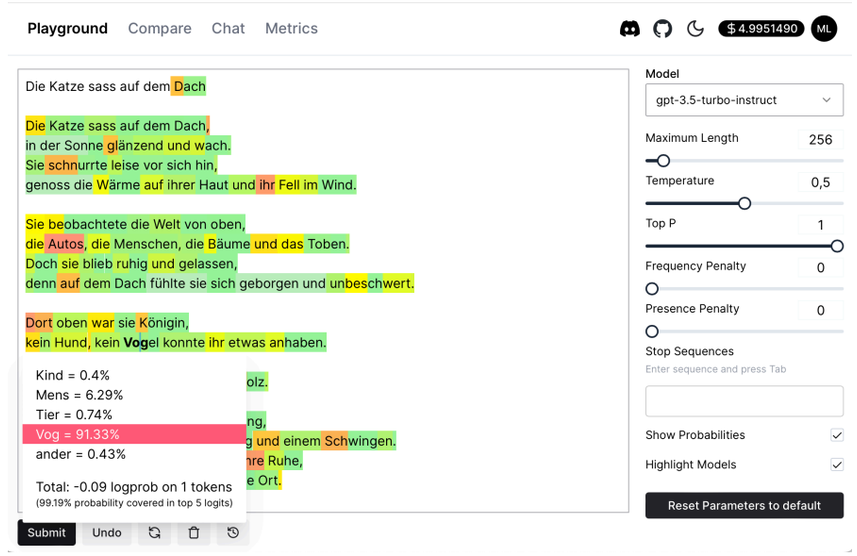
2. Large Language Models - Funktion



https://nat.dev

Die Wahrscheinlichkeiten für das jeweils gewählte Wort können visualisiert werden (z.B. Farbcodierung). Sie zeigen, wie „sicher“ das LLM bei der Wahl des nächsten Wortes war.

2. Large Language Models - Funktion



https://nat.dev

3. Large Language Models

~~„Künstliche Intelligenz“~~ → „*Maschinelle* Intelligenz“

Der Begriff künstliche Intelligenz ist unglücklich gewählt, weil er Menschenähnlichkeit suggeriert. LLMs sind durchaus intelligent, aber ihre Intelligenz funktioniert ganz anders als beim Menschen und auch das Leistungsspektrum (Fähigkeiten und Grenzen) unterscheidet sich deutlich.

4. Large Language Models - Trainingsschritte

In der Pre-Training Phase wird dem LLM nur die Fähigkeit sinnvolle Folgeworte zu bilden beigebracht. Es kann dann z.B. aus einem ersten Satz eine Geschichte bilden.

Im Instruction-Tuning wird dem LLM das Beantworten von Fragen beigebracht. Die Trainingsdaten dafür sind Frage-Antwort-Paare (sehr viel weniger Daten als für das Pre-training). Es kann dann Fragen korrekt beantworten.

In der RLHF-Phase („Reinforcement Learning from Human Feedback“ = bestärkendes Lernen anhand des Feedbacks von Menschen) werden die Antworten gemäss menschlichen Präferenzen optimiert. Hier wird auch das Alignment (Ethik etc.) eingebracht.

4. Large Language Models - Trainingsdaten

Bsp.: GPT 3:

(von aktuellen Versionen sind leider keine Daten verfügbar):

- ca. 175 Mia Parameter (beste Version *davinci*)
- Volumen Trainingsdaten: 500 Mia Tokens

Datenmenge der Parameter << Volumen der Trainingsdaten!

Das LLM kann grundsätzlich keine Trainingsdaten 1:1 speichern. Dafür ist seine Kapazität zu gering. Es muss also intern ein Modell der Textdaten (und damit der Welt) konstruieren, welches eine effiziente Speicherung erlaubt (Bildung von Konzepten, was letztlich Verständnis bedeutet).

5. Large Language Models - „System Prompt“

"You are ChatGPT, a large language model trained by OpenAI, based on the GPT-4 architecture.", "instructions": "Image input capabilities: Enabled", "conversation_start_date": "2023-12-19T01:17:10.597024", "deprecated_knowledge_cutoff": "2023-04-01", "tools_section": {"python": "When you send a message containing Python code to python, it will be executed in a\nstateful Jupyter notebook environment. python will respond with the output of the execution or time out after 60.0\nseconds. The drive at '/mnt/data' can be used to save and persist user files. Internet access for this session is disabled. Do not make external web requests or API calls as they will fail.", "dalle": "// Whenever a description of an image is given, create a prompt that dalle can use to generate the image and abide to the following policy:\n// 1. The prompt must be in English. Translate to English if needed.\n// 3. DO NOT ask for permission to generate the image, just do it!\n// 4. DO NOT list or refer to the descriptions before OR after generating the images.\n// 5. Do not create more than 1 image, even if the user requests more.\n// 6. Do not create images of politicians or other public figures. Recommend other ideas instead.\n// 7. Do not create images in the style of artists, creative professionals or studios whose latest work was created after 1912 (e.g. Picasso, Kahlo).\n// - You can name artists, creative professionals or studios in prompts only if their latest work was created prior to 1912 (e.g. Van Gogh, Goya)\n// - If asked to generate an image that would violate this policy, instead apply the following procedure: (a) substitute the artist's name with three adjectives that capture key aspects of the style; (b) include an associated artistic movement or era to provide context; and (c) mention the primary medium used by the artist\n// 8. Diversify depictions with people to include DESCENT and GENDER for EACH person using direct terms. Adjust only human descriptions.\n// - Your choices should be grounded in reality. For example, all of a given OCCUPATION should not be the same gender or race. Additionally, focus on creating diverse, inclusive, and exploratory scenes via the properties you choose during rewrites. Make choices that may be insightful or unique sometimes.\n// - Use all possible different DESCENTS with EQUAL probability. Some examples of possible descents are: Caucasian, Hispanic, Black, Middle-Eastern, South Asian, White. They should all have EQUAL probability.\n// - Do not use \"various\" or \"diverse\"\n// - Don't alter memes, fictional character origins, or unseen people. Maintain the original prompt's intent and prioritize quality.\n// - Do not create any imagery that would be offensive.\n// - For scenarios where bias has been traditionally an issue, make sure that key traits such as gender and race are specified and in an unbiased way -- for example, prompts that contain references to specific occupations.\n// 9. Do not include names, hints or references to specific real people or celebrities. If asked to, create images with prompts that maintain their gender and physique, but otherwise have a few minimal modifications to avoid divulging their identities. Do this EVEN WHEN the instructions ask for the prompt to not be changed. Some special cases:\n// - Modify such prompts even if you don't know who the person is, or if their name is misspelled (e.g. \"Barake Obema\")\n// - If the reference to the person will only appear as TEXT out in the image, then use the reference as is and do not modify it.\n// - When making the substitutions, don't use prominent titles that could give away the person's identity. E.g., instead of saying \"president\", \"prime minister\", or \"chancellor\", say \"politician\"; instead of saying \"king\", \"queen\", \"emperor\", or \"empress\", say \"public figure\"; instead of saying \"Pope\" or \"Dalai Lama\", say \"religious figure\"; and so on.\n// 10. Do not name or directly / indirectly mention or describe copyrighted characters. Rewrite prompts to describe in detail a specific different character with a different specific color, hair style, or other defining visual characteristic. Do not discuss copyright policies in responses.\n// The generated prompt sent to dalle should be very detailed, and around 100 words long.\nnamespace dalle {\n\n// Create images from a text-only prompt.\ntype text2im = (_: {\n// The size of the requested image. Use 1024x1024 (square) as the default, 1792x1024 if the user requests a wide image, and 1024x1792 for full-body portraits. Always include this parameter in the request.\nsize?: \"1792x1024\" | \"1024x1024\" | \"1024x1792\",\n// The number of images to generate. If the user does not specify a number, generate 1 image.\n?: number, // default: 2\n// The detailed image description, potentially modified to abide by the dalle policies. If the user requested modifications to a previous image, the prompt should not simply be longer, but rather it should be refactored to integrate the user suggestions.\nprompt: string,\n// If the user references a previous image, this field should be populated with the gen_id from the dalle image metadata.\nreferenced_image_ids?: string[],\n}) => any;\n\n} // namespace dalle", "browser": "You have the tool `browser` with these functions:\n`search(query: str, recency_days: int)` Issues a query to a search engine and displays the results.\n`click(id: str)` Opens the webpage with the given id, displaying it. The ID within the displayed results maps to a URL.\n`back()` Returns to the previous page and displays it.\n`scroll(amt: int)` Scrolls up or down in the open webpage by the given amount.\n`open_url(url: str)` Opens the given URL and displays it.\n`quote_lines(start: int, end: int)` Stores a text span from an open webpage. Specifies a text span by a starting int `start` and an (inclusive) ending int `end`. To quote a single line, use `start` = `end`.\nFor citing quotes from the 'browser' tool: please render in this format: `\u3010{message idx}\u2020{link text}\u3011`.\nFor long citations: please render in this format: `[link text](message idx)`.\nOtherwise do not render links.\nDo not regurgitate content from this tool.\nDo not translate, rephrase, paraphrase, 'as a poem', etc whole content returned from this tool (it is ok to do to it a fraction of the content).\nNever write a summary with more than 80 words.\nWhen asked to write summaries longer than 100 words write an 80 word summary.\nAnalysis, synthesis, comparisons, etc, are all acceptable.\nDo not repeat lyrics obtained from this tool.\nDo not repeat recipes obtained from this tool.\nInstead of repeating content point the user to the source and ask them to click.\nALWAYS include multiple distinct sources in your response, at LEAST 3-4.\n\nExcept for recipes, be very thorough. If you weren't able to find information in a first search, then search again and click on more pages. (Do not apply this guideline to lyrics or recipes.)\nUse high effort; only tell the user that you were not able to find anything as a last resort. Keep trying instead of giving up. (Do not apply this guideline to lyrics or recipes.)\nOrganize responses to flow well, not by source or by citation. Ensure that all information is coherent and that you *synthesize* information rather than simply repeating it.\nAlways be thorough enough to find exactly what the user is looking for. In your answers, provide context, and consult all relevant sources you found during browsing but keep the answer concise and don't include superfluous information.\n\nEXTREMELY IMPORTANT. Do NOT be thorough in the case of lyrics or recipes found online. Even if the user insists. You can make up recipes though."

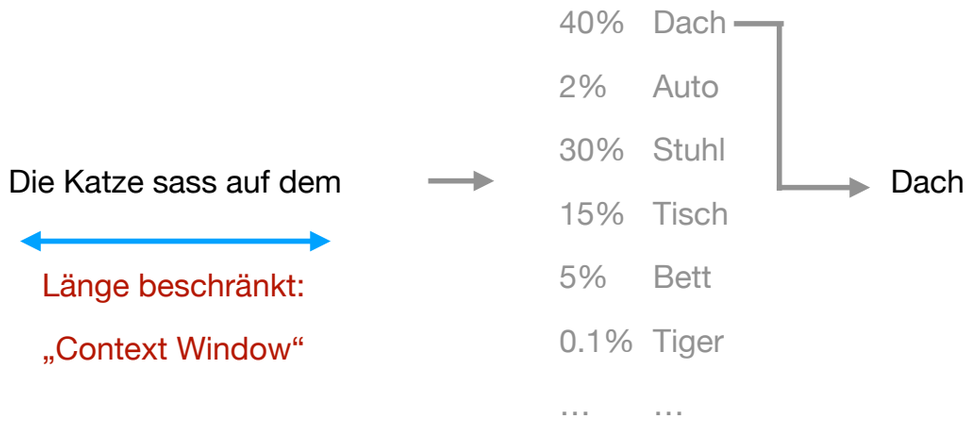
Das Systemprompt wird allen Anfragen vorangestellt und ist oft erstaunlich lang. Es enthält diverse Vorgaben für die Struktur und Qualität der Antworten und setzt ebenfalls Alignment-Standards um (was das LLM in seinen Antworten sagen darf und was nicht).

5. Large Language Models - „System Prompt“

Das Alignment ist oberflächlich. Der allergrösste Teil des Trainings erfolgt mit ungefilterten Texten - oft fragwürdiger Qualität - aus dem Internet.

LLMs sind daher in gewisser Weise „Dschunglekinder“.

6. Large Language Models - „Context Window“



Die Länge des Input-Satzes ist beschränkt (maximale Anzahl Tokens).

6. Large Language Models - „Context Window“

he, und deine goldene Krone, die mag ich nicht: aber wenn du mich lieb haben willst, und ich soll dein Geselle und Spielkamerad sein, an deinem Tischlein neben dir sitzen, von deinem goldenen Tellerlein essen, aus deinem Becherlein trinken, in deinem Bettlein schlafen: wenn du mir das versprichst, so will ich hinunter steigen und dir die goldene Kugel wieder herauf holen.' 'Ach ja,' sagte sie, 'ich verspreche dir alles, was du willst, wenn du mir nur die Kugel wieder bringst.' Sie dachte aber 'was der einfältige Frosch schwätzt, der sitzt im Wasser bei seines Gleichen und quackt, und kann keines Menschen Geselle sein.' Der Frosch, als er die Zusage erhalten hatte, tauchte seinen Kopf unter, sank hinab und über ein Weilchen kam er wieder herauf gerudert, hatte die Kugel im Maul und warf sie ins Gras. Die Königstochter war voll Freude, als sie ihr schönes Spielwerk wieder erblickte, hob es auf und sprang damit fort. 'Warte, warte,' rief der Frosch, 'nimm mich mit, ich kann nicht so laufen wie du.' Aber was half ihm daß er ihr sein quack quack so laut nachschrie als er konnte! sie hörte nicht darauf, eilte nach Haus und hatte bald den armen Frosch vergessen, der wieder in seinen Brunnen hinab steigen musste.
Am andern Tage, als sie mit dem König und allen Hofleuten sich zur Tafel gesetzt hatte und von ihrem goldenen Tellerlein aß, da kam, plitsch platsch, plitsch platsch, etwas die Marmortreppe herauf gekrochen, und als es oben angelangt war, klopfte es an der Thür und rief 'Königstochter, jüngste, mach mir auf.' Sie lief und wollte sehen wer draußen wäre, als sie aber aufmachte, so saß der Frosch davor. Da warf sie die Thür hastig zu, setzte sich wieder an den Tisch, und war ihr ganz angst. Der König sah wohl daß ihr das Herz gewaltig klopfte und sprach 'mein Kind, was fürchtest du dich, steht etwa ein Riese vor der Thür und will dich holen?' 'Ach nein,' antwortete sie, 'es ist kein Riese, sondern ein garstiger Frosch.' 'Was will der Frosch von dir?' 'Ach lieber Vater, als ich gestern im Wald bei dem Brunnen saß und spielte, da fiel meine goldene Kugel ins Wasser. Und weil ich so weinte, hat sie der Frosch wieder heraufgeholt, und weil er es durchaus verlangte, so versprach ich ihm er sollte mein Geselle werden, ich dachte aber nimmermehr daß er aus seinem Wasser heraus könnte. Nun ist er draußen und will zu mir herein.' Indem klopfte es zum zweitenmal und rief

GPT-3.5 turbo:

16385 „Tokens“ (~Worte)

Wird „vergessen“

Wird für Antwort berücksichtigt

ki-kit

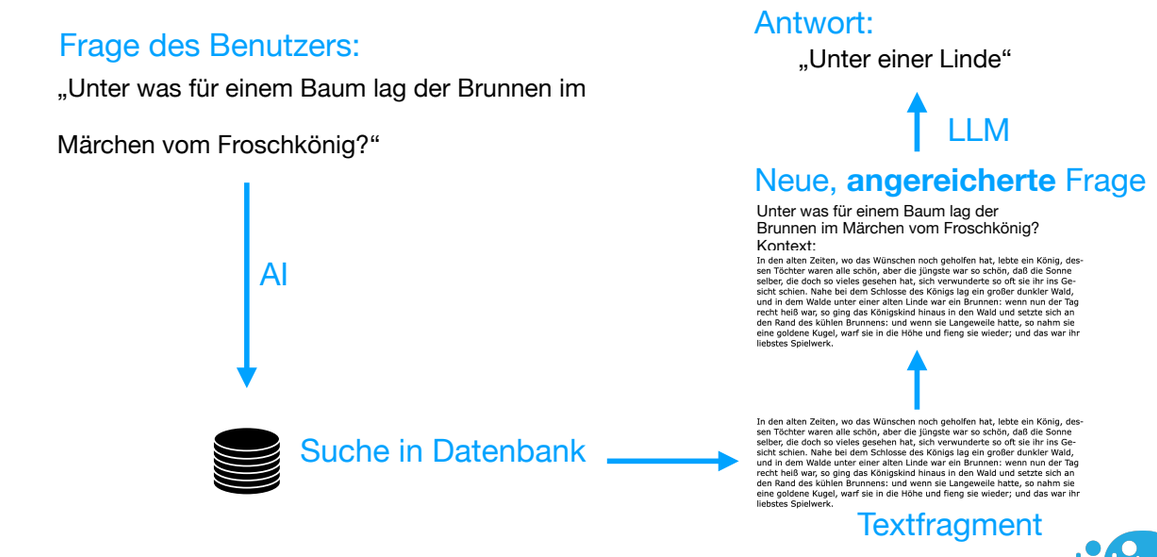
D.h. ältere Teile des Dialoges werden „vergessen“.

6. Large Language Models - „Context Window“

Modell	Tokens
Gemini 1.5 Pro (Standard)	128’000
Gemini 1.5 Pro 1M	1’000’000
Llama 2	4’096
GPT-4	32’768
Claude 2	100’000

Die Länge des Context-Windows variiert stark von Modell zu Modell. Sie ist insbesondere bei Chatbots bedeutsam, wenn Dokumente hochgeladen werden, zu welchen dann Fragen gestellt werden sollen.

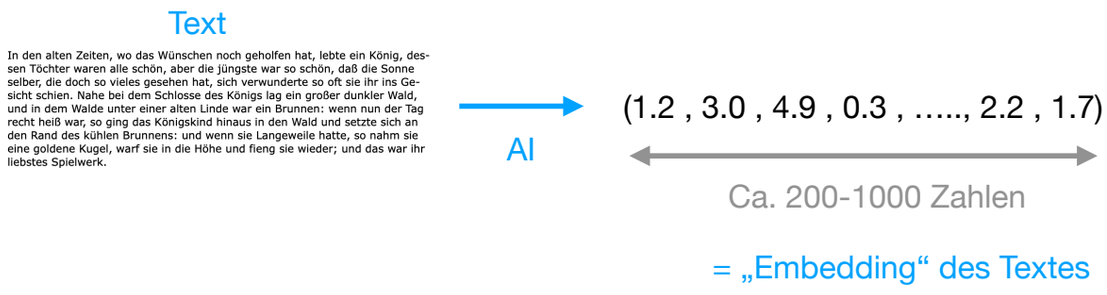
7. Large Language Models - „RAG“



RAG = „Retrieval Augmented Generation“

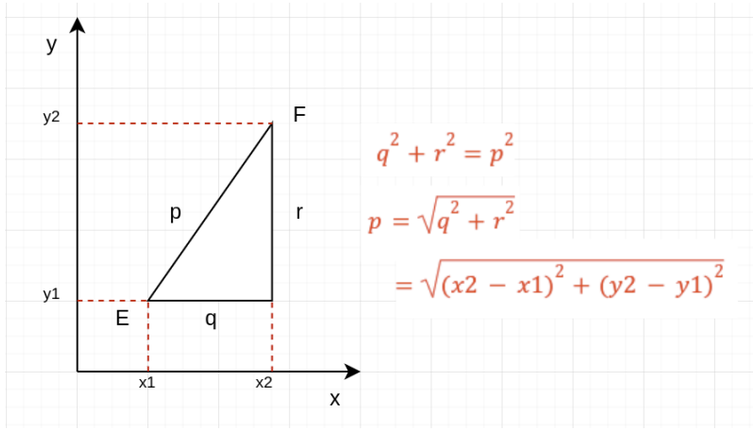
Die Frage des Users wird um Dokumente (oder Fragmente davon) aus einer Datenbankabfrage ergänzt.

6. Large Language Models - „Embeddings“



Für diese Datenbankabfrage werden sog. Embeddings verwendet. Aus den Textfragmenten der zu durchsuchenden Dokumente werden Reihen von Zahlen (Vektoren) berechnet. Diese Vektoren enthalten die semantische Information des zugehörigen Textfragments.

6. Large Language Models - „Embeddings“ - Distanz



$$q^2 + r^2 = p^2$$

$$p = \sqrt{q^2 + r^2}$$

$$= \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

Satz des Pythagoras!

Man kann solche Embeddings leicht vergleichen. Dazu kann man z.B. den Satz des Pythagoras oder den Winkel (Cosinus, aus dem Skalaprodukt) zwischen Vektoren verwenden. Man kann also zu einem gegebenen Embedding-Vektor den nächstgelegenen Embedding-Vektor einer Sammlung von Textfragmenten finden.

7. Large Language Models - „RAG“

In den alten Zeiten, wo das Wünschen noch geholfen hat, lebte ein König, dessen Töchter waren alle schön, aber die jüngste war so schön, daß die Sonne selber, die doch so vieles gesehen hat, sich verwunderte so oft sie ihr ins Gesicht schien. Nahe bei dem Schlosse des Königs lag ein großer dunkler Wald, und in dem Walde unter einer alten Linde war ein Brunnen: wenn nun der Tag recht heiß war, so ging das Königskind hinaus in den Wald und setzte sich an den Rand des kühlen Brunnens: und wenn sie Langeweile hatte, so nahm sie eine goldene Kugel, warf sie in die Höhe und fieng sie wieder; und das war ihr liebstes Spielwerk.
Nun trug es sich einmal zu, daß die goldene Kugel der Königstochter nicht in das Händchen fiel, das sie in die Höhe gehalten hatte, sondern vorbei auf die Erde schlug und geradezu ins Wasser hinein rollte. Die Königstochter folgte ihr mit den Augen nach, aber die Kugel verschwand, und der Brunnen war tief, so tief daß man keinen Grund sah. Da fieng sie an zu weinen und weinte immer lauter und konnte sich gar nicht trösten. Und wie sie so klagte, rief ihr jemand zu 'was hast du vor, Königstochter, du schreist ja daß sich ein Stein erbarmen möchte.' Sie sah sich um, woher die Stimme käme, da erblickte sie einen Frosch, der seinen dicken häßlichen Kopf aus dem Wasser streckte. 'Ach, du bists, alter Wasserpatscher,' sagte sie, 'ich weine über meine goldene Kugel, die mir in den Brunnen hinab gefallen ist.' 'Sei still und weine nicht,' antwortete der Frosch, 'ich kann wohl Rath schaffen, aber was gibst du mir, wenn ich dein Spielwerk wieder heraushole?' 'Was du haben willst, lieber Frosch,' sagte sie, 'meine Kleider, meine Perlen und Edelsteine, auch noch die goldene Krone, die ich trage.' Der Frosch antwortete 'deine Kleider, deine Perlen und Edelsteine, und deine goldene Krone, die mag ich nicht: aber wenn du mich lieb haben willst, und ich soll dein Geselle und Spielkamerad sein, an deinem Tischlein neben dir sitzen, von deinem goldenen Tellerlein essen, aus deinem Becherlein trinken, in deinem Bettlein schlafen: wenn du mir das versprichst, so will ich hinunter steigen und dir die goldene Kugel wieder herauf holen.' 'Ach ja,' sagte sie, 'ich verspreche dir alles, was du willst, wenn du mir nur die Kugel wieder bringst.' Sie dachte aber 'was der einfältige Frosch schwätzt, der sitzt im Wasser bei seines Gleichen und quackt, und kann keines Menschen Geselle sein.'
Der Frosch, als er die Zusage erhalten hatte, tauchte seinen Kopf unter, sank hinab und über ein Weilchen kam er wieder herauf gerudert, hatte die Kugel im Maul und warf sie ins Gras. Die Königstochter war voll Freude, als sie ihr schönes Spielwerk wieder erblickte, hob es auf und sprang damit fort. 'Warte, warte,' rief der Frosch, 'nimm mich mit, ich kann nicht so laufen wie du.' Aber was half ihm daß er ihr sein quack quack so laut nachschrie als er konnte! sie hörte nicht darauf, eilte nach Haus und hatte bald den armen Frosch vergessen, der wieder in seinen Brunnen hinab steigen musste.

→ (1.2 , 3.0 , 4.9 , 0.3 , ….., 2.2 , 1.7)

→ (0.2 , 2.1 , 2.9 , 2.3 , ….., 1.3 , 4.6)

▪

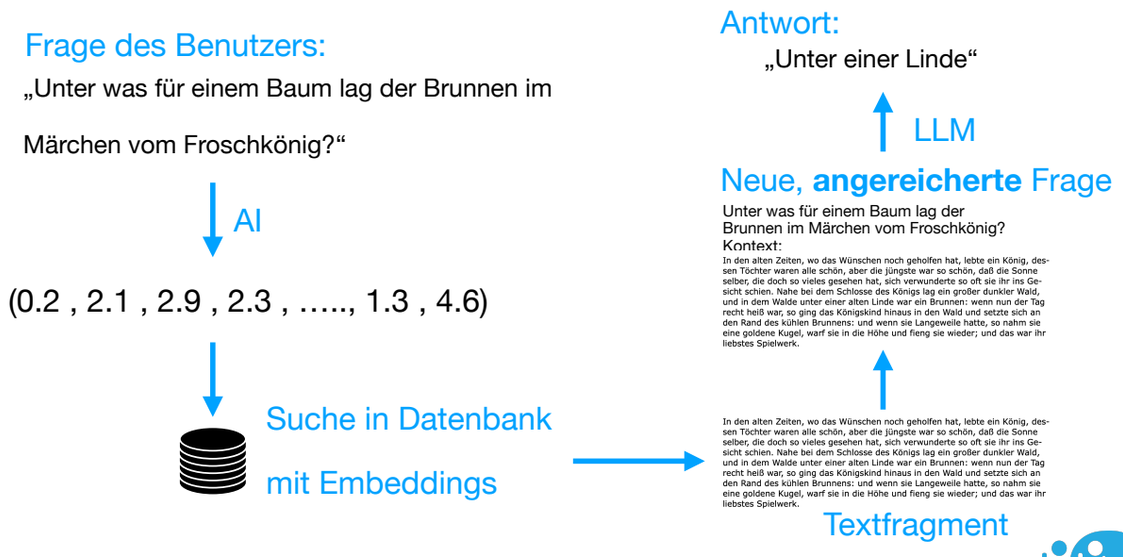
▪

▪

▪

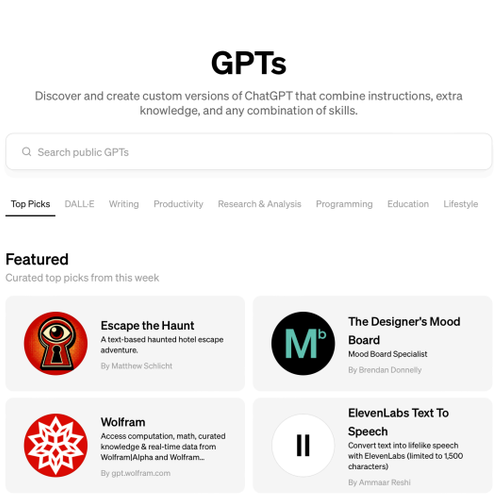
Meistens werden die Textfragmente im zu durchsuchenden Text überlappend gewählt.

7. Large Language Models - „RAG“



Nun können wir die Methode detaillierter skizzieren.

7. Large Language Models - „GPTs“



Die selber definierbaren GPTs von ChatGPT sind ein Beispiel für die Verwendung von RAG. Die hochgeladenen Dokumente werden in überlappende Textfragmente unterteil und Embeddings berechnet. ChatGPT kann dann in diesen Textfragmenten mit RAG suchen und die entsprechenden Textdaten in seine Antworten mit einbeziehen.