

Praxisbuch Large Language Models

Sprache mit KI verarbeiten und generieren

» Hier geht's
direkt
zum Buch

DAS VORWORT

Large Language Models (LLMs), auch als große Sprachmodelle bezeichnet, verändern die Welt grundlegend. Dank LLMs können Maschinen menschliche Sprache nicht nur besser verstehen, sondern auch selbst generieren. Dadurch eröffnen sich völlig neue Möglichkeiten im Bereich der künstlichen Intelligenz (KI), die bereits zahlreiche Branchen maßgeblich beeinflussen und auch in Zukunft weiter verändern werden.

Dieses Buch bietet Ihnen eine ebenso umfassende wie anschauliche Einführung in die Welt der LLMs – beginnend mit den theoretischen Grundlagen bis hin zu einer Vielzahl praktischer Anwendungen. Dabei lernen Sie wichtige Entwicklungen im Zusammenhang mit LLMs kennen, darunter einige, die dem Deep Learning vorausgingen, wie die Darstellung von Wörtern als Vektoren. Zudem erfahren Sie mehr über die Transformer-Architektur, die maßgeblich zum Erfolg von LLMs beigetragen hat. Sie erhalten außerdem einen Einblick in die Funktionsweise von LLMs und lernen die verschiedenen Architekturen sowie Techniken zum Trainieren und Feintunen kennen. Darüber hinaus gewinnen Sie einen umfassenden Überblick über die vielfältigen Anwendungsbereiche von LLMs, wie beispielsweise die Textklassifikation, das Clustering, das Topic Modeling, Chatbots, Suchmaschinen und vieles mehr.

Mit diesem Buch, das Ihnen durch zahlreiche praktische Anwendungen und Illustrationen einen leicht verständlichen Zugang zur Thematik bietet, möchten wir die ideale Grundlage für alle schaffen, die die aufregende Welt der LLMs erkunden möchten. Ganz gleich, ob Sie gerade erst in die Materie einsteigen oder bereits Experte sind: Tauchen Sie ein in die Welt der LLMs und lernen Sie, wie Sie diese selbst erstellen können.

Ein leicht verständlicher Zugang

Das Hauptziel dieses Buchs ist es, Ihnen einen *leicht verständlichen* Zugang zum Thema LLMs zu verschaffen. Da sich die Fortschritte im Bereich der Language AI – also KI-basierter Sprachanwendungen – in einem geradezu atemberaubenden Tempo vollziehen, kann es herausfordernd sein, mit den neuesten Entwicklungen Schritt zu

halten. Aus diesem Grund liegt der Schwerpunkt dieses Buchs auf den Grundlagen von LLMs, um Ihnen einen unterhaltsamen und einfachen Lernprozess zu ermöglichen.

Um Ihnen das Thema *so nachvollziehbar wie möglich* zu vermitteln, begegnen Ihnen im Laufe des Buchs zahlreiche Illustrationen. Mithilfe dieser Illustrationen sollen Ihnen die wichtigsten Konzepte und Techniken im Zusammenhang mit LLMs auf anschauliche Weise nähergebracht und der Einstieg in dieses spannende und potenziell weltverändernde Gebiet erleichtert werden.¹

Im gesamten Buch wird strikt zwischen sogenannten Representation-Modellen und generativen Sprachmodellen unterschieden. Bei Representation-Modellen handelt es sich um LLMs, die nicht dazu dienen, Text zu generieren, sondern vielmehr darauf ausgelegt sind, Texte bestmöglich abzubilden. Sie werden häufig für aufgabenspezifische Anwendungsfälle wie die Klassifikation verwendet. Generative Modelle sind hingegen LLMs, die Texte generieren, wie beispielsweise GPT-Modelle. Obwohl einem generative Modelle in der Regel als Erstes in den Sinn kommen, wenn man an LLMs denkt, bieten Representation-Modelle ebenfalls ein breites Spektrum an Anwendungsmöglichkeiten. Im weiteren Verlauf wird das Wort »Large« (groß) in *Large Language Models* relativ vage verwendet. Zum Teil werden sie auch einfach als Sprachmodelle bzw. Language Models bezeichnet, da Abgrenzungen hinsichtlich ihrer Größe oft willkürlich vorgenommen werden und nur bedingt Rückschlüsse auf die Leistungsfähigkeit von Modellen ermöglichen.

An wen sich dieses Buch richtet

In diesem Buch wird vorausgesetzt, dass Sie bereits über Programmiererfahrung in Python verfügen und sich mit den Grundlagen des maschinellen Lernens auskennen. Der Schwerpunkt liegt weniger auf der mathematischen Herleitung von Gleichungen als vielmehr auf der Vermittlung eines guten, intuitiven Verständnisses. Zu diesem Zweck werden in diesem Buch zahlreiche Abbildungen und viele praktische Beispiele angeführt, die die Lerninhalte veranschaulichen und verständlicher machen. Vorkenntnisse bezüglich gängiger Deep-Learning-Frameworks wie PyTorch oder TensorFlow oder im Bereich der Erstellung generativer Modelle sind für dieses Buch nicht erforderlich.

Sollten Sie noch keine Erfahrung im Umgang mit Python haben, empfehlen wir Ihnen als Einstieg die Onlinetutorials von Learn Python (<https://oreil.ly/arcIm>), die die wesentlichen Grundlagen der Programmiersprache vermitteln. Um Ihnen den Lernprozess so einfach wie möglich zu gestalten, steht Ihnen der gesamte Code auf Google Colab (<https://oreil.ly/kSucO>) zur Verfügung – einer Plattform, auf der sich der Code ausführen lässt, ohne dass eine lokale Installation erforderlich ist.

¹ J. Alamar. »Machine learning research communication via illustrated and interactive web articles.« *Beyond Static Papers: Rethinking How We Share Scientific Understanding in ML*. ICLR 2021 Workshop (2021).

Aufbau des Buchs

Das Buch ist grundsätzlich in drei Hauptteile gegliedert. Abbildung 1 bietet Ihnen einen Gesamtüberblick über den Inhalt der einzelnen Kapitel. Diese können auch unabhängig voneinander gelesen werden. Daher können Sie die Kapitel, mit deren Inhalt Sie bereits vertraut sind, problemlos überfliegen.

Teil I: Die Funktionsweise von Sprachmodellen verstehen

In Teil I des Buchs wird die Funktionsweise kleiner und großer Sprachmodelle näher beleuchtet. Zunächst wird ein Überblick über das Fachgebiet und gängige Methoden vermittelt (siehe Kapitel 1). Anschließend werden zwei der wichtigsten Bausteine dieser Modelle, Tokenisierung und Embeddings, vorgestellt (siehe Kapitel 2). Abgerundet wird dieser Teil des Buchs durch eine aktualisierte und um weitere Inhalte ergänzte Fassung von Jays bekanntem Beitrag *The Illustrated Transformer* (<https://oreil.ly/UI4lN>), in der detailliert auf die Architektur dieser Modelle eingegangen wird (siehe Kapitel 3). Dabei werden zahlreiche Begriffe eingeführt und grundlegende Konzepte erläutert, die im weiteren Verlauf des Buchs immer wieder von Bedeutung sein werden.

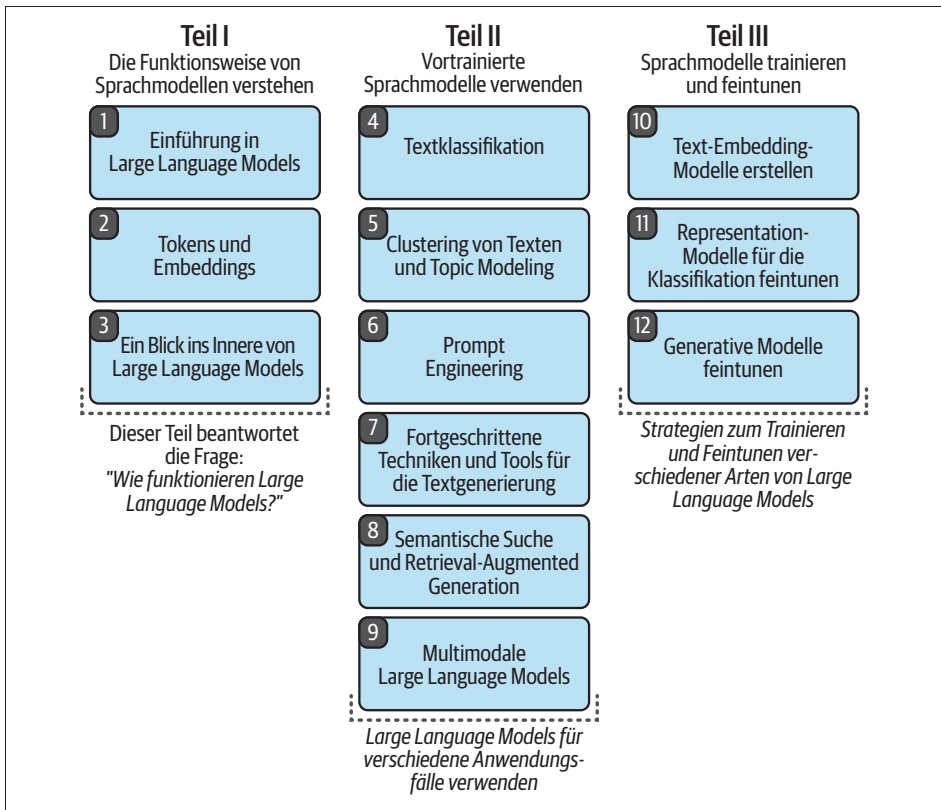


Abbildung 1: Übersicht über die einzelnen Teile und Kapitel des Buchs

Teil II: Vortrainierte Sprachmodelle verwenden

In Teil II werden anhand gängiger Anwendungsfälle verschiedene Einsatzmöglichkeiten von LLMs aufgezeigt. Dabei wird auf vortrainierte Modelle zurückgegriffen und gezeigt, wozu diese bereits ohne weiteres Feintuning imstande sind.

Sie erfahren, wie Sie Sprachmodelle zur überwachten Klassifikation (siehe Kapitel 4), zum Clustering von Texten und zum Topic Modeling (siehe Kapitel 5) verwenden können. Zudem erfahren Sie, wie Sie Texte generieren (siehe die Kapitel 6 und 7) und Embedding-Modelle für die semantische Suche nutzen können (siehe Kapitel 8). Abschließend können Sie sich ein Bild davon machen, wie sich die Möglichkeiten im Bereich der Textgenerierung auf Bilder übertragen lassen (siehe Kapitel 9).

Nachdem Sie sich mit diesen grundlegenden Anwendungsmöglichkeiten von Sprachmodellen vertraut gemacht haben, sind Sie in der Lage, eigenständig Problemstellungen verschiedenster Art zu lösen und zunehmend ausgefeiltere Systeme und Pipelines zu entwickeln.

Teil III: Sprachmodelle trainieren und feintunen

In Teil III des Buchs lernen Sie anspruchsvollere Ansätze kennen und erfahren, wie sich Sprachmodelle trainieren und feintunen lassen. Sie lernen, wie Sie ein Embedding-Modell erstellen und feintunen (siehe Kapitel 10), wie Sie ein BERT-Modell feintunen können, um eine Klassifikation durchzuführen (siehe Kapitel 11), und schließlich auch, wie Ihnen das Feintuning generativer Modelle gelingt (siehe Kapitel 12).

Hardware- und Softwarevoraussetzungen

Generative Modelle auszuführen, erfordert in der Regel eine hohe Rechenleistung und somit einen Computer mit einem leistungsstarken Grafikprozessor (engl. *Graphics Processing Unit*, GPU) bzw. einer leistungsstarken Grafikkarte. Da nicht alle Leserinnen und Leser über einen solchen Computer verfügen, sind alle Beispiele in diesem Buch so konzipiert, dass sie über eine online verfügbare Plattform namens Google Colaboratory (<https://oreil.ly/HQawv>) (Google Colab) ausgeführt werden können. Zum Zeitpunkt des Verfassens dieses Buchs können Sie auf dieser Plattform kostenfrei eine T4 von NVIDIA verwenden, um den Code auszuführen. Diese GPU verfügt über 16 GB Grafikspeicher (engl. *Video Random-Access Memory*, VRAM), was der mindestens erforderlichen Speichergröße für die meisten der im Buch enthaltenen Codebeispiele entspricht.



Nicht für alle Kapitel ist ein Grafikspeicher von mindestens 16 GB erforderlich, da einige Beispiele, wie das Trainieren und Feintunen von Sprachmodellen, rechenintensiver sind als andere, etwa das Prompt Engineering. Die genauen Mindestvoraussetzungen hinsichtlich des Grafikspeichers für die einzelnen Kapitel können Sie dem Repository entnehmen.

Den gesamten Code, die erforderlichen Voraussetzungen und zusätzliche Tutorials finden Sie im Repository dieses Buchs (<https://github.com/HandsOnLLM/Hands-On-Large-Language-Models>). Wenn Sie die Beispiele lokal ausführen wollen, empfehlen wir Ihnen die Verwendung einer Grafikkarte von NVIDIA mit einem Grafikspeicher von mindestens 16 GB. Möchten Sie eine lokale Installation, beispielsweise mit `conda`, durchführen, empfehlen wir Ihnen, Ihre Umgebung wie folgt einzurichten:

```
conda create -n thellmbook python=3.10
conda activate thellmbook
```

Sie können alle erforderlichen Abhängigkeiten installieren, indem Sie das Repository forken oder klonen und dann Folgendes in Ihrer neu erstellten Python-3.10-Umgebung ausführen:

```
pip install -r requirements.txt
```

API-Schlüssel

In den Beispielen verwenden wir sowohl Open-Source- als auch proprietäre Modelle, um Ihnen die jeweiligen Vor- und Nachteile vor Augen zu führen. Um die in diesem Buch vorgestellten proprietären Modelle von OpenAI und Cohere zu verwenden, müssen Sie zunächst ein kostenloses Nutzerkonto erstellen:

OpenAI (<https://oreil.ly/M4nAa>)

Klicken Sie auf der Website auf die Schaltfläche *Sign up*, um ein kostenloses Nutzerkonto zu erstellen. Mit diesem können Sie einen API-Schlüssel erzeugen, mit dem Sie später auf GPT-3.5 zugreifen können. Klicken Sie dann auf das Settings-Symbol und anschließend auf *API keys*, um einen geheimen Schlüssel zu erzeugen.

Cohere (<https://oreil.ly/T63GA>)

Erstellen Sie zunächst ein kostenloses Nutzerkonto auf der Website. Klicken Sie dann auf die Schaltfläche *API keys*, um einen geheimen Schlüssel zu erzeugen.

Bei beiden Nutzerkonten gibt es allerdings Einschränkungen bezüglich der Anzahl der API-Aufrufe, das heißt, mit diesen kostenlosen API-Schlüsseln können Sie nur eine begrenzte Anzahl von Aufrufen pro Minute durchführen. Wir haben diesen Umstand bei allen Beispielen berücksichtigt und falls erforderlich entsprechende lokal ausführbare Alternativen bereitgestellt.

Zum Verwenden der Open-Source-Modelle müssen Sie kein Nutzerkonto erstellen. Eine Ausnahme bildet das in Kapitel 2 verwendete Llama-2-Modell, das Sie nur mit einem Hugging-Face-Nutzerkonto verwenden können:

Hugging Face (https://oreil.ly/_uV3A)

Klicken Sie auf der Website von Hugging Face auf *Sign up*, um ein kostenloses Nutzerkonto zu erstellen. Wählen Sie anschließend im Menü *Settings* die Option *Access Tokens* aus und erstellen Sie ein Token, das Sie zum Herunterladen bestimmter LLMs verwenden können.

Anwendungsfälle mit deutschsprachigen Daten

Die in diesem Buch verwendeten Open-Source-Modelle wurden von den Autoren hinsichtlich ihrer Performance bei den vorgestellten englischsprachigen Codebeispielen ausgewählt (neben weiteren Aspekten wie beispielsweise den erforderlichen Rechenressourcen, um eine Ausführbarkeit über Google Colab zu gewährleisten). Wenn Sie die Codebeispiele mit deutschen Prompts durchspielen oder auch eigene Anwendungsfälle mit deutschsprachigen Daten umsetzen wollen, sollten Sie auf andere Modelle ausweichen, die für den jeweiligen Anwendungsfall besser geeignet sind. Die in den Codebeispielen verwendeten Modelle können Sie problemlos austauschen, indem Sie ein anderes Modell angeben. Je nach Aufgabengebiet kommen eine Vielzahl mehr- bzw. deutschsprachiger Modelle infrage, wie derzeit z.B. Qwen3 (<https://huggingface.co/models?search=qwen3>), Gemma 3 (<https://huggingface.co/models?search=gemma3>), XLM-RoBERTa (<https://huggingface.co/models?search=xlm-roberta>) oder auch EuroBERT (<https://huggingface.co/models?search=eurobert>). Die Suche nach einem geeigneten Modell können Sie am einfachsten über den Model Hub von Hugging Face unter <https://huggingface.co/models?language=de&sort=trending> durchführen. Dort können Sie auch weitere Attribute wie z.B. die von Ihnen verfolgte Aufgabe (unter »Tasks«) oder die Größe des Modells (»Parameters« unter »Main«) angeben.

In diesem Buch verwendete Konventionen

Die folgenden typografischen Konventionen werden in diesem Buch verwendet:

Kursiv

Kennzeichnet neue Begriffe, URLs, E-Mail-Adressen, Dateinamen und Dateierweiterungen.

Konstante Zeichenbreite

Wird für Programmlistings und für Programmelemente in Textabschnitten wie Namen von Variablen und Funktionen, Datenbanken, Datentypen, Umgebungsvariablen, Anweisungen und Schlüsselwörter verwendet.

Konstante Zeichenbreite, fett

Kennzeichnet Befehle oder anderen Text, den die Nutzerin bzw. der Nutzer wörtlich eingeben sollte.

Konstante Zeichenbreite, kursiv

Kennzeichnet Text, den die Nutzerin bzw. der Nutzer je nach Kontext durch entsprechende Werte ersetzen sollte.



Dieses Symbol steht für einen Tipp oder eine Empfehlung.