

Natural Language Processing mit Transformern

Sprachanwendungen mit Hugging Face erstellen

» Hier geht's
direkt
zum Buch

DAS VORWORT

Während Sie diese Zeilen lesen, geschieht ein Wunder: Die Schnörkel auf dieser Seite formen sich zu Wörtern, Konzepten und Emotionen, während sie sich ihren Weg durch Ihren Kortex bahnen. Meine Gedanken vom November 2021 sind nun erfolgreich in Ihr Gehirn eingedrungen. Sollte es ihnen gelingen, Ihre Aufmerksamkeit zu erregen und lange genug in dieser rauen und hart umkämpften Umgebung zu überleben, haben sie vielleicht sogar die Chance, sich weiter zu verbreiten, wenn Sie diese Gedanken mit anderen teilen. Dank der Sprache sind Gedanken zu übertragbaren und hochansteckenden Gehirnbakterien geworden – und ein Impfstoff ist nicht in Sicht.

Glücklicherweise sind die meisten Gehirnbakterien harmlos,¹ und einige sind sogar überaus nützlich. Tatsächlich formen diese menschlichen Gehirnbakterien zwei unserer wertvollsten Schätze: Wissen und Kultur. So wie wir ohne gesunde Darmbakterien nicht richtig verdauen können, können wir ohne gesunde Gehirnbakterien nicht richtig denken. Die meisten Ihrer Gedanken stammen gar nicht von Ihnen: Sie sind in vielen anderen Gehirnen entstanden, gewachsen und haben sich entwickelt, bevor sie Sie infiziert haben. Wenn wir also intelligente Maschinen erschaffen möchten, müssen wir einen Weg finden, auch sie zu infizieren.

Die gute Nachricht ist, dass sich in den letzten Jahren ein weiteres Wunder ereignet hat: Dank mehrerer Durchbrüche auf dem Gebiet des Deep Learning wurden leistungsfähige Sprachmodelle hervorgebracht. Da Sie dieses Buch lesen, sind Ihnen wahrscheinlich schon einige erstaunliche Ausführungen dieser Sprachmodelle begegnet. So z.B. GPT-3, das nach einer kurzen Texteingabe (einem sogenannten Prompt) wie »ein Frosch trifft ein Krokodil« eine ganze Geschichte niederschreiben kann. Obwohl es noch nicht ganz Shakespeare ist, ist es manchmal schwer zu glauben, dass diese Texte von einem künstlichen neuronalen Netz geschrieben wurden. Tatsächlich hilft mir das Copilot-System von GitHub beim Schreiben dieser Zeilen: Sie werden nie erfahren, wie viel ich davon wirklich selbst verfasst habe.

1  Tipps zur Gehirnhygiene finden Sie in CGP Greys hervorragendem Video über Memes (https://youtu.be/rE3j_RHkqJc).

Die Revolution geht weit über die Generierung von Texten hinaus. Sie umfasst den gesamten Bereich der maschinellen Verarbeitung natürlicher Sprache (engl. Natural Language Processing, NLP) – im Deutschen auch als Maschinelle Sprachverarbeitung oder Computerlinguistik (CL) bezeichnet –, von der Textklassifizierung bis zur automatischen Zusammenfassung von Texten, maschinellen Übersetzung, Beantwortung von Fragen, Chatbots, dem Verstehen natürlicher Sprache (engl. Natural Language Understanding, NLU) und mehr. Wo immer Sprache, gesprochener oder geschriebener Text vorkommt, gibt es eine Anwendung im NLP. Sie können bereits Ihr Telefon nach dem morgigen Wetter fragen, mit einem virtuellen Helpdesk-Assistenten chatten, um ein Problem zu beheben, oder aussagekräftige Ergebnisse von Suchmaschinen erhalten, die Ihre Anfrage wirklich zu verstehen scheinen. Doch diese Technologie ist so neu, dass uns das Beste wahrscheinlich erst noch bevorsteht.

Wie die meisten Fortschritte in der Wissenschaft beruht auch die jüngste Revolution im Bereich des NLP auf der harten Arbeit von Hunderten von unbesungenen Helden. Für den Erfolg sind allerdings drei wesentliche Faktoren ausschlaggebend:

- Der *Transformer* ist eine Architektur für neuronale Netze, die im Jahr 2017 in einer bahnbrechenden Arbeit mit dem Titel »Attention Is All You Need« (<https://arxiv.org/abs/1706.03762>) von einem Team von Forschern von Google vorgeschlagen wurde. In nur wenigen Jahren hat sie sich durchgesetzt und die vorherigen Architekturen, die in der Regel auf rekurrenten neuronalen Netzen (engl. Recurrent Neural Networks, RNNs) basieren, verdrängt. Die Transformer-Architektur eignet sich hervorragend zur Erfassung von Mustern in langen Datensequenzen und zur Bewältigung riesiger Datensätze – so gut, dass ihr Einsatz inzwischen weit über den Bereich des NLP hinausgeht und beispielsweise auch in der Verarbeitung von Bildern (engl. Image Processing) Anwendung findet.
- Im Rahmen der meisten Projekte können Sie auf keinen großen Datensatz, mit dem Sie ein Modell von Grund auf trainieren können, zugreifen. Glücklicherweise ist es oftmals möglich, ein Modell herunterzuladen, das bereits auf einem generischen Datensatz *vortrainiert* wurde: Sie müssen es dann nur noch auf Ihrem eigenen (bedeutend kleineren) Datensatz feintunen. Modelle vorzutrainieren, ist seit Anfang der 2010er-Jahre in der Bildverarbeitung gang und gäbe, im NLP beschränkte es sich jedoch auf kontextlose Worteinbettungen (d.h. dichtbesetzte [engl. dense] Vektordarstellungen einzelner Wörter). So hatte zum Beispiel das englische Wort »bear« die gleiche vortrainierte Einbettung (engl. Embedding) im Zusammenhang mit der Nutzung von »teddy bear«, also dem Plüschbären, und »to bear«, was so viel wie aushalten bzw. ertragen bedeutet. Im Jahr 2018 wurden dann in mehreren Veröffentlichungen vollwertige Sprachmodelle vorgeschlagen, die für eine Vielzahl von NLP-Aufgaben vortrainiert und feingetunt werden können. Dadurch änderte sich das gesamte Vorgehen grundlegend.

- Sogenannte *Model Hubs* wie der von Hugging Face sind ebenfalls »Game-Changer«. Anfangs wurden fertig vortrainierte Modelle einfach irgendwo veröffentlicht, sodass es nicht einfach war, das geeignete Modell zu finden, das man benötigte. Murphys Gesetz sorgte dafür, dass PyTorch-Benutzer nur TensorFlow-Modelle ausfindig machen konnten, und umgekehrt. Und wenn man ein Modell gefunden hatte, war es nicht immer einfach, herauszufinden, wie man es feintunen konnte. Hier kommt die Transformers-Bibliothek von Hugging Face ins Spiel: Sie ist quelloffen, unterstützt sowohl TensorFlow als auch PyTorch und erlaubt es, ein hochmodernes, vortrainiertes Modell vom Hugging Face Hub herunterzuladen, es für Ihre Aufgabe zu konfigurieren, es auf Ihrem Datensatz feinzutunen und es zu evaluieren. Die Bibliothek findet zunehmend mehr Verwendung: Im vierten Quartal 2021 wurde sie von mehr als fünftausend Unternehmen und Einrichtungen genutzt und über vier Millionen Mal pro Monat mit dem Paketverwaltungsprogramm `pip` installiert. Darüber hinaus erweitern sich die Bibliothek und ihr Ökosystem über NLP hinaus, sodass inzwischen auch Bildverarbeitungsmodelle (engl. Image Processing) verfügbar sind. Ebenso können Sie zahlreiche Datensätze vom Hub herunterladen, mit denen Sie Ihre Modelle trainieren oder evaluieren können.

Was kann man sich also noch wünschen? Nun, dieses Buch! Es wurde von den Open-Source-Entwicklern von Hugging Face verfasst – einschließlich des Begründers der Transformers-Bibliothek – und das merkt man: Die Breite und Tiefe der Informationen, die Sie auf diesen Seiten finden, ist erstaunlich. Es deckt von der Transformer-Architektur selbst bis hin zur Transformers-Bibliothek und dem gesamten Ökosystem, das sie umgibt, alles ab. Besonders gut gefallen hat mir der praxisnahe Ansatz: Während Sie das Buch durcharbeiten, können Sie gleichzeitig den gesamten Code, der in Jupyter Notebooks vorliegt, direkt nachvollziehen und ausführen. Alle Codebeispiele sind direkt auf den Punkt gebracht und einfach zu verstehen. Die Autoren bringen langjährige Erfahrung im Trainieren sehr großer Transformer-Modelle mit und liefern eine Fülle von Tipps und Tricks, mit denen Sie alles effizient zum Laufen bringen. Und nicht zuletzt ist ihr Schreibstil geradlinig und lebendig: Es liest sich wie ein Roman.

Kurzum, ich habe dieses Buch sehr gerne gelesen und ich bin sicher, Sie werden ebenfalls Gefallen daran finden. Jeder, der an der Entwicklung von Produkten mit modernsten Sprachverarbeitungsfunktionen interessiert ist, sollte es lesen. Es ist randvoll mit all den nützlichen Gehirnbakterien!

Aurélien Géron

November 2021, Auckland (Neuseeland)

Seit ihrer Einführung im Jahr 2017 haben sich Transformer-Modelle zum De-facto-Standard für die Bewältigung einer Vielzahl von Aufgaben im Bereich der natürlichen Sprachverarbeitung (engl. Natural Language Processing, NLP) sowohl in der Wissenschaft als auch in der Industrie entwickelt. Ohne dass Sie es bemerkt haben, haben Sie heute wahrscheinlich bereits mit einem Transformer interagiert: Google verwendet heutzutage das BERT-Modell, um die Suchanfragen der Nutzer besser zu verstehen und so die Suchmaschine zu verbessern. Auch die Modelle der GPT-Familie von OpenAI haben in den Mainstream-Medien wiederholt für Schlagzeilen gesorgt, weil sie in der Lage sind, wie von Menschen hervorgebrachte Texte und Bilder zu generieren.¹ Mithilfe dieser Transformer-basierten Modelle werden Anwendungen wie GitHub's Copilot (<https://copilot.github.com>) betrieben, die, wie in Abbildung 1-1 gezeigt, einen bloßen Kommentar in Quellcode umwandeln können, mit dem automatisch ein neuronales Netz (engl. Neural Network) für Sie erstellt wird!

Weshalb also haben Transformer das Gebiet fast über Nacht verändert? Wie bei vielen großen wissenschaftlichen Durchbrüchen handelte es sich um die Synthese mehrerer Ideen, wie *Attention*, *Transfer Learning* und der *Skalierung neuronaler Netze*, die zu dieser Zeit in der Forschungsgemeinschaft kursierten.

Aber wie nützlich sie auch sein mögen – um in der Industrie Fuß zu fassen, braucht jede ausgefallene neue Methode Werkzeuge, die sie zugänglich machen. Die 🤖² Transformers-Bibliothek (<https://oreil.ly/Z79jF>) und das sie umgebende Ökosystem sind genau darauf ausgerichtet und erleichtern Praktikern, Modelle zu verwenden, zu trainieren und sie mit anderen zu teilen. Dies hat die Verbreitung von Transformer-Modellen stark begünstigt, und die Bibliothek wird heute von über fünftausend Unternehmen und Einrichtungen genutzt. In diesem Buch zeigen wir Ihnen, wie Sie diese Modelle für praktische Anwendungen trainieren und optimieren können.

1 NLP-Forscher neigen dazu, die von ihnen entwickelten Modelle nach Figuren aus der *Sesamstraße* zu benennen. Was diese Akronyme bedeuten, erklären wir in Kapitel 1.

2 Sie erkennen es bestimmt, dies ist das Emoji *Hugging Face*.

```

1 # Create a convolutional neural network to classify MNIST images in PyTorch.
2 class ConvNet(nn.Module):
    def __init__(self):
        super(ConvNet, self).__init__()
        self.conv1 = nn.Conv2d(1, 10, kernel_size=5)
        self.conv2 = nn.Conv2d(10, 20, kernel_size=5)
        self.conv2_drop = nn.Dropout2d()
        self.fc1 = nn.Linear(320, 50)
        self.fc2 = nn.Linear(50, 10)

    def forward(self, x):
        x = F.relu(F.max_pool2d(self.conv1(x), 2))
        x = F.relu(F.max_pool2d(self.conv2_drop(self.conv2(x)), 2))
        x = x.view(-1, 320)
        x = F.relu(self.fc1(x))
        x = F.dropout(x, training=self.training)
        x = self.fc2(x)
        return F.log_softmax(x, dim=1)

```

Abbildung 1-1: Ein Beispiel für GitHub's Copilot-System, das infolge einer kurzen Beschreibung der Aufgabe einen Vorschlag für die gesamte Klasse liefert (alles, was auf `class` folgt, wurde automatisch generiert)

An wen richtet sich dieses Buch?

Dieses Buch richtet sich an Data Scientists und Machine Learning Engineers, die vielleicht schon von den jüngsten Durchbrüchen mit Transformern gehört haben, denen aber ein detaillierter Leitfaden fehlt, um diese Modelle an ihre eigenen Anwendungsfälle anzupassen. Das Buch ist nicht als Einführung in das Machine Learning zu verstehen. Wir gehen davon aus, dass Sie mit der Programmierung in Python vertraut sind und ein grundlegendes Verständnis von Deep-Learning-Frameworks wie PyTorch (<https://pytorch.org>) oder TensorFlow (<https://www.tensorflow.org>) haben. Wir gehen auch davon aus, dass Sie einige praktische Erfahrungen mit dem Trainieren von Modellen auf GPUs besitzen. Obwohl sich das Buch auf die PyTorch-API der `torch.nn`-Bibliothek konzentriert, zeigen wir Ihnen in Kapitel 2, wie Sie alle Beispiele in TensorFlow überführen können.

Die folgenden Ressourcen bieten Ihnen eine gute Grundlage für die in diesem Buch behandelten Themen. Wir gehen davon aus, dass Ihr Kenntnisstand in etwa auf deren Niveau liegt:

- *Praxiseinstieg Machine Learning mit Scikit-Learn, Keras und TensorFlow* von Aurélien Géron (O'Reilly)
- *Deep Learning for Coders with fastai and PyTorch* von Jeremy Howard und Sylvain Gugger (O'Reilly)
- *Natural Language Processing mit PyTorch* von Delip Rao und Brian McMahan (O'Reilly)
- Der Onlinekurs von Hugging Face (<https://oreil.ly/n3MaR>) des Open-Source-Teams von Hugging Face, auch auf Deutsch unter <https://huggingface.co/course/de/>

Was Sie lernen werden

Das Ziel dieses Buchs ist es, Sie in die Lage zu versetzen, Ihre eigenen Sprachanwendungen zu erstellen. Zu diesem Zweck konzentriert es sich auf praktische Anwendungsfälle und geht nur dort auf die theoretischen Aspekte ein, wo es notwendig ist. Der Ansatz des Buchs ist praxisorientiert, und wir empfehlen Ihnen dringend, die Codebeispiele selbst auszuprobieren.

Das Buch deckt alle wichtigen Anwendungen von Transformern im NLP ab, wobei jedes Kapitel (mit wenigen Ausnahmen) einer bestimmten Aufgabenstellung, verbunden mit einem realistischen Anwendungsfall und Datensatz, gewidmet ist. In jedem Kapitel werden außerdem einige zusätzliche Konzepte vorgestellt. Hier ist ein Überblick über die behandelten Aufgabenstellungen (engl. Tasks) und Themen:

- Kapitel 1, *Hallo Transformer*, stellt Transformer vor und ordnet sie in den Kontext ein. Außerdem wird eine Einführung in das Hugging-Face-Ökosystem gegeben.
- Kapitel 2, *Textklassifizierung*, konzentriert sich auf die Sentiment- bzw. Stimmungsanalyse – engl. Sentiment Analysis – (ein gängiges Textklassifizierungsproblem) und stellt die Trainer-Klasse vor.
- Kapitel 3, *Die Anatomie von Transformer-Modellen*, geht näher auf die Transformer-Architektur ein, um Sie auf die folgenden Kapitel vorzubereiten.
- Kapitel 4, *Multilinguale Named Entity Recognition*, konzentriert sich auf die Identifizierung von Entitäten bzw. Eigennamen in verschiedensprachigen Texten (eine Problemstellung im Rahmen der Klassifizierung von Tokens).
- Kapitel 5, *Textgenerierung*, untersucht die Fähigkeit von Transformer-Modellen, Text zu generieren, und stellt Decodierungsstrategien und Maße zur Beurteilung der Qualität vor.
- Kapitel 6, *Automatische Textzusammenfassung (Summarization)*, befasst sich mit der komplexen Sequence-to-Sequence-Aufgabe der Textzusammenfassung und erläutert die für diese Aufgabe verwendeten Maße.
- Kapitel 7, *Question Answering*, konzentriert sich auf den Aufbau eines rezensionsbasierten Fragebeantwortungssystems und stellt das Retrieval mit Haystack vor.
- Kapitel 8, *Effizientere Transformer-Modelle für die Produktion*, befasst sich mit der Leistungsfähigkeit der Modelle. Wir werden die Aufgabe der Intentionserkennung – engl. Intent Detection – (eine Art von Sequenzklassifizierungsproblem) betrachten und Techniken wie Knowledge Distillation, Quantisierung und Pruning untersuchen.
- Kapitel 9, *Ansätze bei wenig bis gar keinen zur Verfügung stehenden gelabelten Daten*, zeigt Möglichkeiten zur Verbesserung der Modellleistung auf, wenn keine großen Mengen an gelabelten Daten zur Verfügung stehen. Wir werden

einen GitHub Issues Tagger erstellen und Techniken wie Zero-Shot-Klassifikation und Datenerweiterung (engl. Data Augmentation) untersuchen.

- Kapitel 10, *Transformer-Modelle von Grund auf trainieren*, zeigt Ihnen, wie Sie ein Modell für die automatische Vervollständigung von Python-Quellcode von Grund auf erstellen und trainieren können. Wir befassen uns mit dem Streaming von Datensätzen und dem Training von Modellen in großem Maßstab und erstellen unseren eigenen Tokenizer.
- Kapitel 11, *Künftige Herausforderungen*, untersucht die Herausforderungen, mit denen Transformer konfrontiert sind, und einige der spannenden neuen Richtungen, die die Forschung in diesem Bereich einschlägt.

Die 🤖 Transformers-Bibliothek bietet mehrere Abstraktionsebenen für die Verwendung und das Training von Transformer-Modellen. Wir beginnen mit den benutzerfreundlichen Pipelines, die es uns ermöglichen, Textbeispiele durch die Modelle zu leiten und die Vorhersagen mit nur wenigen Codezeilen zu ermitteln. Anschließend befassen wir uns mit Tokenizern, Modellklassen und der `Trainer`-Klasse, mit der wir Modelle für unsere eigenen Anwendungsfälle trainieren können. Später werden wir Ihnen zeigen, wie Sie die `Trainer`-Klasse durch die 🚀 Accelerate-Bibliothek ersetzen können, die uns die volle Kontrolle über die Trainingschleife gibt und es uns ermöglicht, große Transformer-Modelle komplett von Grund auf zu trainieren! Jedes Kapitel ist weitgehend in sich abgeschlossen, wobei der Schwierigkeitsgrad der Aufgaben in den späteren Kapiteln zunimmt. Aus diesem Grund empfehlen wir, mit den Kapiteln 1 und 2 zu beginnen, bevor Sie sich dem Thema zuwenden, das Sie am meisten interessiert.

Neben der 🤖 Transformers- und der 🚀 Accelerate-Bibliothek werden wir auch ausgiebig von der 📄 Datasets-Bibliothek Gebrauch machen, die sich nahtlos in andere Bibliotheken integrieren lässt. Die 📄 Datasets-Bibliothek bietet ähnliche Funktionen für die Datenverarbeitung wie Pandas, ist jedoch von Grund auf für die Verarbeitung großer Datenmengen und Machine Learning (bzw. maschinelles Lernen) konzipiert.

Mit diesen Tools haben Sie alles, was Sie benötigen, um fast jede Herausforderung im Bereich des NLP zu meistern!

Software- und Hardwareanforderungen

Aufgrund des praxisorientierten Ansatzes dieses Buchs empfehlen wir Ihnen dringend, die Codebeispiele auszuführen, während Sie die einzelnen Kapitel lesen. Da wir es mit Transformern zu tun haben, benötigen Sie Zugang zu einem Computer mit einer NVIDIA-GPU, um diese Modelle trainieren zu können. Glücklicherweise gibt es online mehrere kostenlose Optionen, die Sie nutzen können, u. a.:

- Google Colaboratory (<https://oreil.ly/jyXgA>)
- Kaggle Notebooks (<https://oreil.ly/RnMP3>)
- Paperspace Gradient Notebooks (<https://oreil.ly/mZEKy>)

Um die Beispiele ausführen zu können, müssen Sie die Installationsanleitung befolgen, die wir im GitHub-Repository des Buchs bereitstellen. Sie finden die Anleitung und die Codebeispiele unter <https://github.com/nlp-with-transformers/notebooks>.



Wir haben die meisten Kapitel mit NVIDIA Tesla P100 GPUs entwickelt, die über 16 GB an Speicher verfügen. Einige der freien Plattformen bieten GPUs mit einem geringeren Speicher an, sodass Sie beim Trainieren der Modelle möglicherweise die Batchgröße verringern müssen.

In diesem Buch verwendete Konventionen

Die folgenden typografischen Konventionen werden in diesem Buch verwendet:

Kursiv

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

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 der Nutzer wörtlich eingeben sollte.

Konstante Zeichenbreite, kursiv

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



Tipp

Dieses Symbol steht für einen Tipp oder eine Empfehlung.



Hinweis

Dieses Symbol steht für einen allgemeinen Hinweis.



Warnung

Dieses Symbol warnt oder mahnt zur Vorsicht.

Verwenden von Codebeispielen

Zusätzliche Materialien (Codebeispiele, Übungen usw.) können Sie unter <https://github.com/nlp-with-transformers/notebooks> herunterladen.

Wir haben eine Webseite für dieses Buch, auf der wir Errata, Beispiele und zusätzliche Informationen veröffentlichen. Sie können diese Seite unter <https://www.oreilly.com/library/view/natural-language-processing/9781098136789/> aufrufen.

Dieses Buch dient dazu, Ihnen bei der Erledigung Ihrer Arbeit zu helfen. Im Allgemeinen dürfen Sie die Codebeispiele aus diesem Buch in Ihren eigenen Programmen und der dazugehörigen Dokumentation verwenden. Sie müssen uns dazu nicht um Erlaubnis bitten, solange Sie nicht einen beträchtlichen Teil des Codes reproduzieren. Beispielsweise benötigen Sie keine Erlaubnis, um ein Programm zu schreiben, in dem mehrere Codefragmente aus diesem Buch vorkommen. Wollen Sie dagegen eine CD-ROM mit Beispielen aus Büchern von O'Reilly verkaufen oder verbreiten, benötigen Sie eine Erlaubnis. Eine Frage zu beantworten, indem Sie aus diesem Buch zitieren und ein Codebeispiel wiedergeben, benötigt keine Erlaubnis. Eine beträchtliche Menge Beispielcode aus diesem Buch in die Dokumentation Ihres Produkts aufzunehmen, bedarf hingegen unserer ausdrücklichen Zustimmung.

Wir freuen uns über Zitate, verlangen diese aber nicht. Ein Zitat enthält Titel, Autor, Verlag und ISBN. Beispiel: »*Natural Language Processing mit Transformern* von Lewis Tunstall, Leandro von Werra und Thomas Wolf (O'Reilly). Copyright 2023 dpunkt.verlag, ISBN 978-3-96009-202-5.«

Wenn Sie glauben, dass Ihre Verwendung von Codebeispielen über die übliche Nutzung hinausgeht oder außerhalb der oben vorgestellten Nutzungsbedingungen liegt, kontaktieren Sie uns bitte unter komentar@oreilly.de.

Danksagungen

Das Schreiben eines Buchs über einen der sich am schnellsten entwickelnden Bereiche des maschinellen Lernens wäre ohne die Hilfe vieler Menschen nicht möglich gewesen. Wir danken dem wunderbaren O'Reilly-Team und insbesondere Melissa Potter, Rebecca Novack und Katherine Tozer für ihre Unterstützung und Beratung. Das Buch hat auch von großartigen Fachgutachtern profitiert, die unzählige Stunden damit verbracht haben, uns unschätzbare Feedback zu geben. Besonders dankbar sind wir Luca Perozzi, Hamel Husain, Shabie Iqbal, Umberto Lupo, Malte Pietsch, Timo Möller und Aurélien Géron für ihre ausführlichen Rezensionen. Wir danken Branden Chan von deepset (<https://www.deepset.ai>) für seine Hilfe bei der Erweiterung der Haystack-Bibliothek zur Unterstützung des Anwendungsfalls in Kapitel 7. Die wunderschönen Illustrationen in diesem Buch verdanken wir der fantastischen Christa Lanz (<https://christalanz.ch>) – vielen Dank, dass Sie dieses Buch zu etwas ganz Besonderem gemacht haben. Wir hatten auch das Glück, die Unterstützung des gesamten Hugging-Face-Teams zu erhalten. Vielen Dank an