Die Tokenisierungs- und Vorhersageschritte in GPT-Modellen entmystifizieren

LLMs erhalten als Eingabe einen Prompt und generieren als Reaktion darauf einen Text. Dieser Prozess wird als *Textvervollständigung* bezeichnet. Der Prompt könnte beispielsweise lauten: »*The weather is nice today, so I decided to*«. Das Modell könnte dann vielleicht »*go for a walk*« ausgeben. Möglicherweise fragen Sie sich, wie das LLM-Modell diesen Ausgabetext aus dem Eingabe-Prompt erzeugt. Wie Sie sehen werden, ist das vor allem eine Frage der Wahrscheinlichkeiten.

Wird ein Prompt an ein LLM geschickt, teilt dieses die Eingabe zunächst in kleinere Elemente auf – in sogenannte *Tokens*. Diese Tokens stehen für einzelne Wörter, Wortteile oder Leer- und Satzzeichen. Der obige Prompt könnte beispielsweise unterteilt werden in `["The", "wea", "ther", "is", "nice", "today", ",", "so", "I", "de", "ci", "ded", "to"]`. Jedes Sprachmodell bringt seinen eigenen Tokenizer mit. Der Tokenizer von GPT-3.5 und GPT-4 steht online auf der OpenAI-Plattform (*https://oreil.ly/hbKT7*) zu Testzwecken zur Verfügung.

Als Faustregel kann man bei Tokens annehmen, dass 100 Tokens ungefähr 75 Wörtern in einem englischen Text entsprechen. Bei anderen Sprachen mag das anders aussehen, und die Anzahl an Tokens kann für die gleiche Zahl an Wörtern größer sein.

Dank des Attention-Prinzips und der Transformer-Architektur verarbeitet das LLM diese Tokens und kann die Beziehung zwischen ihnen sowie die Gesamtbedeutung des Prompts interpretieren. Die Transformer-Architektur erlaubt einem Modell, die entscheidenden Informationen und den Kontext im Text effizient zu erfassen.

Um einen neuen Satz zu erstellen, sagt das LLM die Tokens voraus, die basierend auf dem vom User angegebenen Eingabekontext des Prompts am wahrscheinlichsten folgen werden. OpenAI bietet viele Versionen von GPT-4 an. Zuerst hatten Sie nur die Wahl zwischen einem Eingabekontextfenster mit maximal 8.192 oder 32.768 Tokens. Anfang 2024 wurden die neuesten Modelle GPT-4 Turbo und GPT-4o von OpenAI veröffentlicht, die ein größeres Eingabekontextfenster von 128.000 Tokens aufweisen – das entspricht fast 300 Seiten englischen Texts. Anders als die vorherigen rekurrenten Modelle, die Probleme mit langen Eingabesequenzen hatten, kann das moderne LLM durch die

Transformer-Architektur und den Attention-Mechanismus den Kontext im Ganzen berücksichtigen. Abhängig von diesem Kontext weist das Modell jedem potenziell folgenden Token einen Wahrscheinlichkeitswert zu. Das Token mit der höchsten Wahrscheinlichkeit wird dann als Nächstes in der Sequenz gewählt. In unserem Beispiel könnte auf »The weather is nice today, so I decided to« als nächstes bestes Token »go« folgen.

Wie Sie im nächsten Kapitel sehen werden, ist es mithilfe eines *Temperatur*-Parameters möglich, nicht einfach das nächste Token mit der höchsten Wahrscheinlichkeit zu nutzen, sondern es dem Modell zu erlauben, das nächste Token aus einer Reihe von Tokens mit der höchsten Wahrscheinlichkeit auszuwählen. Das sorgt für mehr Variabilität und Kreativität in der Antwort des Modells.

Dieser Prozess wird anschließend wiederholt, aber nun wird der Kontext zu »The weather is nice today, so I decided to go« – also mit dem zuvor vorhergesagten Token »go« am Ende des ursprünglichen Prompts. Das zweite Token, das das Modell möglicherweise vorhersagt, könnte »for« sein. Dieser Prozess wird so lange wiederholt, bis ein vollständiger Satz geformt ist: »The weather is nice today, so I decided to go for a walk«. Der Prozess baut auf der Fähigkeit des LLM auf, das wahrscheinlichste nächste Wort aus den riesigen Textdaten gelernt zu haben. Abbildung 1-5 zeigt diesen Prozess.

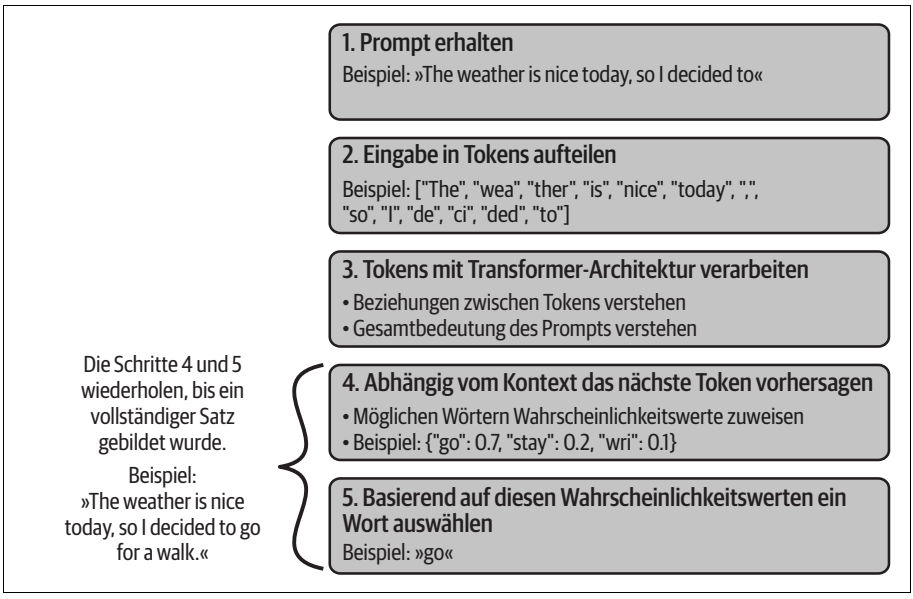


Abbildung 1-5: Der Vervollständigungsprozess läuft iterativ ab – Token für Token.

Einbinden von Vision in ein LLM

Mit GPT-4 Vision wird die GPT-4-Serie um multimodale Fertigkeiten ergänzt, sodass sie jetzt auch mit mehr als nur mit Text umgehen kann. Die spezifischen Mechanismen sorgen dafür, dass dieses Feature proprietär bleibt und nicht so einfach eingesehen werden kann. Es lassen sich aber Schlüsse aus Open-Source-LLMs ziehen, die visuelle Daten integrieren, sodass wir zumindest ein grundlegendes Verständnis der potenziellen Methoden erlangen können, mit denen GPT-4 um solch eine multimodale Funktionalität erweitert wird. In diesem Abschnitt sollen die Prozesse vorgestellt werden, die sich in diesen Open-Source-Gegenstücken beobachten lassen, damit Sie eine Idee davon bekommen, wie die Bild-Text-Integration in GPT-4 möglicherweise umgesetzt ist.

Convolutional Neural Networks (CNNs) waren lange Zeit State-of-the-Art bei Aufgaben zur Bildverarbeitung. Sie sind sehr gut beim Klassifizieren von Bildern und Erkennen von Objekten. Das erreichen sie, indem sie Filterschichten nutzen, die über ein Eingabebild geschoben werden. Diese Filter können die räumlichen Beziehungen zwischen den Pixeln des Bilds aufdecken, und sie helfen den CNNs dabei, Muster zu erkennen – von einfachen Kanten in den ersten Schichten bis hin zu komplexen Formen und Objekten in den tieferen Schichten.

Aber ähnlich wie die Einführung der Transformer-Architekturen im Jahr 2017 NLP revolutionierte und RNNs ablöste, wurden im Jahr 2020 neue Modelle für die Bildverarbeitung vorgeschlagen, die auf Transformer-Architekturen basieren. Seitdem wird die seit Langem bestehende Dominanz von CNNs in der Bildverarbeitung infrage gestellt. Im Jahr 2021 zeigte ein Artikel in Google mit dem Titel »An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale« von Dosovitskiy et al. (*https://oreil.ly/ijPSk*), dass ein reines Transformer-Modell namens *Vision Transformer* (ViT) für viele Bildklassifikationsaufgaben eine bessere Leistung liefern kann.

Sie fragen sich vielleicht, wie der Transformer die Bilddaten verarbeitet. Aus der Ferne betrachtet, ist das Vorgehen dem bei Text ziemlich ähnlich. Wird ein Text-Prompt an ein LLM geschickt, teilt es den Text erst in kleinere Zeichenabschnitte auf – die Tokens – und verarbeitet diese dann, um das nächste Token vorherzusagen. Bei Bildern teilt der ViT das Bild erst in kleine, gleich große Kacheln. In Abbildung 1-6 sehen Sie ein Beispiel dafür.

Diese Bildkacheln werden dann mit den Text-Tokens in einer einheitlichen Eingabesequenz zusammengeführt. Ohne nun allzu sehr in die technischen Details zu gehen: Wenn ein LLM Textdaten verarbeitet, wird jedes Token zunächst in einen hochdimensionalen *Vektor* umgewandelt. Diese Mapping-

Funktion zwischen den Tokens und den hochdimensionalen Vektoren wird während des Lernprozesses des LLM berechnet. Eine Mapping-Funktion zwischen den Kacheln und dem gleichen hochdimensionalen Raum wird ebenfalls während des Lernprozesses ermittelt. So finden sich danach sowohl die Tokens wie auch die Kacheln im gleichen hochdimensionalen Raum. Die kombinierte Sequenz aus Text und Bild kann dann von der Transformer-Architektur verarbeitet werden, um das nächste Token vorherzusagen. Durch die Kombination aus Bildkacheln und Text-Tokens im gleichen hochdimensionalen Darstellungsraum kann das Modell Self-Attention-Mechanismen für beide Modalitäten nutzen und Antworten erzeugen, die Text- und Bildinformationen berücksichtigen. Bei der Python-Entwicklung kann die Fähigkeit, Bilder zu verarbeiten, die Interaktion der User mit Ihrer KI-Anwendung deutlich besser machen, zum Beispiel über intuitivere Chatbots oder Lerntools, die Inhalte aus Bildern verstehen und erläutern können.

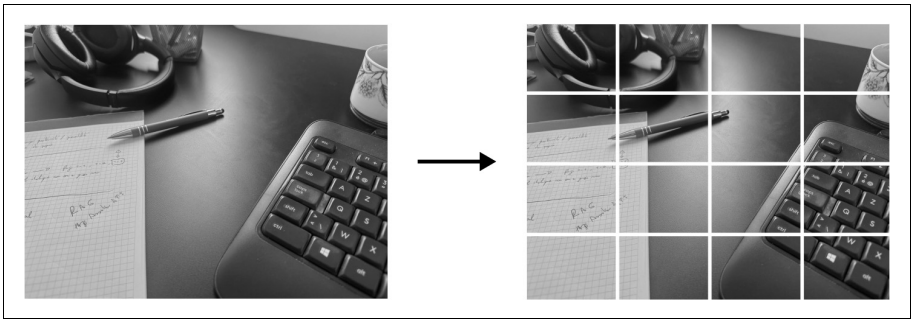


Abbildung 1-6: Ein Bild wird in gleich große Kacheln unterteilt, bevor es an den Transformer übergeben wird.

Eine kurze Geschichte des GPT: von GPT-1 bis GPT-4

In diesem Abschnitt werden wir die Entwicklung der GPT-Modelle von OpenAI von GPT-1 bis GPT-4 betrachten.

GPT-1

Mitte 2018 – nur ein Jahr nach der Erfindung der Transformer-Architektur – hat OpenAI den Artikel »Improving Language Understanding by Generative Pre-Training« (*https://oreil.ly/Yakwa*) von Radford et al. veröffentlicht, in dem die Firma den *Generative Pre-Trained Transformer* vorstellt – auch bekannt als GPT-1.

Vor GPT-1 war das Standardvorgehen zum Aufbauen hoch performanter neuronaler NLP-Modelle das Supervised Learning (überwachtes Lernen). Diese Lerntechnik nutzt große Mengen manuell gelabelter Daten. Bei solch einer Sentimentanalyse, deren Ziel beispielsweise darin besteht, zu klassifizieren, ob ein bestimmter Text eine positive oder negative Stimmung aufweist, würde man üblicherweise die Strategie wählen, Tausende manuell gelabelter Textbeispiele zu sammeln, um ein effektives Klassifikationsmodell zu bauen. Aber die Notwendigkeit großer Mengen gut ausgezeichneter und überwachter Daten hat die Leistungsfähigkeit dieser Techniken beschränkt, weil solche Datensätze nur schwierig und mit hohen Kosten zu erzeugen sind.

In ihrem Artikel haben die Autoren von GPT-1 einen neuen Lernprozess vorgeschlagen, bei dem ein unüberwachter Pretraining-Schritt eingeführt wird. In diesem Schritt sind keine gelabelten Daten erforderlich. Stattdessen wird das Modell darauf trainiert, vorherzusagen, was das nächste Token sein wird. Dank des Einsatzes der Transformer-Architektur, die eine Parallelisierung erlaubt, wurde dieses Pretraining mit einer großen Menge an Daten durchgeführt. Für das Pretraining hat das GPT-1-Modell den *BookCorpus*-Datensatz genutzt, der den Text von ungefähr 11.000 unveröffentlichten Büchern enthält. Dieser Datensatz wurde im Jahr 2015 in dem wissenschaftlichen Artikel »Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books« (*https://oreil.ly/3hWl1*) von Zhu et al. vorgestellt und ursprünglich auf einer Webseite der University of Toronto verfügbar gemacht. Die offizielle Version des ursprünglichen Datensatzes ist allerdings mittlerweile nicht mehr öffentlich zugänglich.

Das GPT-1-Modell zeigte sich bei einer Reihe einfacher Vervollständigungsaufgaben sehr effektiv. In der Phase des Unsupervised Learning (unüberwachtes Lernen) hatte das Modell gelernt, das nächste Element in den Texten des BookCorpus-Datensatzes vorherzusagen. Aber da GPT-1 ein kleines Modell ist, konnte es ohne Optimierungen keine komplexen Aufgaben erledigen. Daher wurde die Optimierung in einem zweiten überwachten Lernschritt mit einer kleinen Menge manuell gelabelter Daten durchgeführt, um das Modell an eine spezifische Aufgabe anzupassen. So kann es beispielsweise bei einer Klassifikationsaufgabe (wie einer Sentimentanalyse) erforderlich sein, das Modell mit einem kleinen Satz manuell gelabelter Textbeispiele neu zu trainieren, um eine vernünftige Genauigkeit zu erhalten. Dieser Prozess erlaubte es, die in der ersten Pretraining-Phase gelernten Parameter so anzupassen, dass sie besser zur aktuellen Aufgabe passten.

Trotz der recht kleinen Größe zeigte GPT-1 bei vielen NLP-Aufgaben erstaunliche Leistungen, wobei für die Optimierung nur wenige manuell gelabelte Daten erforderlich waren. Die GPT-1-Architektur bestand aus einem Decoder, der dem des ursprünglichen 2017 vorgestellten Transformers mit 117 Millionen Parametern ähnelte. Dieses erste GPT-Modell war die Grundlage für leistungsfähigere Modelle mit größeren Datensätzen und mehr Parametern, mit denen die Vorteile des Potenzials der Transformer-Architektur ausgeschöpft werden konnten.

GPT-2

Anfang 2019 schlug OpenAI GPT-2 vor – eine größere Version des GPT-1-Modells, bei der die Anzahl der Parameter und die Größe des Trainingsdatensatzes verzehnfacht wurden. Es gab nun 1,5 Milliarden Parameter, die mit 40 GB Text trainiert wurden. Im November 2019 veröffentlichte OpenAI die vollständige Version des GPT-2-Sprachmodells.

GPT-2 steht öffentlich zur Verfügung und lässt sich bei Hugging Face (*https://oreil.ly/Q7KfI*) oder GitHub (*https://oreil.ly/mkNT6*) herunterladen.

GPT-2 hat gezeigt, dass durch das Trainieren eines größeren Sprachmodells mit einem größeren Datensatz die Fähigkeiten des Modells verbessert werden können. Bei vielen Aufgaben war es nun besser als die bisherigen Spitzenreiter. Zudem ließ sich zeigen, dass noch größere Sprachmodelle auch natürliche Sprache besser verarbeiten können.

GPT-3

OpenAI veröffentlichte Version 3 von GPT im Juni 2020. Die Hauptunterschiede zu GPT-2 bestehen in der Größe des Modells und der Menge an Daten, die beim Training zum Einsatz kamen. GPT-3 ist ein viel größeres Modell als GPT-2 – es hat 175 Milliarden Parameter, wodurch komplexere Muster erfasst werden können. Zudem wurde GPT-3 mit einem größeren Datensatz trainiert. Dazu gehörte auch Common Crawl (*https://oreil.ly/GOAR0*), ein großes Webarchiv mit Texten aus Milliarden von Webseiten und anderen Quellen, zum Beispiel Wikipedia. Dieser Trainingsdatensatz, der Inhalte von Webseiten, Büchern

und Artikeln enthält, erlaubte GPT-3, ein tieferes Verständnis der Sprache und des Kontexts zu entwickeln. Im Ergebnis besaß GPT-3 eine verbesserte Leistung bei vielen linguistischen Aufgaben. Darüber hinaus zeigte es bei seinen generierten Texten eine sehr gute Kohärenz und viel Kreativität. Es konnte sogar Codeschnipsel schreiben, wie zum Beispiel SQL-Abfragen, und andere intelligente Ausgaben erledigen. Zudem war bei GPT-3 kein Optimierungsschritt mehr erforderlich, was bei den Vorgängern noch notwendig war.

Von GPT-3 zu InstructGPT

Bei GPT-3 gab es aber eine Diskrepanz zwischen den von den Endanwendern und -anwenderinnen gegebenen Aufgaben und dem, was das Modell während des Trainings zu Gesicht bekommen hatte. Wie wir gesehen haben, werden Sprachmodelle darauf trainiert, abhängig vom Eingabekontext das nächste Token vorherzusagen. Dieser Trainingsprozess ist nicht notwendigerweise direkt mit den Aufgaben verbunden, die das Modell am Ende gestellt bekommt. Zudem sorgt ein Vergrößern des Sprachmodells nicht zwingend dafür, dass es besser darin wird, die Wünsche oder Anweisungen der User zu erfüllen. Und Modelle wie GPT-3 werden mit Daten aus unterschiedlichen Quellen im Internet trainiert. Bei der Auswahl der Quellen wird zwar auf Qualität geachtet, trotzdem können die Lerndaten falsche oder problematische Texte enthalten, zum Beispiel rassistische oder sexistische Texte, Fehlinformationen oder Desinformation. Daher sagt das Modell manchmal etwas Falsches oder liefert toxische Texte. 2021 wurde ein neues Release des GPT-3-Modells veröffentlicht – die *Instruct Series*. Anders als das ursprüngliche GPT-3-Basismodell sind die Instruct-Modelle durch *Reinforcement Learning* mit menschlichem Feedback optimiert – sie nutzen also das Feedback, um mit der Zeit zu lernen und besser zu werden. Das erlaubt den Modellen, von menschlichen Anweisungen zu lernen und dabei wahrhaftiger und weniger toxisch zu sein.

Um den Unterschied deutlich zu machen, übergeben wir als Prompt »Explain what is meant by time complexity« und erhalten das Folgende:

- Beim Standard-GPT-3 bekommen wir die Ausgabe: »Explain what is meant by space complexity. Explain what is meant by the big-O notation.«
- Mit dem InstructGPT-3-Modell bekommen wir: »Time complexity is a way of measuring the amount of time it takes for an algorithm to run and complete its task. It is usually expressed using Big O notation, which measures the complexity of an algorithm in terms of the number of operations it performs. The time complexity of an algorithm is important because it determines how efficient the algorithm is and how well it scales with larger inputs.«

Wir sehen, dass bei gleicher Eingabe das erste Modell die Frage nicht beantworten kann (die Antwort ist zudem etwas bizarr), während das zweite Modell eine Antwort liefert. Es ist natürlich möglich, die gewünschte Antwort auch mit einem Standard-GPT-3-Modell zu erhalten. Aber im Gegensatz zu Instruct-Modellen ist es notwendig, ein spezifisches Prompt-Design und Optimierungstechniken zu verwenden. Diese Technik wird als *Prompt Engineering* bezeichnet und in den folgenden Kapiteln genauer behandelt.

OpenAI beschreibt in dem wissenschaftlichen Artikel »Training Language Models to Follow Instructions with Human Feedback« (*https://oreil.ly/sz90A*) von Ouyang et al., wie die Instruct Series aufgebaut wurden.

Das Training besteht aus zwei großen Stufen, um von einem GPT-3- zu einem Instruct-GPT-3-Modell zu gelangen: *Supervised Fine-Tuning* (*SFT*) und *Reinforcement Learning from Human Feedback* (*RLHF*). In jeder Stufe werden die Ergebnisse der vorherigen Stufe optimiert. Die SFT-Stufe bekommt also das GPT-3-Modell übergeben und gibt ein neues Modell zurück, das dann an die RLHF-Stufe geschickt wird, um die Instruct-Version zu erhalten.

Abbildung 1-7 (aus dem OpenAI-Artikel) demonstriert den ganzen Prozess etwas detaillierter.

Wir werden diese Schritte nacheinander durchgehen.

Im SFT-Schritt wird das Original-GPT-3-Modell mit einem klassischen Supervised Learning optimiert (Schritt 1 in Abbildung 1-1). OpenAI besitzt eine Sammlung von Prompts von Endanwenderinnen und -anwendern. Der Prozess beginnt mit der zufälligen Auswahl eines Prompts aus der Menge der verfügbaren Prompts. Eine Person (als *Labeler* bezeichnet) wird dann gebeten, ein Beispiel für eine ideale Antwort für diesen Prompt einzugeben. Dieser Prozess wird Tausende Male wiederholt, um einen überwachten Trainingsdatensatz aus Prompts und den dazugehörigen idealen Antworten zu erhalten. Mit diesem Datensatz wird dann das GPT-3-Modell optimiert, damit es auf Benutzeranfragen konsistentere Antworten geben kann. Das sich daraus ergebende Modell wird als SFT-Modell bezeichnet.

Der RLHF-Schritt ist in zwei Unterschritte aufgeteilt. Zuerst wird ein Reward-Modell (RM) erstellt (Schritt 2 in Abbildung 1-7), dann kommt dieses RM beim Reinforcement Learning zum Einsatz (Schritt 3 in Abbildung 1-7).

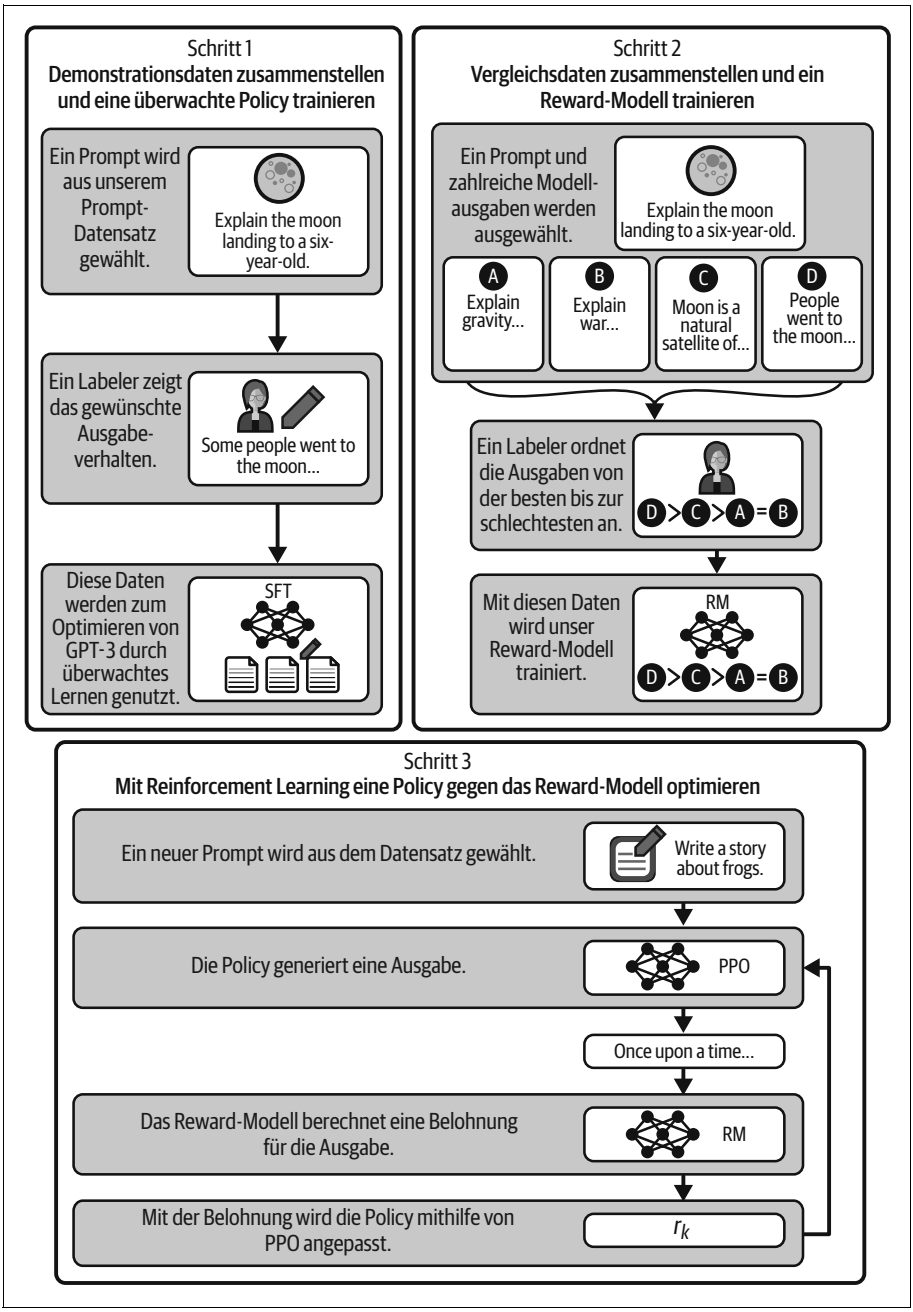


Abbildung 1-7: Die Schritte zum Erhalten des Instruct-Modells (nach einem Bild von Ouyang et al.)

Das Ziel des RM besteht darin, automatisch eine Bewertung für eine Antwort auf einen Prompt zu geben. Passt die Antwort zu dem, was im Prompt gewünscht war, sollte die RM-Bewertung hoch sein – passt sie nicht, sollte sie niedrig sein. Um das RM zu erstellen, beginnt OpenAI damit, zufällig eine Frage auszuwählen und dann mit dem SFT-Modell diverse mögliche Antworten zu generieren. Wie Sie später sehen werden, ist es möglich, für den gleichen Eingabe-Prompt viele Antworten zu erzeugen, indem ein Parameter namens *Temperatur* zum Einsatz kommt. Ein menschlicher Labeler wird dann gebeten, die Antworten in einer Reihenfolge anzuordnen, wobei Kriterien wie eine Übereinstimmung mit dem Prompt und die Toxizität der Antwort zum Einsatz kommen. Nachdem diese Prozedur viele Male durchgeführt wurde, dient ein Datensatz dem Optimieren des SFT-Modells für das Scoring. Dieses RM wird dann genutzt, um das finale InstructGPT-Modell zu bauen.

Der letzte Schritt besteht im Trainieren von InstructGPT-Modellen per Reinforcement Learning in einem iterativen Prozess. Er beginnt mit einem initialen generativen Modell – wie zum Beispiel dem SFT-Modell. Dann wird zufällig ein Prompt ausgewählt, und das Modell sagt eine Ausgabe vorher, die das RM nun bewertet. Abhängig von der erhaltenen Belohnung wird das generative Modell entsprechend angepasst. Dieser Prozess kann unzählige Male ohne menschliche Beteiligung wiederholt werden – ein effizienterer und automatisierter Ansatz, um das Modell zugunsten einer besseren Leistung anzupassen.

InstructGPT-Modelle sind besser darin, korrekte Vervollständigungen für die Eingabe-Prompts zu erzeugen. OpenAI empfiehlt, statt der ursprünglichen Serie die InstructGPT-Serie einzusetzen.

GPT-3.5, ChatGPT und Codex

Im März 2022 veröffentlichte OpenAI eine neue Version von GPT-3. Diese neuen Modelle können Text bearbeiten und Inhalte in Text einfügen. Sie waren bis Juni 2021 mit Daten trainiert worden und sollten leistungsfähiger als frühere Versionen sein. Seit Ende November 2022 bezeichnet OpenAI diese Modelle als zugehörig zur GPT-3.5-Serie.

Im November 2022 stellte OpenAI ChatGPT (*https://oreil.ly/2Qv91*) als experimentelles Gesprächstool vor. Das Modell hinter diesem Tool war eine optimierte Version von GPT-3.5 namens GPT-3.5 Turbo. Es war besonders gut in interaktiven Dialogen, wobei eine Technik zum Einsatz kam, die der in Abbildung 1-7 ähnelt.

Als ChatGPT erstmals veröffentlicht wurde, nutzte man den Namen *ChatGPT* sowohl für das Modell als auch für die Weboberfläche des Chatbots, und daher konnte man ChatGPT sowohl für das Modell als auch für die Oberfläche einsetzen.

Im Rest dieses Buchs werden wir folgende Unterscheidung treffen:

- *GPT-3.5* und *GPT-4* beziehen sich auf die beiden Familien der Large Language Models von OpenAI, wobei jede Familie eine Reihe von unterschiedlichen Versionen des Modells bietet.
- *ChatGPT* bezieht sich auf die Chat-Weboberfläche, die diese Modelle nutzt. Standardmäßig läuft es mit einem GPT-3.5-Modell.

OpenAI schlug ebenfalls den Einsatz des Codex-Modells vor. Dabei handelt es sich um ein GPT-3-Modell, das mit Milliarden von Codezeilen optimiert wurde und von GitHub Copilot (*https://oreil.ly/GhOer*) eingesetzt wird – eine Autovervollständigung beim Programmieren, die beim Entwickeln hilft und von vielen Texteditoren wie zum Beispiel Visual Studio Code, Jetbrains und sogar Neovim unterstützt wird. Allerdings wurde das Codex-Modell im März 2023 von OpenAI als deprecated abgekündigt. Stattdessen wird empfohlen, von Codex zu GPT-3.5 Turbo oder GPT-4 zu wechseln. Gleichzeitig hat GitHub Copilot X veröffentlicht, das auf GPT-4 basiert und viel mehr Funktionalität als die vorherige Version zu bieten hat.

Das Abkündigen des Codex-Modells durch OpenAI ist ein gutes Beispiel für die inhärenten Risiken beim Arbeiten mit APIs: Sie können sich jederzeit ändern oder plötzlich nicht mehr funktionieren, wenn neuere, effizientere Modelle entwickelt und ausgerollt werden.

GPT-4

Im März 2023 stellte OpenAI GPT-4 bereit. Wir wissen sehr wenig über die Architektur dieser neuen Art von Modellen, da OpenAI kaum Informationen dazu herausrückt. Es ist das aktuell ausgefeilteste System von OpenAI, und es sollte sicherere und nützlichere Antworten generieren. Die Firma behauptet, dass GPT-4 bezüglich seiner Möglichkeiten zum Schlussfolgern besser ist als GPT-3.5 Turbo.

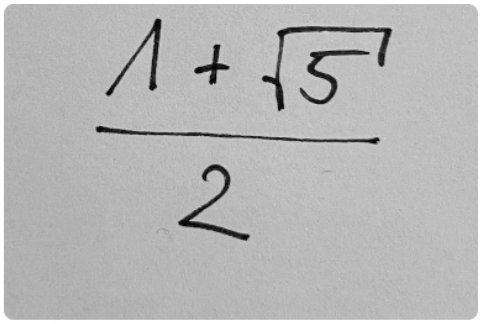
Als das Modell veröffentlicht wurde, stellte OpenAI einen technischen Bericht bereit (*https://oreil.ly/a3uj1*), der die Fähigkeiten des Modells vorstellt und viele Vergleiche mit früheren OpenAI-Modellen wie InstructGPT und GPT-3 liefert.

Anders als andere Modelle der OpenAI-GPT-Familie ist GPT-4 das erste multimodale Modell, das nicht nur Text, sondern auch Bilder verarbeiten kann. GPT-4 berücksichtigt also im vom Modell genutzten Kontext Bilder und Text, um eine Ausgabesequenz zu erzeugen, wodurch es möglich ist, einen Prompt um ein Bild zu ergänzen und Fragen dazu zu stellen.

OpenAI hat dieses Feature in GPT-4 zunächst nicht öffentlich verfügbar gemacht. Erst im November 2023 kündigte das Unternehmen das Modell GPT-4 Turbo mit Vision-Fähigkeiten an. Dieses neue Modell brachte außerdem ein Kontextfenster von 128.000 Tokens mit – eine einzelne Eingabe kann dann das Äquivalent von 300 Seiten englischem Text enthalten! Zudem ist dieses GPT-4-Turbo-Modell auch noch günstiger als das ursprüngliche GPT-4.

Im Beispiel in Abbildung 1-8 haben wir eine Gleichung auf ein Stück Papier geschrieben, es abfotografiert und GPT-4 Turbo gebeten, die Gleichung im Bild zu beschreiben. Wie Sie sehen, hat das Modell problemlos erkannt, dass es sich um den Goldenen Schnitt handelt.

Es gibt mittlerweile viele Modelle auf dem Markt, und es wird immer wichtiger, sie objektiv miteinander vergleichen zu können, um herauszufinden, welche bei welchen Aufgaben besser funktionieren. Eine einfache Möglichkeit ist, ihre Ergebnisse bei Universitätstests zu vergleichen. In diesem Rahmen haben die Modelle eine Reihe von Prüfungen absolviert, und GPT-4 hat GPT-3.5 Turbo hinter sich gelassen, indem es jeweils höhere Perzentile erreichte. So hat GPT-3.5 Turbo beispielsweise beim Uniform Bar Exam (*https://oreil.ly/opXec*) den Bereich der 10. Perzentile erreicht, während GPT-4 den der 90. Perzentile geschafft hat. Das Gleiche gilt für die International Biology Olympiad (*https://oreil.ly/a8CP6*), bei der GPT-3.5 Turbo die 31. Perzentile erreicht hat, während GPT-4 die 99. Perzentile schaffte. Dieser Fortschritt ist sehr beeindruckend, insbesondere wenn man berücksichtigt, dass er innerhalb von einem Jahr vonstattenging. Kürzlich hat OpenAI GPT-4o veröffentlicht (»o« für *omni*) – sein neuestes Aushängeschild, das das bisherige GPT-4-Modell in vielen Benchmarks zu schlagen scheint.



Was kannst du über diese Gleichung sagen?

Diese Gleichung ist eine bekannte mathematische Ausdruck, der oft im Zusammenhang mit dem goldenen Schnitt erwähnt wird. Der Ausdruck $\frac{1+\sqrt{5}}{2}$ ist tatsächlich der Wert des goldenen Schnitts ϕ, der etwa 1,6180339887 beträgt.

Der goldene Schnitt ist eine irrationale Zahl, die in vielen Bereichen der Mathematik, Kunst und Architektur vorkommt. Es handelt sich um eine spezielle Zahl, die eine wichtige Rolle in der Geometrie spielt, insbesondere bei der Teilung einer Strecke in zwei Teile, sodass das Verhältnis des Ganzen zu dem größeren Teil dem Verhältnis des größeren Teils zum kleineren Teil entspricht.

Kurz gesagt, dieser Ausdruck stellt den goldenen Schnitt dar.

Abbildung 1-8: Die visuellen Fähigkeiten von GPT-4 (August 2024)

Sprachmodelle werden auch gern verglichen, indem Menschen gebeten werden, Interaktionen mit unterschiedlichen Modellen blind zu bewerten, sodass sie nicht wissen können, mit welchem Modell sie gerade reden. Das LMSYS Chatbot Arena Leaderboard (*https://oreil.ly/p11EX*), gehostet von Hugging Face, bietet solche Vergleiche. Dabei handelt es sich um eine crowdgesourcte, randomisierte Battle-Plattform für LLMs. Hier können User gleichzeitig mit zwei zufällig gewählten Modellen kommunizieren, ohne dass sie wissen, welche das sind, und dann angeben, welche der zwei Antworten sie relevanter fanden. Das ist wie ein kompetitives Spiel mit Wettbewerben und einem ELO-Score, der zum Bewerten dieser Modelle genutzt wird (siehe den nachfolgenden Kasten).

Warum wird der ELO genutzt, um die Modelle zu vergleichen?

Das ELO-Bewertungssystem wurde von Arpad Elo entwickelt, einem in Ungarn geborenen amerikanischen Physikprofessor und Schachspieler. Er schlug dieses System als Verbesserung gegenüber einem älteren System vor, das von der United States Chess Federation (USCF) genutzt wurde. Die USCF übernahm Elos System 1960, die World Chess Federation im Jahr 1970. Es wird mittlerweile auch dazu eingesetzt, Spielerinnen und Spieler in anderen Wettbewerbssituationen zu vergleichen, zum Beispiel bei Videospielen, bei denen es zum Klassifizieren von Spielern und Spielerinnen von League of Legends zum Einsatz kommt.

Inzwischen dient das ELO-System ebenfalls dem Vergleichen von LLMs. Zwei LLMs treten in einem Blindvergleich gegeneinander an, in dem ein Mensch diesen beiden LLMs Fragen stellt (die beiden Modelle erhalten die gleichen Eingabenachrichten). Dann muss der User sagen, welche der beiden Antworten die bessere ist. Das LMSYS Chatbot Arena Leaderboard (*https://oreil.ly/p11EX*), das von Hugging Face gehostet wird, erlaubt einen einfachen Vergleich zwischen Chatbots.

Mit dem ELO-System können Spielerinnen und Spieler in kompetitiven Nullsummenspielen mit Wettbewerben bewertet werden. *Nullsummenspiele* sind solche, in denen der Gewinn der einen Seite dem Verlust der anderen Seite entspricht. Die Schwierigkeit beim Bewerten von Wettbewerben liegt in der dynamischen Natur der Konfrontationen zwischen Spielenden und dem fortlaufenden Hinzukommen neuer Mitbewerber. Dieses Bewertungssystem ist so entworfen, dass es flexibel ist – es passt die Einstufung der Spielerin oder des Spielers so an, dass es das Ergebnis jedes Spiels berücksichtigt und eine Möglichkeit hat, die relativen Fähigkeiten der Spielenden zu bewerten.

Die ELO-Bewertung weist jedem Spieler und jeder Spielerin eine Zahl zu – ein höherer Wert steht beim Vergleichen für mehr Fertigkeiten. Eines der zentralen Features des ELO-Bewertungssystems ist, dass es einen direkten Weg zum Ermitteln der Wahrscheinlichkeit bietet, mit der eine Spielerin einen anderen schlagen wird – nur basierend auf der Differenz der beiden ELO-Werte.

Angenommen, *Ei* und *Ej* sind die ELO-Zahlen der beiden Spielenden *i* und *j*, dann ist die Wahrscheinlichkeit, dass Spieler *i* ein Spiel gewinnen wird, wie folgt:

$$P(i \text{ gewinnt gegen } j) = \frac{1}{1 + 10^{E_j - E_i/400}}$$

→

Aktuell sind die drei besten Modelle alle GPT-4-Modelle, und das mit der höchsten ELO-Zahl ist das GPT-4o-Modell *gpt-4o-2024-05-13*. Auf dem vierten Platz liegt das Modell Gemini 1.4 Pro von Google. GPT-3.5 Turbo findet sich an 30. Stelle.

Bringen Sie eine Person mit zwei Modellen in Kontakt – zum Beispiel *gpt-4o-2024-05-13* mit einer ELO-Zahl von 1287 und GPT-3.5-Turbo-0613 mit einer ELO-Zahl von 1120 –, ohne dass diese weiß, welches welches Modell ist, können Sie die Wahrscheinlichkeit abschätzen, mit der die Person das Modell *gpt-4o-2024-05-13* bevorzugen wird, indem Sie die ELO-Zahlen in die obige Gleichung einsetzen. In diesem Fall ergibt die Wahrscheinlichkeitsschätzung 72 %.

Tabelle 1-1 fasst die Entwicklung der GPT-Modelle zusammen.

Tabelle 1-1: Entwicklung der GPT-Modelle

2017	Der Artikel »Attention Is All You Need« von Vaswani et al. wird veröffentlicht.
2018	Das erste GPT-Modell mit 117 Millionen Parametern wird vorgestellt.
2019	Das GPT-2-Modell mit 1,5 Milliarden Parametern wird veröffentlicht.
2020	Das GPT-3-Modell mit 175 Milliarden Parametern wird vorgestellt.
2022	Das Modell GPT-3.5 (ChatGPT) mit 175 Milliarden Parametern wird veröffentlicht.
2023	Das GPT-4-Modell wird vorgestellt, die Menge an Parametern ist aber nicht bekannt.
2024	Das Modell GPT-4o wird Mitte des Jahres von OpenAI vorgestellt.

Sie haben vielleicht schon vom *Foundation-Modell* gehört. Anders als klassische Modelle, die für bestimmte Aufgaben trainiert wurden, werden Foundation-Modelle mit sehr unterschiedlichen Daten trainiert. Dieses umfassende Training verschafft den Modellen ein tiefes Verständnis unterschiedlichster Domänen – und dieses Wissen kann dann optimiert werden, damit die Modelle spezifische Aufgaben erledigen können. Die GPT-Modelle sind Foundation-Modelle. Wie Sie gesehen haben, zeigen sie eine bemerkenswerte Fähigkeit, menschlichen Text für unterschiedlichste Themen zu erzeugen. Durch Optimieren kann das breite Wissen von GPT so spezialisiert werden, dass es in diversen Aufgaben glänzen kann – vom Schreiben von Artikeln bis hin zum Programmieren. So können Foundation-Modelle an Auf-

Inhalt