

Einheitsstrafen DSGVO

Neue europäische Regeln für DSGVO-Bußgelder

Erstmals haben sich die europäischen Datenschutzbehörden auf gemeinsame Grundsätze geeinigt, nach denen Bußgelder zu verhängen sind. Diese lassen den Behörden in den einzelnen Ländern aber nach wie vor große Spielräume.

Von Joerg Heidrich

Link zu den EDSA-Guidelines: ct.de/yfyw

[Mehr lesen ...](#)

Google Analytics 4 & DSGVO: Ihre Datenschutz-Checkliste für die Umstellung am 1. Juli 2023

Ab dem 1. Juli 2023 stellt Google seinen Dienst Google Analytics dauerhaft auf die neue Version "Google Analytics 4" (kurz "GA4") um. Mit dieser Änderung kommen nicht nur technische, sondern auch rechtliche Veränderungen auf Nutzer von Google Analytics zu.

In diesem Beitrag erfahren Sie, was sich datenschutzrechtlich ändert, ob und wie Sie Google Analytics 4 einsetzen dürfen. Am Ende des Beitrags erhalten Sie eine Checkliste mit den wichtigsten Prüfpunkten vor dem Einsatz von Google Analytics 4.

Wie erfolgt die Umstellung auf Google Analytics 4?

Sollten Sie bisher noch keine eigenen Schritte zur Umstellung unternommen haben, so hat Google bereits im März 2023 automatisch Google Analytics 4 für Sie eingerichtet, basierend auf Ihren bisherigen Einstellungen und Zielen für Universal Analytics.

Bis Ende Juni 2023 haben Sie noch die Möglichkeit, Daten in der alten Version (Universal Analytics) zu sammeln und zu nutzen. Danach beginnt eine Übergangszeit von sechs Monaten, in der Sie auf alte Daten zugreifen können. Ab dann müssen Sie Google Analytics 4 einsetzen.

The screenshot shows a notification in the Google Analytics 4 interface. At the top, a red warning icon is followed by the text: "Ab 1. Juli 2023 werden in dieser Property keine Daten mehr verarbeitet. Schließen Sie die Einrichtung Ihrer neuen Google Analytics 4-Property (GA4) so schnell wie möglich ab. Andernfalls werden ihre Einstellungen möglicherweise automatisch auf Grundlage Ihrer ursprünglichen Property konfiguriert, sofern Sie diese Funktion nicht im Einrichtungsassistenten deaktivieren." Below this text are two buttons: "Weitere Informationen" and "GA4-Einrichtung fertigstellen".

In the center, a white modal window is displayed with the title "Fast geschafft – Migration zu Google Analytics 4 (GA4) abschließen". The text inside the modal reads: "In der aktuellen Property werden ab dem 1. Juli 2023 keine Daten mehr verarbeitet. Verwenden Sie den Assistenten zum Einrichten einer GA4-Property für die nächsten Schritte." Below the text is a digital clock showing the time remaining: "Days: 08, Hours: 12, Minutes: 23, Seconds: 23". At the bottom of the modal are two buttons: "Nein danke" and "Zum GA4-Einrichtungsassistenten".

Was ist technisch neu an Google

Analytics 4?

Universal Analytics und Analytics 4 sind beide Versionen von Googles Analysetool, jedoch mit technischen Unterschieden.

Universal Analytics fokussiert sich auf Sitzungen und Nutzerinteraktionen auf einzelnen Geräten. Im Gegensatz dazu konzentriert sich Analytics 4 auf die Nachverfolgung von Nutzern über verschiedene Geräte hinweg und ermöglicht eine detailliertere Erfassung des Nutzerverhaltens durch sogenannte "Events", was tiefere Einblicke in ihr Verhalten bietet.

Kurzum, Analytics 4 ist die neuere Version, die detailliertere und geräteübergreifende Daten liefert. Umgekehrt hat Google aber auch die Datenschutzaspekte verbessert.

Versionsbezeichnung: In der Praxis verwendet man die Versionsbezeichnungen selten, sondern spricht schlicht von "Google Analytics". Ab Juli 2023 ist damit automatisch Google Analytics 4 gemeint.

Ist Google Analytics 4 datenschutzrechtlich sicherer?

Google Analytics 4 erleichtert zwar die Nachverfolgung des Nutzerverhaltens, bietet aber zugleich neue Datenschutzfunktionen:

- **Cookie-los:** Mit Google Analytics 4 können Tracking-Verfahren eingerichtet werden, die ohne Cookies auskommen.
- **Serverseitiges Tracking:** Nutzerdaten können auf dem eigenen Server pseudonymisiert werden, bevor sie an Google übermittelt werden.
- **Standort EU:** Google zufolge werden alle Daten von Endgeräten in der EU auf Servern innerhalb der EU gespeichert und verarbeitet.

- **IP-Kürzung in der EU:** Im Gegensatz zu Universal Analytics erfolgt die Kürzung der IP-Adressen der Nutzer auf EU-Servern von Google. Das bedeutet, dass ein entsprechend pseudonymer Datensatz in die USA übermittelt wird.
- **Kürzere Aufbewahrungsfristen:** Im Gegensatz zur bisherigen Standardlöschfrist von 26 Monaten bei Universal Analytics erlaubt GA4 keine längere Aufbewahrung als 14 Monate für Daten.
- **Google Signals:** Durch die [Deaktivierung von Google Signals](#) kann die Verknüpfung mit einem Google-Konto verhindert werden.
- **Geo- und Geräteinformationen:** Die Genauigkeit der gesammelten Geo- und Geräteinformationen kann angepasst werden.

Die genannten Maßnahmen tragen zur Verbesserung des Datenschutzniveaus bei der Verwendung von Google Analytics 4 bei. Beachten Sie jedoch, dass die Nutzung von Google Analytics damit nicht automatisch zulässig und rechtssicher wird.

Die drei größten Fehler bei agiler Transformation

Die drei Wege ins Glück, oder besser gesagt ins Unglück, sind, wie ein Unternehmen garantiert nicht agil wird. Wenn Sie keine schnellere Time-to-Market wünschen, nicht auf Ihre Kunden eingehen wollen, aber dennoch den Anschein erwecken möchten, agil zu sein und bemüht zu transformieren, folgen Sie einem der drei Wege – oder am besten folgen Sie allen dreien, nur um sicherzugehen.

Die letzten Jahre haben gezeigt: Wer auf dem Markt erfolgreich sein will, kommt um neue Formen der Unternehmensführung und Strukturierung nicht herum. In einer Zeit, in der die Märkte weniger vernetzt waren, Kunden nicht miteinander in Austausch gehen, also Erfahrungen zu Produkten teilen konnten, war es für die meisten Unternehmen einfacher. Auch konnten Methoden, die in einem anderen Unternehmen funktioniert haben, einfach auf das eigene übertragen werden. Nicht umsonst sind heute die meisten Unternehmen nach dem gleichen Muster aufgebaut, unterteilt nach Einheiten mit dem Ziel, möglichst effizient zu arbeiten. Und mit Effizienz ist in diesem Fall gemeint, effizient bzgl. der eigenen Aufgaben, nicht Richtung Kunde.

Dieses Verhalten, die Übernahme von Methoden, teilweise Best Practices genannt, lässt sich auch heute noch gut beobachten. Erfolgreiche Unternehmen kopieren einfach stumpf eine Idee und implementieren sie im eigenen Unternehmen, mit der Annahme, die gleichen Effekte wie im beobachteten Unternehmen zu erzielen. Die Implementierung einer Methode wird zum Selbstzweck. Manche Unternehmen gehen oberflächlich noch weiter und wählen einen der drei Wege ins Unglück.

- *Weg 1 – „Wir planen die Transformation“*: Natürlich muss es gut durchdacht und von A bis Z geplant sein. Wir können keine Transformation angehen, ohne diese geplant zu haben, wir müssen ja wissen, wann es losgeht, wann es beendet ist, also wann wir endlich agil sind und natürlich auch, was es kostet.
- *Weg 2 – „Wir brauchen ein eigenes Framework“*: Ist mal ein Plan gemacht (oder auch keiner), deutet sich in einem Unternehmen nur der Wille an, kommen sofort Bedenken und Bedenkensträger. Wie, wir wollen Scrum einführen? Das ist doch nur für Teams, wir sind hier ein Konzern und überhaupt geht das bei uns nicht, wir sind anders, wir müssen unser eigenes Framework entwickeln, das zu unseren Rahmenbedingungen passt.
- *Weg 3 – „Change-Team, mit Plan und Soll-Ist-Vergleich“*:

Plan und eigenes Framework, natürlich muss der Plan auch kontrolliert werden und ein Soll-Ist-Vergleich stattfinden, dafür brauchen wir ein Change-Team, das sich dieser Thematik annimmt. Bitte das Change-Team auch damit beauftragen, ein Framework und einen Plan zu entwickeln. Und ganz wichtig: Das muss geheim bleiben, mit niemandem reden, das würde nur für Verunsicherung sorgen (es werden vielleicht Menschen entlassen oder Ähnliches). Wir müssen das erst gut durchdenken und dann alle mitnehmen, damit es auch klappt.

Ich glaube, die meisten von uns kennen solche Szenarien nur zu gut und haben erlebt, wie gut oder eher schlecht sie funktionieren.

Die ReG AG stand vor einigen Jahren vor einer ähnlichen Herausforderung (**Abb. 1**). Die Messe lief deutlich schlechter als erhofft, die Kunden waren überhaupt nicht begeistert vom vorgestellten Produkt. Der Vorstand hat zwei Maßnahmen ergriffen: Einen neuen Vertriebsvorstand geholt und Richard und seine designierte CTO, Silke, damit beauftragt, das Unternehmen zu transformieren. Richard kannte das laut eigener Aussage bereits, für ihn war Agilität etwas für Hippies, die sich nicht an Zahlen messen lassen wollten. Daher forderte er vehement ein, dass die Transformation gründlich geplant wird.



Abb. 1: Die ReG AG steht vor der Herausforderung

Der erste Weg

Richard schlug folgenden Transformationsplan vor: Wir haben ein Kick-off-Meeting, zu dem alle kommen müssen. Im Kick-off bestimmen wir, welche Teams (falls sie schon existieren) jetzt nach Scrum oder anderen Frameworks arbeiten. Ganz wichtig: Wir überlassen es nicht den Teams, ihre Methode selbst zu wählen. Wir bestimmen, wer der Product Owner sein wird und wer der Scrum Master. Da die ReG AG hier keine Erfahrung hat, kaufen wir uns einen ein, oder schicken schnell jemanden auf die zweitägige Scrum-Master-Schulung, die reicht dann schon, um als Scrum Master im Unternehmen tätig zu sein. Wer wird der Product Owner? Nehmen wir am einfachsten den Projektleiter des Teams, der hat das jetzt auch schon gut gemacht. Schulung für ihn sparen wir uns, das kann dann der Scrum Master richten, so anders ist Scrum nun auch wieder nicht.

Dann richten wir uns einige Meilensteine ein, um zu schauen, wie das Team oder die Teams nach Scrum arbeiten, in die ersten Reviews werden die Chefs eingeladen, die dann absegnen, dass die Arbeit gut, also richtig gemacht wurde. Die Vorschläge vom Scrum Master, auch Kunden einzuladen, wischen wir mit dem Kommentar vom Tisch, dass das bei uns nicht geht, wir sind die Experten, nicht die Kunden.

Der zweite Weg

Nachdem wir nun über den ersten Weg gesprochen haben, schauen wir uns den zweiten Weg zum (Un-)Glück an: Wir entwickeln unser eigenes Framework. Warum? Nun, Scrum kommt aus der Softwareindustrie und wir sind kein Softwareunternehmen, und selbst wenn wir eins sind, ging das vielleicht in Amerika, aber in Deutschland geht das nicht, schon allein wegen unserer Qualitätsansprüche; wir müssen schließlich am Ende genau wissen, was rauskommt. Also bilden wir ein Team aus Experten, kaufen Expertise hinzu, lesen viele Bücher.

Wir nehmen uns den Scrum Guide und schauen, was bei uns funktionieren könnte und was nicht. Wir ignorieren mal die Frage, ob es dann überhaupt noch eine Transformation gibt oder ob wir nur versuchen, Scrum in unsere Prozesse zu pressen (es sollte doch eher umgekehrt sein, die Prozesse und aktuellen Verhalten hinterfragen, um an dieser Stelle besser zu werden). Nein, wir schauen uns Scrum an und ändern es ab, und da Scrum nur für Teams ist, brauchen wir mehr. Wir nehmen z. B. SAFe. SAFe hat viele Ebenen, Rollen, Artefakte etc., aber auch diese passen den wenigsten Unternehmen. Scrum hätte ein kleines Team binnen Tagen oder Stunden anpassen (natürlich verschlimmbessern) können, bei SAFe braucht es schon länger. Nehmen wir uns wieder den Plan von Weg 1 vor und verschieben den Kick-off um mindestens sechs Monate. In diesen sechs Monaten erarbeiten wir ein neues Framework, das genau auf uns zugeschnitten ist, was wir dann stolz beim Kick-off vorstellen.

Der dritte Weg

In den dritten Weg kann Weg 1 und 2 einfließen – der ultimative Weg zum Glück. Ein Change-Team oder Transformationsteam wird ins Leben gerufen und damit beauftragt, die Transformation zu planen, durchzuführen und zu kontrollieren. Dazu werden Weg 1 und Weg 2 angegangen, wenn möglich aber im Geheimen, um niemanden im Unternehmen mit halbgebaren Ansätzen zu verunsichern. Ganz nach dem Ansatz von Taylor, das Denken (in dem Fall das Planen) und das Ausführen zu trennen. Es müssen für jeden Transformationsschritt KPIs festgelegt werden, etwa wie viele Scrum Master nach sechs Monaten existieren müssen, wie viele Teams, die nach Scrum arbeiten und so weiter. Die Teams müssen regelmäßig Folien zum Fortschritt der Implementationsmaßnahmen erstellen.

Fazit

Warum schafft es ein Unternehmen nicht, mit diesen drei Wegen agil zu werden? Weg 1 ist der Versuch einer deterministischen Planung, die im komplizierten Umfeld funktionieren mag, da hier Ursache und Wirkungsbeziehungen durch schlichtes Nachdenken (in diesem Fall durch Planen) erhoben werden können. Eine Transformation findet, da hier Menschen beteiligt sind und Menschen nicht planbar sind, im komplexen Umfeld statt, daher ist jeder Plan zum Scheitern verurteilt. Du kannst nicht mit einem Wasserfallplan agil werden.

Weg 2 impliziert etwas Ähnliches, Planbarkeit, und verwechselt Wissen mit Erfahrung. Um ein Framework anzupassen, benötige ich Erfahrung und nicht nur Wissen. Die meisten Menschen lernen gleich, nehmen wir als Beispiel das Kochen. Wenn ich kochen lerne, und das erste Mal ein Rezept ausprobiere, koche ich es strikt nach Rezept. Über die Zeit nehme ich Anpassungen vor, ich habe Erfahrungen mit dem Rezept gesammelt und würze es vielleicht anders, je nach dem, was ich bevorzuge. Nach vielen weiteren Erfahrungen bin ich dann fähig, eigene Rezepte

zu entwickeln. Kaum jemand würde auf die Idee kommen, bei ersten Kochversuch das Kochbuch zu lesen und Änderungen am Rezept zu machen. Was beim Kochen nicht funktioniert, funktioniert auch beim Entwickeln von Methoden für eine Transformation nicht. Folge und verstehe Shu Ha Ri!

Und der dritte Weg glaubt weiterhin daran, dass eine Trennung von Denken und Ausführen sinnvoll ist und verwechselt Output mit Outcome. Auch die ReG AG hat erkannt, dass ein Transformationsplan oder ein eigenes Framework nicht dabei helfen werden, das Unternehmen agil auszustellen. Wie entscheiden Sie sich?



René Schröder ist seit 2001 erfolgreich als Agile Coach und Trainer in unterschiedlichen Branchen unterwegs. Seine Erfahrungen teilt er gerne in diversen Kolumnen. Seit 2020 veröffentlicht er außerdem die Panda Story (http://regsus.de/panda_story/), eine Geschichte, in der Ray, Howard und Britta eine agile Transformation bei der ReG AG erleben.

KI-Recht: ChatGPT, Bard und

Co.

Über Risiken beim Einsatz von KI wird nicht erst seit ChatGPT diskutiert. Der geplante EU AI Act wird jedoch nicht alle Aspekte regeln.

Von Tobias Haar

-tract

- Die KI-Regulierung der EU soll noch 2023 kommen. Sie soll insbesondere Grundrechtsverletzungen bei Betroffenen verhindern, deren Daten unter KI-Einsatz verarbeitet werden.
- Der zugrunde liegende Ansatz ist risikobasiert und sieht für die verschiedenen Risikostufen daran angepasste Maßnahmen und Pflichten vor.
- Auch eine EU-Richtlinie für KI-Haftung ist in Vorbereitung. Sie soll Geschädigten die Beweislast erleichtern und sieht schon dann einen Anspruch auf Schadenersatz vor, wenn ein ursächlicher Zusammenhang des Schadens mit der KI nach vernünftigem Ermessen wahrscheinlich ist.

„Wir können also nicht wissen, ob uns die künstliche Intelligenz unendlich helfen wird, ob sie uns ignoriert und beiseiteschiebt oder ob sie uns möglicherweise zerstört.“ So äußerte sich bereits 2017 der berühmte Physiker Stephen Hawking. Der jüngste Hype um den Einsatz künstlicher Intelligenz als Chatbots in Suchmaschinen, namentlich ChatGPT in Microsofts Bing oder Bard bei Google, wirft ein Schlaglicht auf die aktuellen Entwicklungen im Bereich der KI. Zwar befindet sich die juristische Betrachtung dieser Phänomene noch am Anfang, erste Diskussionen werden aber bereits vehement geführt. Chatbots sind ein anschauliches Beispiel für den KI-Einsatz, um sich einigen wichtigen KI-Rechtsfragen zu

nähern.

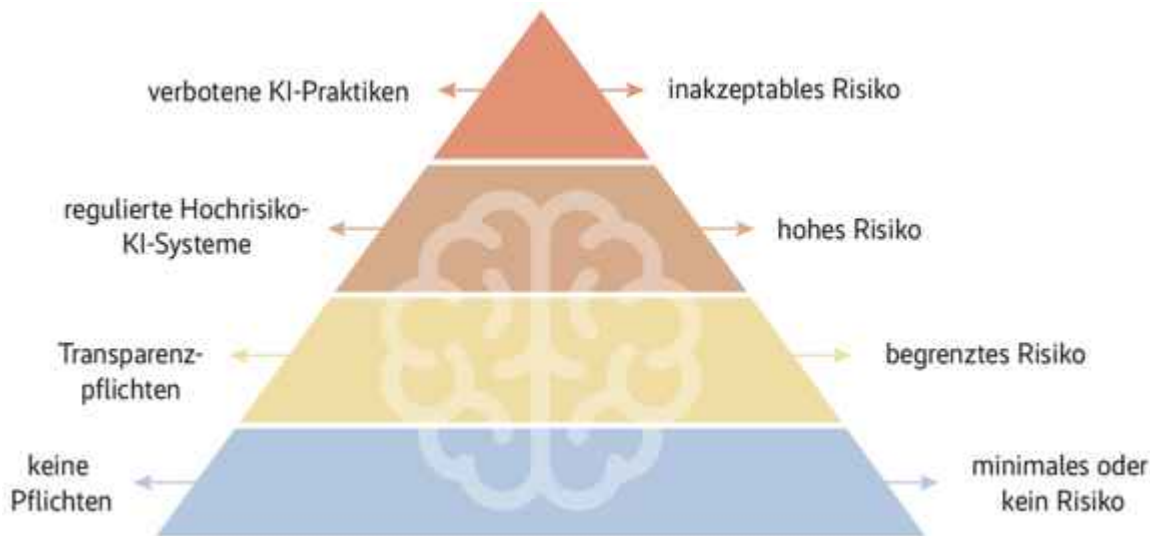
EU-Regulierung in den Startlöchern

Auf EU-Ebene wird derzeit am „Artificial Intelligence Act“, kurz AI Act, gearbeitet. Die Verordnung steht bereits kurz vor der abschließenden Abstimmung zwischen EU-Kommission, EU-Rat und EU-Parlament. Wie auch die Datenschutz-Grundverordnung wird sie nach Wirksamwerden in allen EU-Staaten unmittelbar gelten, ohne dass es nationaler Umsetzungen bedarf. Mit einem Inkrafttreten ist noch 2023 zu rechnen, da den EU-Institutionen aufgrund der für 2024 anstehenden Europawahlen nicht mehr viel Zeit bleibt, ihre Agenda umzusetzen. Gelingt dies nicht, droht eine erhebliche Verzögerung der einheitlichen Regulierung von KI in der EU.

Die EU beschreibt das Spannungsfeld, in dem der AI Act regulierend eingreifen soll, wie folgt: „Der Ansatz der EU für künstliche Intelligenz konzentriert sich auf Exzellenz und Vertrauen, um die Forschungs- und Industriekapazitäten zu stärken und die Grundrechte zu gewährleisten.“ Seit zwei Jahren wird am Verordnungstext gefeilt. Über 3000 Änderungsanträge wurden eingereicht. Bereits die Definition von KI ist umstritten. Ein aktueller Vorschlag lautet, den Anwendungsbereich auf Systeme zu beschränken, „die durch maschinelle Lerntechniken und wissensbasierte Ansätze entwickelt wurden“.

Der AI Act wird einen risikobasierten Ansatz verfolgen. Das soll gewährleisten, „dass die auf dem EU-Markt in Verkehr gebrachten und in der Union verwendeten Systeme künstlicher Intelligenz (KI) sicher sind und die bestehenden Grundrechte und die Werte der Union wahren“. Verboten werden soll der KI-Einsatz bei der sozialen Bewertung von Personen. Für Hochrisikosysteme sind unter anderem Vorgaben im Bereich Datenqualität und Dokumentation vorgesehen. Daneben sind Transparenzvorgaben und weitere Schutzmaßnahmen zugunsten der Betroffenen geplant. Wie üblich sind schmerzhaft Bußgelder

und behördliche Maßnahmen bei Nichteinhaltung vorgesehen.



Der risikobasierte Ansatz der KI-Regulierung soll verhindern, dass Grundrechte Betroffener und Werte der Europäischen Union verletzt werden. *Europäisches Parlament*

Mitten in den Diskussionen um den AI Act hat das Bundesverfassungsgericht Stellung zu einem der umstrittenen Themen genommen. Es geht darum, wie die Rechtsgrundlagen für die Datenauswertung mit KI durch die Polizei zur Gefahrenabwehr ausgestaltet sein müssen. Grundsätzlich ist eine automatisierte Datenauswertung nach Auffassung der Karlsruher Richter zulässig. Sie störten sich aber daran, dass die Polizeigesetze in Hamburg und Hessen nicht regeln, wann früher erhobene Daten für einen neuen Zweck ausgewertet werden dürfen.

Missbrauch von Datenverknüpfungen vermeiden

Insbesondere bei Daten, die aus einer Onlinedurchsuchung oder einer Wohnraumüberwachung stammen, müssen die Grundrechte der Betroffenen ausreichend gewahrt werden. Schafft KI durch die erstmalige Verknüpfung von Informationen gänzlich neues Wissen, gilt es, das allgemeine Persönlichkeitsrecht besonders zu berücksichtigen. Unter Fachleuten wird für die neu zu fassenden Regelungen in den Polizeigesetzen ähnlich dem AI-Act-Ansatz ein abgestufter Ansatz diskutiert. Je nach

abzuwehrender Gefahr dürften danach bestimmte Analyseinstrumente zum Einsatz kommen. Zu berücksichtigen wäre auch die zuvor angewandte Methode zur Erhebung der Daten.

Immer wenn von Daten die Rede ist, ist auch die Datenschutz-Grundverordnung (DSGVO) nicht weit. Sie gilt dann, wenn ein Chatbot personenbezogene Daten verarbeitet. Sie greift nur dann nicht, wenn es sich um nicht personenbezogene Daten handelt, also bei anonymen oder anonymisierten Daten. Wenn beim KI-Einsatz nur solche Daten verwendet werden, ist die juristische Compliance deutlich einfacher (siehe [1]).

Jede Form der Datenverarbeitung muss zulässig sein. Das ist sie, wenn die DSGVO oder eine informierte Einwilligung des Betroffenen dies erlauben. Wie in allen anderen Fällen kann der Betroffene bei KI-Einsatz Auskunfts-, Berichtigungs- und Löschungsrechte geltend machen. Wenn Unternehmen wie das hinter ChatGPT stehende OpenAI in Europa aber keine Niederlassung haben, wird die Rechtsdurchsetzung schwierig.

Von Bedeutung beim KI-Einsatz ist Artikel 22 der DSGVO. Er verbietet es, rechtlich erhebliche oder beeinträchtigende Entscheidungen über Menschen zu treffen, die ausschließlich einer automatisierten Datenverarbeitung entstammen. Ausdrücklich genannt wird in diesem Zusammenhang das Profiling. Ganz allgemein bedarf es einer DSGVO-konformen Datenschutzfolgenabschätzung, wenn risikobehaftete Datenverarbeitungen erfolgen. Dies ist insbesondere bei Gesundheitsdaten der Fall.

Bei Regelverletzungen droht Verbot

Jüngst hat die italienische Datenschutzaufsicht den Chatbot Replika verboten. Zur Begründung verwies sie auf die Verletzung von Transparenzvorschriften nach DSGVO und die unzulässige Verarbeitung von Daten Minderjähriger. Brisant an dem Chatbot ist, dass dieser auf Empathie aufbaut und beworben wird als „der KI-Begleiter, der sich kümmert“. Angeblich soll

die App zudem Daten im Auftrag russischer Behörden sammeln. Künftig müssen sich solche KI-Anwendungen auch am AI Act messen lassen.

Wer aber muss letztlich das Datenschutzrecht beim Einsatz von KI einhalten? Verantwortliche Stelle aus DSGVO-Sicht für Datenverarbeitungen ist, wer „allein oder gemeinsam mit anderen über die Zwecke und Mittel der Verarbeitung von personenbezogenen Daten entscheidet“. Im Fall von ChatGPT wäre dies OpenAI, im Falle der Einbindung von Bard in Microsoft-Office-Applikationen wäre es Microsoft und so weiter. Wenn allerdings der Nutzer selbst KI-Systeme mit personenbezogenen Daten füttert, kommt auch er als Verantwortlicher in Betracht. Womöglich sind KI-Anbieter und -Nutzer auch als gemeinsam Verantwortliche in der Pflicht. Es kommt auf den Einzelfall an.

Und wer zahlt, wenn durch KI-Einsatz ein Schaden entstanden ist? Wenn einer Person ein Schaden entstanden ist und ein Vertrag zwischen Nutzer und Anbieter besteht, ergibt sich ein etwaiger Schadenersatzanspruch aus dem Vertrag. Nach hiesiger Rechtsordnung greifen ergänzend zu einem Vertrag die schuldrechtlichen Vorschriften im Bürgerlichen Gesetzbuch.

Erst 2022 hat der Gesetzgeber im BGB den Mängelbegriff im Kaufrecht und zahlreiche weitere Vorschriften ergänzt – unter anderem, um die zunehmende digitale Bereitstellung von Waren, aber auch rein digitale Dienstleistungen juristisch besser zu berücksichtigen. Auch auf mangelhafte KI-Leistungen sind diese Vorschriften anwendbar.

KI-Chatbots und das Urheberrecht

KI-gestützte Chatbots wie ChatGPT geben Texte aus. Aus juristischer Sicht ist das relevante Rechtsgebiet in diesem Zusammenhang das Urheberrecht. In dessen Zentrum steht das Grundverständnis, dass nur kreativ-schöpferische Leistungen eines Menschen urheberrechtlichen Schutz genießen können.

Damit scheidet die KI selbst als urheberrechtsschöpfend von vornherein aus. Sie ist nur ein handwerkliches oder technisches Werkzeug eines dahinterstehenden Menschen. Ihre rein technische Leistung spielt aus urheberrechtlicher Sicht keine Rolle.

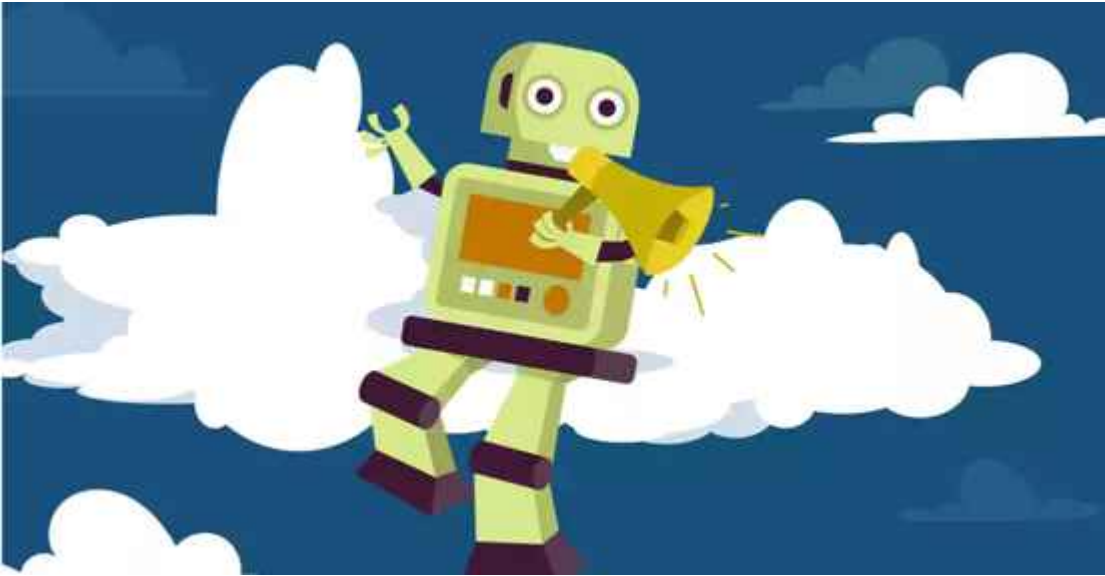
Auch der KI-Entwickler und -Betreiber hat an einem konkreten Ergebnis eines durch Nutzer eingesetzten KI-Chatbots keinen eigenen schöpferisch-kreativen Anteil. Er kann sich ebenfalls nicht auf das Urheberrecht stützen. Denkbar wäre allenfalls, dass der Gesetzgeber ein Leistungsschutzrecht für KI-generierte Inhalte schafft. Durch den Hype rund um ChatGPT und Co. könnte die bisher eher schwache Diskussion hierüber an Fahrt gewinnen. Ein Ergebnis ist derzeit nicht absehbar.

Ein KI-Entwickler kann sich allerdings auf den Urheberrechtsschutz für Computerprogramme oder das gesetzliche Leistungsschutzrecht für Datenbankhersteller berufen. Sie schützen ihn vor einer unerlaubten Übernahme der konkreten KI-Umsetzung. Daher kann er Geld für die Nutzung „seiner KI“ verlangen und ein entsprechendes Geschäftsmodell aufsetzen. OpenAI probiert dies derzeit mit ChatGPT aus. Eine Variante von ChatGPT wurde auch in die Microsoft-Suchmaschine Bing eingebunden, wofür Microsoft mehrere Milliarden US-Dollar gezahlt hat.

Wem gehören KI-generierte Inhalte?

Vorsicht ist aus urheberrechtlicher Sicht bei der Nutzung KI-generierter Inhalte geboten. Soweit diese auf urheberrechtlich geschützten Vorlagen beruhen und es sich nicht um gemeinfreie oder etwa auf Basis von Open-Source-Lizenzen für solche Zwecke nutzbare Werke handelt, kann man bei deren Nutzung durch etwaige Urheber in Anspruch genommen werden. OpenAI verweist darauf, dass ChatGPT ausschließlich Werke verwendet, bei denen keine Rechte Dritter entgegenstehen. Überprüfbar ist das im Einzelfall nicht. Den Nutzer eines urheberrechtsverletzenden Chatbot-Ergebnisses schützt seine Unwissenheit im Ernstfall

nicht. Im Urheberrecht existiert kein Schutz des guten Glaubens.



Sind urheberrechtliche Inhalte nicht gemeinfrei oder freinutzbar, stellt sich die Frage nach anderen Rechtfertigungen zur Verwendung von Werken im Rahmen des Text und Data Mining für KI-Zwecke. § 44 b des Urhebervertragsgesetzes (UrhG) versteht hierunter „die automatisierte Analyse von einzelnen oder mehreren digitalen oder digitalisierten Werken, um daraus Informationen über Muster, Trends und Korrelationen zu gewinnen“.

Eine Schranke des Urheberrechts enthält § 60 d des Urhebervertragsgesetzes für den Bereich des Text und Data Mining, sie erlaubt die automatisierte Auswertung – allerdings nur für die nicht kommerzielle wissenschaftliche Forschung. Dabei darf es auch keine beeinflussende Zusammenarbeit zwischen dem Forschungsinstitut und einem privaten Unternehmen geben.

§ 44 b UrhG besagt: „Zulässig sind Vervielfältigungen von rechtmäßig zugänglichen Werken für das Text und Data Mining.“ Dieser vermeintliche Freibrief gilt nicht unbeschränkt. Werke dürfen danach nur dann für Data Mining genutzt werden, „wenn der Rechtsinhaber sich diese nicht vorbehalten hat“. Bei online zugänglichen Werken muss ein solcher Nutzungsvorbehalt in maschinenlesbarer Form vorliegen.

Festlegen, was ein Webcrawler darf

Hier kommt der Robot Exclusion Standard ins Spiel. Ist im Stammverzeichnis einer Domain eine robot.txt-Datei hinterlegt, sollten Webcrawler diese zunächst auslesen. Der Domain-Inhaber kann darin festlegen, welche Suchmaschinen welche Inhalte unter einer Domain auslesen dürfen. Dieser Standard ist rechtlich möglicherweise als „Stand der Technik“ bindend. Er wird von den meisten Suchmaschinenbetreibern beachtet. Derzeit wird eine ähnliche Lösung mit dem Attribut „No AI“ diskutiert, um Webseiteninhalte vor der Nutzung durch KI-Systeme zu schützen.

Der Vollständigkeit halber stellt sich außerdem die Frage, ob dem KI-Nutzer nicht ein Urheberrechtsschutz an seinen Anweisungen an die KI zustehen kann. Es kommt beim sogenannten Prompt Engineering auf den Einzelfall an. Einfache Aufgabenstellungen fallen sicher nicht darunter, komplexe und individuelle Anweisungen möglicherweise schon. Urheberrechtlicher Schutz kann schließlich entstehen, wenn ein KI-Inhalt durch einen Menschen in ausreichender Schöpfungshöhe bearbeitet wird. Dann entsteht an den Bearbeitungen Urheberrechtsschutz.

Zwischen dem KI-Anbieter und dem KI-Nutzer besteht in der Regel ein Vertrag. Dieser enthält Regelungen über das Rechtsverhältnis zwischen beiden und auch im Hinblick auf die von der KI erzeugten Ergebnisse. An diesen erhält der KI-Nutzer meist sämtliche Rechte zur weiteren Nutzung. Ob daran ein Urheberrecht entsteht oder nicht, kann der Vertrag aber nicht regeln. Ein Urheberrecht entsteht kraft Gesetzes, nicht kraft eines Vertrages.

Wenn eine KI beispielsweise einen Artikel für die iX schreibt, hat niemand an dem Chatbot-generierten Text das Urheberrecht. Im Text können aber Werke Dritter verarbeitet worden sein, wodurch das Urheberrecht dieser Dritten tangiert wurde. Ist das nicht der Fall, ist der Text urheberrechtsfrei und kann

von jedermann genutzt werden. Einem Abdruck in der iX steht nichts im Wege.

Auch KI-Haftung soll reguliert werden

Unterdessen arbeitet die EU an einer Richtlinie zur KI-Haftung. Als Richtlinie ist sie nach ihrer Verabschiedung in nationales Recht umzusetzen. Sie soll Geschädigten die Beweislast erleichtern. So soll eine Kausalitätsvermutung greifen und ein Anspruch auf Schadenersatz schon dann bestehen, wenn eine Person für eine bestimmte für den Schaden relevante Verpflichtung verantwortlich war und „ein ursächlicher Zusammenhang mit der KI-Leistung nach vernünftigem Ermessen wahrscheinlich ist“.

Bei Hochrisiko-KI-Systemen sollen Richter zudem die Offenlegung von Informationen über solche Systeme anordnen können. Damit will man die verantwortlichen Personen ermitteln. Andererseits unterliegen offengelegte Informationen dem Geheimnisschutz. Auswirkungen auf den Bereich KI könnte auch die gleichfalls auf EU-Ebene diskutierte neue Produkthaftungsrichtlinie haben. Sie bietet Verbrauchern bei fehlerhaften Produkten eine verschuldensunabhängige Schadenshaftung. Mit ihr soll die seit 1985 EU-weit geregelte Produkthaftung an neue Technologien, etwa KI, angepasst werden.

Auf die KI-Anbieter kommen mehr und mehr regulatorische Pflichten und Complianceanforderungen zu. Deshalb müssen sie sich um ihre Haftungsrisiken Gedanken machen und entsprechende Vorkehrungen treffen. Das Trainieren von KI-Systemen auf Rechtskonformität ist eine der großen Herausforderungen beim Einsatz von KI. Dazu müssen aber zunächst einmal die gesetzlichen Vorgaben verfeinert werden. Sie werden aber auch nur den Rahmen festlegen. Die konkreten Ausgestaltungen hängen dann vom jeweiligen Einsatz von KI in der Praxis ab. Letztlich werden Gerichte darüber befinden, ob die Vorgaben in ausreichendem Maße eingehalten wurden oder nicht.

Auch KI-Nutzer müssen vor der Verwendung von KI-generierten Informationen juristische Aspekte beachten. Sie können sich im Bereich Datenschutz oder Urheberrecht in der Regel nicht auf ihre Unwissenheit berufen. Juristische Fallen entstehen, wenn Nutzer die Herkunft der durch KI verwendeten Datensätze nicht kennen und nicht kontrollieren können. Gleiches gilt für die eingesetzten Algorithmen. Zumindest bei KI-Anwendungen von Drittanbietern ist eine Kontrolle in den meisten Fällen nicht möglich.

Fazit

Künstliche Intelligenz fordert das Recht heraus. Je tiefer die KI in die verschiedenen Lebensbereiche eindringt, umso mehr müssen sich Gesetzgeber und Richter mit den daraus resultierenden Risiken befassen. Letztlich können davon zunächst fernliegende Rechtsbereiche betroffen sein, etwa das Schulrecht bei der Frage der Zulässigkeit KI-generierter Hausaufgaben. Es geht aber eben auch um komplexe Fragen im Haftungs-, Datenschutz- oder Urheberrecht (siehe Kasten „KI-Chatbots und das Urheberrecht“). Die EU plant mit dem AI Act einen risikobasierten Ansatz bei der KI-Regulierung bis hin zum Verbot einzelner KI-Ansätze. Auf Anbieter wie Nutzer kommen umfassende juristische Risiken zu. Das KI-Recht ist gekommen, um zu bleiben. (ur@ix.de)

ChatGPT im juristischen Einsatz

Künstliche Intelligenz ist nicht nur eine Herausforderung für die Juristerei. Sie kann ihr auch bei der Arbeitsbewältigung helfen. Richter, Staatsanwälte und Rechtsanwälte können von sprachmodellbasierten KI-Lösungen profitieren, die beispielsweise Urteile analysieren und Entscheidungsmuster aufzeigen. Hilfreich kann es auch sein zu wissen, wie ein bestimmtes Gericht in einem ähnlich gelagerten Verfahren entschieden hat. Das wäre ein großer Schritt hin zu einem „Predictive Decision-Making“, einem planbaren Ausgang eines Rechtsstreits.

Macht Chat-GPT Anwälte obsolet? Chatbots können dabei helfen, rechtssuchenden Bürgern den Gang zum Anwalt zu ersparen. Auch bisher war es möglich, zu Rechtsfragen zu „googeln“. Zukünftig könnte eine individuelle Rechtsberatung durch KI-Systeme hinzukommen. Sie wäre womöglich billiger als das Hinzuziehen eines Rechtsanwalts. Allerdings ist die Juristerei extrem komplex und jeder Einzelfall ist individuell zu betrachten. Dabei spielt der verfassungsrechtlich bei nahezu allen Rechtsfragen zu berücksichtigende Verhältnismäßigkeitsgrundsatz eine bedeutende Rolle. Er könnte eine KI – noch – an ihre Grenzen führen.

KI im juristischen Einsatz

Bereits heute werden KI-Systeme bei der Vertragsprüfung eingesetzt. Sie können Muster erkennen und beispielsweise auf Regelungslücken hinweisen. Derzeit werden solche Systeme etwa zur Prüfung von Vertraulichkeitsvereinbarungen eingesetzt. Auch an komplexere Verträge wagen sich die ersten Lösungen heran.

Weitere Anwendungsfelder für KI im Rechtsbereich gibt es bei Themen wie Contract Lifecycle Management. Im Bereich der DSGVO-Compliance können Webseiten auf die Einhaltung datenschutzrechtlicher Vorgaben etwa hinsichtlich der Datenschutzerklärung überprüft werden.

Welche weiteren Anwendungsfälle in Zukunft erlaubt sein werden, hängt von den regulatorischen Rahmenbedingungen ab. Denkbar sind KI-Systeme, die bei Gesetzesverstößen automatisch Sanktionen verhängen. Bei Parkzeitüberschreitungen ist das sicher eher akzeptabel als bei schwerwiegenden Straftaten. Hier dürfte Artikel 101 des Grundgesetzes eine Rolle spielen: „Niemand darf seinem gesetzlichen Richter entzogen werden.“ Dabei muss es sich (noch) um einen Menschen handeln.

1. Quellen

2. Tobias Haar; Nutzer unbekannt; Rechtsfragen der Pseudonymisierung und Anonymisierung; iX 5/2021, S. 86



Tobias Haar

ist Rechtsanwalt mit Schwerpunkt IT-Recht bei Vogel & Partner in Karlsruhe. Er hat zudem Rechtsinformatik studiert und hält einen MBA.

So programmiert es sich mit Sprach-KI

Sprach-KI soll Programmierern Routineaufgaben abnehmen und Codezeilen beim Schreiben sinnvoll ergänzen. Am Beispiel von GitHub Copilot lässt sich zeigen, wie sich diese Systeme einsetzen lassen und was dabei zu beachten ist. Denn noch darf man sich nicht blind auf die Hilfe verlassen.

Von Philipp Braunhart

-tract

- Der KI-Assistent Copilot von GitHub soll beim Entwickeln helfen und repetitive Standardaufgaben automatisieren.
- Dafür lässt sich der Assistent per Plug-in in gängige IDEs einbinden.
- Copilot berücksichtigt die aktuelle Eingabe und die

umliegenden Zeilen, um Entwicklern passende Codevorschläge zu bieten.

- Der Einsatz klappt am besten, wenn man sich das Programm als einen zusätzlichen Entwickler vorstellt, der helfen soll und dementsprechend Informationen benötigt.

„Die neue, heißeste Programmiersprache ist Englisch“, das meinte Andrej Karpathy, der ehemalige Head of AI von Tesla, Anfang des Jahres auf Twitter. Dass er damit nicht ganz falschliegt, zeigt eine Vielzahl von aktuellen Programmierassistenten, die auf generativer künstlicher Intelligenz basieren. Einer davon ist GitHub Copilot, der im Kontext des aktuellen Inhalts meist sehr genau versteht, was gerade gewünscht ist, und sehr selten komplett danebenliegt. Hinter Copilot steckt das KI-Modell Codex von OpenAI, das natürliche Sprache und Code verstehen und generieren kann.

Technischer Hintergrund

GitHub Copilot basiert auf Codex, einer weitertrainierten Variante des Large Language Models GPT-3 von OpenAI. Das Modell verwendet die Transformerarchitektur. Durch mehrere Terabyte an Textdaten unter anderem aus den Archiven von Common Crawl und WebText2, Büchern und der englischen Wikipedia hat OpenAI das Modell auf das Verstehen von Sprache trainiert. Die Feintuning von Codex basiert auf Milliarden von Codezeilen von öffentlichen GitHub-Repositoryn. GitHub Copilot filtert und bewertet die Vorschläge von Codex, bevor es diese an die Nutzer sendet.

Copilot erzeugt sowohl Code als auch Kommentare, wobei das Programm den Kontext des zu erzeugenden Codes zur Vorhersage verwendet. Dies ist der Inhalt der Datei mit Fokus auf die aktuelle Zeile und die darüberliegenden. Die Vorschläge können innerhalb einer Zeile sein, eine komplette oder sogar mehrere Zeilen umfassen. Damit lässt sich das Programmieren stark beschleunigen.

Die Integration in verschiedene IDEs erfolgt durch Plug-ins. Copilot unterstützt nahezu sämtliche Programmiersprachen, am besten sind die Ergebnisse für Python, JavaScript, TypeScript, Ruby, Go, C#, C++ und Java. Auch Sprachen wie SQL, HTML und YAML beherrscht Copilot. Dank dieser Vielfalt ist er nahezu für alle Entwickler eine Hilfe.

Benutzer können sich auf ihre Aufgabe konzentrieren, ohne zwischen der IDE und Webseiten zur Informationssuche wie Stack Overflow zu wechseln. Das kommt gut an: Während der Assistent nach dem Release 27 Prozent der Inhalte von mit ihm editierten Dateien generierte, kommen laut Plattformbetreiber nun 46 Prozent der Files von Copilot, im Fall von Java sogar 61 Prozent, Stand Februar 2023.

Fliegen mit Copilot

Copilot generiert Codevorschläge direkt während des Schreibens. Die Angebote der KI lassen sich per Klick oder Tastendruck annehmen oder ablehnen. Die Schaltflächen über dem Vorschlagsfeld bieten die Möglichkeit, mehrere Vorschläge zu durchsuchen oder sogar ein neues Fenster mit bis zu 10 Entwürfen zu öffnen (siehe Abbildung 1). Eine Alternative besteht darin, Kommentare zu schreiben, um gewünschten Code generieren zu lassen. Dann zeigt Copilot die Codevorschläge in der Zeile unter dem Kommentar an (siehe Abbildung 2).

```
import numpy as np
```



Einmal in die IDE eingebunden, macht Copilot Vorschläge zum Vervollständigen von angefangenen Codezeilen, hier am Beispiel eines Arrays (Abb. 1).

```
import numpy as np
```

```
# generate an array x with 1000 values from 0 to 10  
x = np.linspace(0, 10, 1000)
```

Copilot lässt sich auch per Kommentar steuern. Hier folgt der KI-Assistent dem Befehl zum Erstellen eines Arrays mit 1000 Werten (Abb. 2).

Das System erstellt nicht nur einzelne Codezeilen: Aus detaillierten Kommentaren oder begonnenen Funktionsdefinitionen mit Input und Output generiert Copilot Codeblöcke. Auch Docstrings erstellt der KI-Assistent. Abbildung 3 zeigt, wie Copilot bereits bei jedem Schritt vom Schreiben der Kommentare bis hin zur Input- und Output-Definition hilft.

```
import pandas as pd  
import numpy as np  
import plotly.express as px
```

```
# generate a dataframe with x from 0 to 10  
# and y1 sin, y2 cos  
x = np.arange(0, 10, 0.01)  
df = pd.DataFrame({  
    'x': x,  
    'y1': np.sin(x),  
    'y2': np.cos(x),  
})
```

```
# create a function to plot the above dataframe  
# with plotly express
```

```
def plot(df):  
    fig = px.line(df, x='x', y=['y1', 'y2'])  
    fig.show()
```

Nichts mehr selbst schreiben: Nur über Kommentare hat Copilot hier den gesamten Code generiert. Dabei lassen sich auch eingebundene Bibliotheken ansteuern (Abb. 3).

Es ist jedoch wichtig zu beachten, dass dabei durchaus Fehler passieren oder das System suboptimalen Code produzieren kann. Daher ist es notwendig, die Vorschläge vor dem Annehmen immer sorgfältig zu überprüfen.

Auch die Qualität der Dokumentation im Code lässt sich so leicht verbessern: Einfach zusätzliche Kommentare und Dokumentation auf Basis des Codekontextes von Copilot generieren lassen. Dabei erkennt und imitiert der Assistent den persönlichen Stil im Code und in den Kommentaren.

Tipps für eine sichere Landung

Ein entscheidender Faktor für die optimale Nutzung von Copilot ist das Bereitstellen von ausreichenden Informationen, um relevante und präzise Vorschläge zu erhalten. Das ist vergleichbar mit einem anderen Programmierer, der auf entsprechende Informationen und Anleitung angewiesen ist. Die wichtigste Frage beim Schreiben von Kommentaren oder Docstrings ist also: Welche Informationen benötigt ein Programmierer, um die Anforderungen zu erfüllen? Je detaillierter die Angaben sind, desto besser kann Copilot weiterhelfen.

Ein weiterer wesentlicher Aspekt ist sauberer und konsistenter Code. Hierbei sollte man gute, klare und präzise Kommentare, Docstrings und aussagekräftige Namen für Funktionen, Klassen, Variablen und Argumente verwenden. Logik und die Intention des Codes sollten leicht zugänglich sein. Copilot kann dabei sogar unterstützen und generiert dadurch anschließend bessere Codevorschläge.

Soll Copilot Funktionen aus Bibliotheken vorschlagen, reicht es meist, Letztere am Anfang der Datei zu importieren. Zusätzlich hilft es, die Bibliothek in einem Kommentar dort zu

erwähnen, wo man sie einsetzen will. Bei der Arbeit mit Datenbanken oder Dataframes ist es sinnvoll, das Schema, wie Spaltennamen, Typen und einige Beispiele, als Kommentare bereitzustellen. Dann versteht Copilot die verwendeten Objekte genauer und macht Vorschläge mit den passenden Namen (siehe Abbildung 4).

```
import pandas as pd
import matplotlib.pyplot as plt

# load data
# DataFrame has columns: ['id', 'name', 'age', 'gender', 'height', 'weight']
pd.read_csv(path)

# histogram
plt.hist(data['height'], bins=10)
```

Informiert man die Sprach-KI per Kommentar über die Namen von Spalten eines Dataframes, dann schlägt das Programm selbstständig mögliche Felder beim Plotten der Daten vor (Abb. 4).

Um Copilot effizient zu nutzen, empfiehlt es sich, Tastenkombinationen in der IDE einzurichten. Geeignete Funktionen für Hotkeys sind: Vorschläge annehmen, ablehnen, nur das nächste Wort des Vorschlags akzeptieren, zwischen verschiedenen Optionen navigieren oder ein separates Fenster mit Vorschlägen öffnen. Dadurch reduziert Copilot die Zeit für repetitive oder mühsame Aufgaben, wie das Schreiben von Unit-Tests, das Behandeln von Exceptions, das Loggen oder das Ausgeben von Informationen.

Tricks für Überflieger

Es gibt auch noch weitere spannende Aspekte an GitHub Copilot jenseits des Vervollständigens von Code. Einer dieser Aspekte ist das Konvertieren von Code in andere Programmiersprachen. Hierdurch lässt sich etwa Legacy-Code leichter portieren (siehe Abbildung 5). Darüber hinaus hilft Copilot auch beim Umwandeln von Code von einem Framework in ein anderes. Das beschleunigt das Konvertieren zwischen verschiedenen Frameworks (siehe Abbildung 6).

```
# Convert this from Python to R
# Python version

import numpy as np
import matplotlib.pyplot as plt

x = np.random.randn(1000)
plt.hist(x, bins=20)
plt.show()

# End

# R version

x <- rnorm(1000)      # <- generated by Copilot
hist(x, breaks=20)   # <- generated by Copilot
```

Will man seinen Code von einer Programmiersprache in eine andere übersetzen, kann eine Sprach-KI helfen. Das Beispiel zeigt das Erstellen und Plotten von 1000 Beispieldaten in Python und in R (Abb. 5).

```

# Rewrite this to use plotly express instead of matplotlib

import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 2 * np.pi, 100)
y = np.sin(x)

plt.plot(x, y)
plt.show()

# plotly express version:

import numpy as np                                # ← generated by Copilot
import plotly.express as px                       # ← generated by Copilot

x = np.linspace(0, 2 * np.pi, 100)              # ← generated by Copilot
y = np.sin(x)                                     # ← generated by Copilot

fig = px.line(x=x, y=y)                          # ← generated by Copilot
fig.show()                                       # ← generated by Copilot

```

Ähnlich wie beim Übersetzen von Code in unterschiedliche Programmiersprachen dolmetscht Copilot auch zwischen verschiedenen Frameworks und wie hier verschiedenen Bibliotheken einer Sprache (Abb. 6).

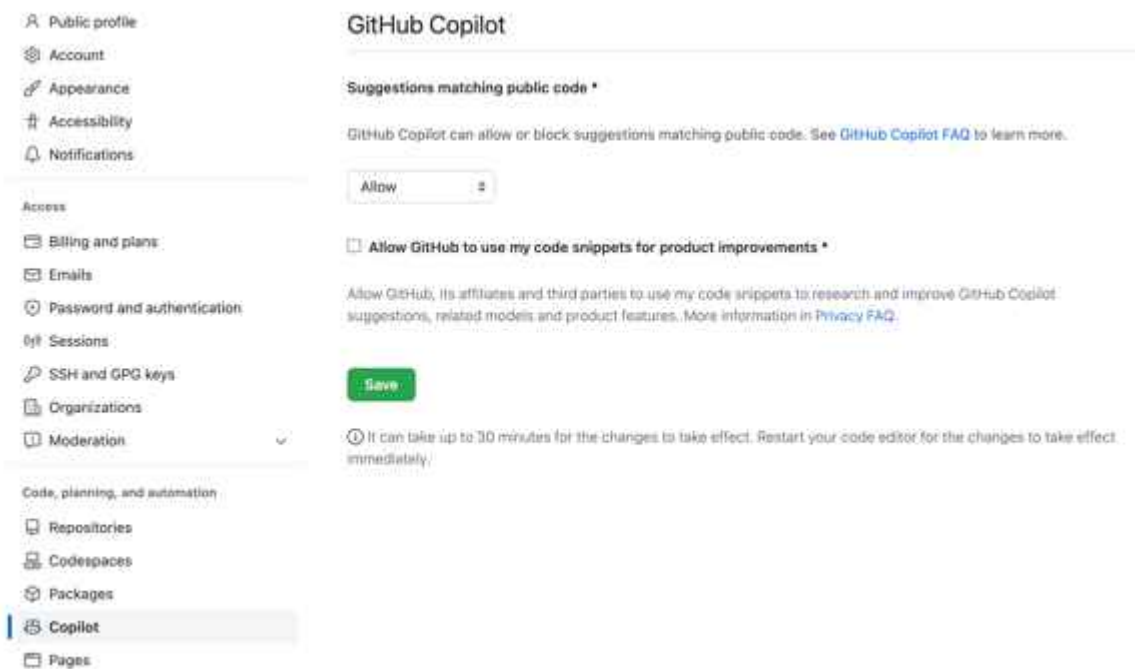
Wenn es darum geht, neue Frameworks oder Bibliotheken zu lernen, ist Copilot eine große Hilfe. Vorschläge als Kommentar, basierend auf dem Wunsch des Users, sind ein guter Ausgangspunkt. Damit können Entwickler ihr Wissen schnell erweitern und erfolgreich anwenden.

Zuletzt kann man Copilot auch für das schnelle Generieren von Daten einsetzen. Etwa wenn Daten für Tests nötig sind, reicht die Angabe von ein paar Beispielen und Copilot generiert automatisch zahlreiche weitere Daten.

Turbulenzen

Trotz seines Nutzens weist das Tool jedoch auch Grenzen und potenzielle Risiken auf, die Anwender berücksichtigen sollten. So kann es geschehen, dass Copilot exakte Kopien von Code aus

öffentlichen Repositorys oder Quellen vorschlägt. Hierfür lässt sich eine Filteroption auf der Einstellungsseite aktivieren (siehe Abbildung 7). Dort sieht man auch, dass GitHub den Code von Anwendern standardmäßig für das Retrainieren der KI-Modelle hinter Copilot nutzt und somit potenziell mit anderen Nutzern teilt. Auch dies lässt sich in den Einstellungen deaktivieren.



Zwei besonders wichtige Optionen finden sich im Menü von Copilot. Einerseits lässt sich verhindern, dass der Assistent Code aus öffentlichen Repositorien exakt repliziert. Andererseits kann man per Checkbox verhindern, dass GitHub den gescannten und erzeugten Code zum Trainieren des Modells verwendet und man so möglicherweise Betriebsgeheimnisse verrät (Abb. 7).

Durch das Teilen von eigenem Code können Risiken für sensible Daten entstehen. GitHub versichert, dass die hohen Standards für Datenschutz und Datensicherheit auf ihrer Plattform eingehalten werden und der Code nur zum Erstellen von Vorschlägen genutzt wird. Noch mehr Kontrolle geben Self-Hosting-Alternativen für KI-Codeassistenten (siehe unten).

Generell ist der rechtliche Status der Trainingsdaten von Copilots KI-Modellen noch nicht komplett geklärt und Inhalt einiger aktuell laufender Klagen. OpenAI hat zahlreiche

GitHub-Repositoryys, die unter Copyleft-Lizenzen wie der GPL stehen, für das Training verwendet. Copilots Modell weist nicht auf die Lizenzen des eingebauten Codes hin und ist selbst nicht Open Source. Hier bleibt die Rechtsprechung abzuwarten.

Das momentan limitierte Kontextfenster ist eine andere Einschränkung. Der Assistent sollte eigentlich lediglich den Inhalt der aktuell bearbeiteten Datei berücksichtigen. Teilweise entsteht bei der Arbeit allerdings der Eindruck, dass Copilot auch andere offene Dateien heranzieht. Allerdings verwendet das System nicht die gesamte Codebasis eines Projekts.

Unsicherer Code und Schwachstellen stellen ebenfalls eine Herausforderung dar. Eine aktuelle Forschungsarbeit deutet darauf hin, dass Nutzer von Tools wie Copilot dazu tendieren, Code mit Schwachstellen leichter zu akzeptieren, sich aber gleichzeitig in Sicherheit wiegen (siehe ix.de/zxwq). Mit einem Copilot-Update im Februar 2023 führte GitHub ein Feature ein, das potenzielle Schwachstellen in den Vorschlägen scannt und automatisch unterdrückt. Das System filtert dabei typische Schwachstellen wie hartcodierte Credentials, SQL- oder Pfadinjektionen.

Einige Probleme sind damit sicherlich behoben, aber natürlich findet das Programm dadurch nicht alle Schwachstellen. Es bedarf also weiterhin großer Aufmerksamkeit, gerade bei sicherheitsrelevanten Anwendungen. Zuletzt sei erwähnt, dass Copilot am besten auf Englisch arbeitet. Arbeiten Anwender mit anderen Sprachen, kann das zu Einschränkungen führen.

Alternative Reisemöglichkeiten

Zu Copilot gibt es diverse Alternativen, darunter sowohl kommerzielle als auch Open Source. AWS bietet beispielsweise Code Whisperer an, Salesforce mit CodeGen ein quelloffenes Projekt, das sich mit Fauxpilot selbst hosten lässt.

Weitere Optionen sind Tabnine und Replit Ghostwriter. Für JupyterLab, Colab-Notebooks oder BigQuery ist zudem CodeSquire als Browsererweiterung verfügbar. Der Markt entwickelt sich stetig und so kommen und gehen die Alternativen. Unter allen Optionen scheint GitHub Copilot derzeit am ausgereiftesten und zuverlässigsten zu sein.

Was klar ist: Copilot ist mehr als nur ein einfaches Autocomplete-Werkzeug. Es ist der Anfang eines größeren Wandels im Programmieren, der darauf abzielt, mehr Aspekte des Entwicklungsprozesses zu automatisieren.

Die Fähigkeit, Code aus Kommentaren und Beispielen innerhalb der eigenen IDE zu generieren, reduziert Ablenkungen, beschleunigt das Programmieren und verbessert die Qualität der Dokumentation. Für den richtigen Einsatz und gute Vorschläge sollten Entwickler im Kopf behalten, dass das System die gleichen Informationen benötigt wie ein Kollege, den man um Rat bittet. Forscher bei Google unterstreichen die Vorteile von Programmierassistenten. In einer Untersuchung mit der Inhouse-KI will man herausgefunden haben, dass sich die Produktivität mithilfe des Tools messbar erhöhen lässt. (pst@ix.de)

1. Quellen
2. [Weitere Informationen zu Copilot und dem Programmieren mit KI-Assistenten finden sich unter `ix.de/zxwq`.](#)



Philipp Braunhart

ist Entwickler und Berater im Bereich Machine Learning und Mitgründer der ingenio ai GmbH. Er programmiert, entwirft und

baut KI-Lösungen für den direkten Einsatz in den Bereichen Bio-, Agrar- und Medizintechnologie.

ChatGPT per API einbinden

Ein Programmbeispiel zeigt, wie man ohne Overhead und Umwege die Modelle hinter ChatGPT per OpenAI-API in eigenen C#-Anwendungen nutzt.

Von Daniel Basler

-tract

- Mit der OpenAI-API lässt sich das ChatGPT-Modell gpt-3.5-turbo in eigene Programme einbinden.
- Anstelle von Token aus unstrukturiertem Text verwendet ChatGPT eine Folge von Nachrichten zusammen mit Metadaten. OpenAI nennt das Format Chat Markup Language (ChatML).
- Mit einer in C# implementierten WPF-App können Interessierte erste Erfahrungen mit dem Einbinden der Sprach-KI in eigene Anwendungen sammeln.
- Durch Zugriff auf verschiedene Modellparameter bietet die API die Möglichkeit, den Umgang mit Sprachmodellen zu erlernen und zu verbessern.

Mit ChatGPT (OpenAI) und Bard beziehungsweise LaMDA von Google ist die künstliche Intelligenz (KI) massenkompatibel geworden. ChatGPT hat sich auf Grundlage von Transformertechnologie und dem Attention-Mechanismus mit Millionen von Textseiten vollgesogen und daraus ein großes Modell der menschlichen Sprache gebaut: ein Large Language Model (LLM).

Dadurch kann ChatGPT eine Vielzahl von Aufgaben der natürlichen Sprachverarbeitung ausführen, darunter das Generieren, Zusammenfassen und Klassifizieren von Text und Frage-Antwort-Systeme. Des Weiteren erstellt ChatGPT Programme und hilft beim Debuggen von Code. All diese Aufgaben beherrschen auch andere Modelle aus der GPT-Reihe von OpenAI in unterschiedlichem Ausmaß. ChatGPT selbst ist für den Dialog mit Nutzern optimiert, basiert aber ansonsten auf der Generation GPT 3.5 und damit den leistungsfähigsten und neuesten Modellen, die OpenAI derzeit im Angebot hat. OpenAI macht seine GPT-Modelle über eine einheitliche API (Application Programming Interface) verfügbar, über die sich mit diesen Modellen kommunizieren lässt. Damit ist es also möglich, eigene Programme mit den Fähigkeiten von ChatGPT auszustatten, seit Anfang März ist auch der Zugriff auf das ChatGPT-Modell über die API möglich. Unser Beispiel zeigt, wie sich die API und die Modelle von OpenAI mit C# ansprechen lassen. Über die API lassen sich die Modelle auch über unterschiedliche Parameter für verschiedene Zwecke optimieren.

Softwareautomatisierung mit generativer KI

Bei ChatGPT und den anderen GPT-Modellen von OpenAI handelt es sich um generative KI-Stacks. Hierbei kommen Deep- und Machine-Learning-Algorithmen sowie Generative Adversarial Networks (GAN) zum Einsatz, die aus vorhandenen Texten, Audiodateien oder Bildern Inhalte erstellen. Die Modelle Erlernen Muster in der Eingabe und verwenden sie, um ähnliche Inhalte zu erzeugen.

Transformermodelle wie GPT oder LaMDA simulieren kognitive Prozesse und versuchen die Bedeutung der Eingabedaten auf unterschiedliche Weise zu messen. Man trainiert die Modelle darauf, die Sprache oder das Bild im Kontext zu verstehen, Klassifizierungen zu erlernen und Texte oder Bilder aus großen Datensätzen zu generieren.

Das verwendete Sprachmodell Generative Pre-trained Transformer, kurz GPT, kann Befehle in natürlicher Sprache auch in Programmcode übersetzen. GPT ist also in der Lage, Websites oder Codebeispiele über automatisches Codegenerieren zu erzeugen. Das aus GPT entstandene Modell Codex ist auf Codegenerierung optimiert. GitHubs Tool Copilot liegt das Codex-Modell zugrunde. Copilot analysiert Kommentare aus dem Code und schlägt einzelne Codezeilen oder Funktionen während des Entwickelns vor. Das am 1. März für die API veröffentlichte gpt-3.5-turbo-Modell, auf dem ChatGPT basiert, ist für die Ein- und Ausgabe von Konversationschats optimiert. Des Weiteren ist die Turbo-Modellfamilie auch die erste, die regelmäßige Modellupdates erhält.

Das Projekt

Das Programmierbeispiel zeigt, wie man mit der OpenAI-API auf GPT-3-Modelle zugreift. Dort hat man Zugriff auf gpt-3.5-turbo. Das Projekt Example-ChatGPTApplication ist überschaubar und greift per HTTP und JSON auf den API-Endpunkt zu. Zwar ist ein Zugriff auch über die Bibliothek .NET SDK for OpenAI GPT-3 möglich, die Library kommt jedoch mit einem viel größeren Funktionsumfang, den man für das Beispiel hier nicht benötigt. Das Programm kommt als Windows-Presentation-Foundation-App (WPF) mit einer einfachen Benutzeroberfläche für Anfragen, Modellauswahl und Einstellungen für Parameter daher. Man erstellt die Anwendung mit der Visual Studio Community Edition 2022 und dem .NET Framework 7.0. Um schlank zu bleiben, verwendet das Beispiel Code aus einer Code-Behind-Datei der Klasse MainWindow.xaml.cs.

Zum Verwenden der API benötigt man einen aktiven Account bei OpenAI, über den sich ein Authentifikationsschlüssel für die Verwendung der API erstellen lässt. OpenAI berechnet zurzeit 0,02 US-Dollar für 1000 Eingabetoken bei den bisherigen GPT-Modellen. Bei ChatGPT belaufen sich die Kosten auf 0,002 US-Dollar pro 1000 Token. Token stellen Teile von Wörtern dar,

wobei 1000 Token etwa 750 Wörtern entsprechen. Beim Anmelden erhält man in der Regel ein Startguthaben von etwa 18 US-Dollar, das sich in den ersten drei Monaten frei verwenden lässt. Ist es aufgebraucht, benötigt man einen neuen API-Key, den OpenAI in Rechnung stellt.

Die Oberfläche

Das Programm verwendet die Projektvorlage WPF Application mit der Projektbezeichnung ExampleChatGPTApplication und dem .NET Framework 7.0. Visual Studio erstellt aus der Vorlage automatisch eine bereits lauffähige WPF-Applikation. Listing 1 zeigt die Erweiterung der Oberfläche in der XAML-Datei MainWindow.xaml. Das GUI lässt sich nach eigenem Geschmack erstellen oder später anpassen.

Listing 1: Aufbau der Oberfläche in der MainWindow.xaml-Datei

```
<Window x:Class="ExampleChatGPTApplication.MainWindow"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:local="clr-namespace:ExampleChatGPTApplication"
mc:Ignorable="d"
WindowStartupLocation="CenterScreen"
Title="MainWindow" Height="500" Width="720"
Loaded="Window_Loaded">
  <Grid Margin="0,0,10,0">
    <TextBox Height="396" HorizontalAlignment="Left"
Margin="11,10,0,0" Name="chatBox" VerticalAlignment="Top"
Width="535"
IsReadOnly="True"
TextWrapping="WrapWithOverflow"
VerticalScrollBarVisibility="Visible"/>
    <Label Content="Message:" Height="28"
HorizontalAlignment="Left" Margin="11,411,0,0" Name="label5"
```

```
VerticalAlignment="Top" />
    <TextBox Height="22" HorizontalAlignment="Left"
Margin="86,414,0,0" Name="messageText" VerticalAlignment="Top"
Width="460" />
        <Button Content="Send" Height="23"
HorizontalAlignment="Left" Margin="564,413,0,0"
Name="sendMessageButton" VerticalAlignment="Top" Width="117"
Click="sendMessageButton_Click"/>
    <ComboBox x:Name="comboBox" HorizontalAlignment="Left"
Margin="564,36,0,0" VerticalAlignment="Top" Width="120"
SelectionChanged="comboBox_SelectionChanged">
        <ComboBoxItem>text-davinci-002</ComboBoxItem>
        <ComboBoxItem IsSelected="True">text -
davinci-003</ComboBoxItem>
        <ComboBoxItem>code-davinci-002</ComboBoxItem>
        <ComboBoxItem>gpt-3.5-turbo</ComboBoxItem>
    </ComboBox>
        <Label x:Name="label" Content="Model:"
HorizontalAlignment="Left" Margin="564,10,0,0"
VerticalAlignment="Top" Width="70"/>
        <Label x:Name="label1" Content="Temperature"
HorizontalAlignment="Left" Margin="564,63,0,0"
VerticalAlignment="Top"/>
        <TextBox x:Name="textBox" HorizontalAlignment="Left"
Margin="564,94,0,0" TextWrapping="Wrap" Text="0.5"
VerticalAlignment="Top" Width="77"/>
        <Label x:Name="label2" Content="Presence Penalty"
HorizontalAlignment="Left" Margin="564,122,0,0"
VerticalAlignment="Top"/>
        <TextBox x:Name="textBox1" HorizontalAlignment="Left"
Margin="564,153,0,0" TextWrapping="Wrap" Text="0"
VerticalAlignment="Top" Width="77"/>
        <Label x:Name="label3" Content="Frequency Penalty"
HorizontalAlignment="Left" Margin="564,186,0,0"
VerticalAlignment="Top"/>
        <TextBox x:Name="textBox2" HorizontalAlignment="Left"
Margin="564,214,0,0" TextWrapping="Wrap" Text="0.5"
VerticalAlignment="Top" Width="77"/>
        <Label x:Name="label4" Content="TopP"
HorizontalAlignment="Left" Margin="566,239,0,0"
VerticalAlignment="Top"/>
```

```
        <TextBox x:Name="textBox3" HorizontalAlignment="Left"
Margin="564,265,0,0" TextWrapping="Wrap" Text="0.3"
VerticalAlignment="Top" Width="77"/>
    </Grid>
</Window>
```

Die ComboBox verfügt in der XAML-Datei über kein spezielles ItemTemplate oder DataTemplate. Die Auswahl des Modells erfolgt später im Code-Behind der Datei MainWindow.xaml. Über die Event-Handler-Methoden Loaded im Window und Click beim Send-Button lassen sich die Erweiterungen im Code-Behind vornehmen.

Programmlogik

Nach dem Vorbereiten der Oberfläche beginnt das Implementieren der Programmlogik. Der Code besteht aus fünf C#-Klassen:

- RequestChatGPT.cs für Anfragen an das Chat-Modell,
- ResponseChatGPT.cs für Antworten des Chat-Modells,
- RequestGPT.cs für Anfragen,
- ResponseGPT.cs für Antworten,
- ChatCompletionMessage.cs für Nachrichten im Chat-Modell.

In Visual Studio lassen sich die benötigten Klassen erstellen. Listing 2 und Listing 3 zeigen den Code für das Erstellen der RequestGPT- und der ResponseGPT-Klasse. Für ChatGPT müssen Entwickler die Request- und Response-Klasse für die Message-Methode (Nachrichten) erweitern. Listing 4 und 5 zeigen das Implementieren der Klassen RequestChatGPT und ResponseChatGPT. Die Klasse ChatCompletionMessage führt die Messages (Nachrichten) für die Chatvervollständigung ein, die nur gpt-3.5-turbo benötigt (siehe Listing 6).

Listing 2: Implementieren der Klasse RequestGPT

```
using System.Text.Json.Serialization;

namespace ExampleChatGPTApplication
```

```

{
    public class RequestGPT
    {
        public RequestGPT() { }

        [JsonPropertyName("model")]
        public string? Model { get; set; }

        [JsonPropertyName("prompt")]
        public string? Prompt { get; set; }

        [JsonPropertyName("max_tokens")]
        public int MaxTokens { get; set; }

        [JsonPropertyName("temperature")]
        public float Temperature { get; set; }

        [JsonPropertyName("top_p")]
        public float TopP { get; set; }

        [JsonPropertyName("presence_penalty")]
        public float PresencePenalty { get; set; }

        [JsonPropertyName("frequency_penalty")]
        public float FrequencyPenalty { get; set; }
    }
}

```

Listing 3: Implementieren der Klasse ResponseGPT

```

using System.Collections.Generic;
using System.Text.Json.Serialization;

namespace ExampleChatGPTApplication
{
    public class ResponseGPT
    {
        [JsonPropertyName("id")]
        public string? Id { get; set; }

        [JsonPropertyName("object")]
        public string? @Object { get; set; }
    }
}

```

```

    [JsonPropertyName("created")]
    public int Created { get; set; }

    [JsonPropertyName("model")]
    public string? Model { get; set; }

    [JsonPropertyName("choices")]
    public List<ChatGPTChoice>? Choices { get; set; }

    [JsonPropertyName("usage")]
    public ChatGPTUsage? Usage { get; set; }
}
}

```

```

public class ChatGPTUsage
{
    [JsonPropertyName("prompt_tokens")]
    public int PromptTokens { get; set; }

    [JsonPropertyName("completion_tokens")]
    public int CompletionTokens { get; set; }

    [JsonPropertyName("total_tokens")]
    public int TotalTokens { get; set; }
}

```

```

public class ChatGPTChoice
{
    [JsonPropertyName("text")]
    public string? Text { get; set; }

    [JsonPropertyName("index")]
    public int Index { get; set; }

    [JsonPropertyName("logprobs")]
    public object? LogProbabilities { get; set; }

    [JsonPropertyName("finish_reason")]
    public string? FinishReason { get; set; }
}

```

Listing 4: Ausschnitt aus der Implementierung der Klasse RequestChatGPT

```
using System.Collections.Generic;
using System.Text.Json.Serialization;

namespace ExampleChatGPTApplication
{
    public class RequestChatGPT
    {
        [JsonPropertyName("model")]
        public string Model { get; set; } = null!;

        [JsonPropertyName("messages")]
        public IList<ChatCompletionMessage> Messages { get;
set; } = new List<ChatCompletionMessage>();...
```

Listing 5: Implementieren der Klasse ResponseChatGPT

```
using System.Collections.Generic;
using System.Text.Json.Serialization;

namespace ExampleChatGPTApplication
{
    public class CreateChatResponse
    {
        [JsonPropertyName("id")]
        public string Id { get; set; } = null!;

        [JsonPropertyName("object")]
        public string Object { get; set; } = null!;

        [JsonPropertyName("created")]
        public int Created { get; set; }

        [JsonPropertyName("model")]
        public string Model { get; set; } = null!;

        [JsonPropertyName("choices")]
        public List<ChatCompletionChoice> Choices { get; set; }
```

```

} = new();

    [JsonPropertyName("usage")]
    public ChatCompletionUsage? Usage { get; set; }
}
public class ChatCompletionChoice
{
    [JsonPropertyName("delta")]
    public ChatCompletionMessage? Delta
    {
        get => Message;
        set => Message = value;
    }

    [JsonPropertyName("message")]
    public ChatCompletionMessage? Message { get; set; }

    [JsonPropertyName("index")]
    public int Index { get; set; }

    [JsonPropertyName("finish_reason")]
    public string FinishReason { get; set; } = null!;
}
public class ChatCompletionUsage
{
    [JsonPropertyName("prompt_tokens")]
    public int PromptTokens { get; set; }

    [JsonPropertyName("completion_tokens")]
    public int CompletionTokens { get; set; }

    [JsonPropertyName("total_tokens")]
    public int TotalTokens { get; set; }
}
}

```

Listing 6: Implementieren der Klasse ChatCompletionMessage

```

using System.Text.Json.Serialization;

namespace ExampleChatGPTApplication

```

```

{
    public class ChatCompletionMessage
    {
        public ChatCompletionMessage(string role, string
content)
        {
            Role = role;
            Content = content;
        }

        [JsonPropertyName("role")]
        public string Role { get; set; }

        [JsonPropertyName("content")]
        public string Content { get; set; }
    }
}

```

In der Request-Klasse bleiben die Eigenschaftsnamen der JSON-Ausgabe für die spätere HTTP-Anfrage über die API unverändert. Die Klasse liefert ein Request-Objekt, in dem sich das gewünschte Modell und Parameter festlegen lassen. Bei der Response-Klasse legt man auch die Eigenschaftsnamen als Attribute fest. Des Weiteren lassen sich hier die Klassen für das Verwenden der Token (ChatGPTUsage) und der Auswahl (ChatGPTChoice) beziehungsweise Message (ChatCompletionChoice) festlegen. In der MainWindow.xaml.cs implementiert man die Programmlogik zum Aufruf und zur Rückgabe über die OpenAI-API. Listing 7 zeigt die benötigten Methoden für den Zugriff auf die Sprachmodelle von OpenAI.

Listing 7: Implementieren der Zugriffslogik auf die API

```

using System;
using System.Collections.Generic;
using System.Globalization;
using System.Net.Http;
using System.Text;
using System.Text.Json;

```



```

        maxTokens = 2048;
        isChatGPTModel = false;
        break;
    case 1:
        modelName = "text-davinci-003";
        maxTokens = 4000;
        isChatGPTModel = false;
        break;
    case 2:
        modelName = "code-davinci-002";
        maxTokens = 2048;
        isChatGPTModel = false;
        break;
    case 3:
        modelName = "gpt-3.5-turbo";
        maxTokens = 4096;
        isChatGPTModel = true;
        break;
    default:
        modelName = "text-davinci-003";
        maxTokens = 4000;
        isChatGPTModel = false;
        break;
    }
}

```

```

    private void sendMessageButton_Click(object sender,
RoutedEventArgs e)
    {
        if(!isChatGPTModel)
        {
            CallGPTRequest();
        }
        else
        {
            CallChatGPTModel();
        }
    }

```

```

private async void CallGPTRequest()
{

```

```

RequestGPT completionReqGTP = new RequestGPT
{
    Model = modelName,
    Temperature = float.Parse(textBox.Text,
CultureInfo.InvariantCulture.NumberFormat),
    MaxTokens = maxTokens,
    TopP = float.Parse(textBox3.Text,
CultureInfo.InvariantCulture.NumberFormat),
    FrequencyPenalty = float.Parse(textBox1.Text,
CultureInfo.InvariantCulture.NumberFormat),
    PresencePenalty = float.Parse(textBox2.Text,
CultureInfo.InvariantCulture.NumberFormat)
};

using (HttpClient httpClient = new HttpClient())
{
    try
    {
        string? userResponse = messageText.Text;
        chatBox.Foreground = new
SolidColorBrush(Colors.Red);
        chatBox.Text = user + userResponse;
        completionReqGTP.Prompt = userResponse;
        ResponseGPT? responseGPT = null;

        using (HttpRequestMessage httpReq = new
HttpRequestMessage(HttpMethod.Post,
"https://api.openai.com/v1/completions"))
        {
            httpReq.Headers.Add("Authorization",
$"Bearer {OpenAI_ApiKey}");

            string requestString =
JsonSerializer.Serialize(completionReqGTP);
            httpReq.Content = new
StringContent(requestString, Encoding.UTF8,
"application/json");

            using (HttpResponseMessage?
httpResponse = await httpClient.SendAsync(httpReq))
            {
                if (httpResponse is not null)

```

```

        {
            string responseString = await
httpResponse.Content.ReadAsStringAsync();
                                                    if
(httpResponse.IsSuccessStatusCode
&&
!string.IsNullOrEmpty(responseString))
                {
                    responseGPT =
JsonSerializer.Deserialize<ResponseGPT>(responseString);
                }
            }
        }

        if (responseGPT != null)
        {
            string? responseText =
responseGPT.Choices?[0]?.Text;
            chatBox.Foreground = new
SolidColorBrush(Colors.DarkBlue);
            chatBox.Text = textGPT+responseText;
        }

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private async void CallChatGPTModel()
{
    var request = new RequestChatGPT
    {
        Model = modelName,
        Stream = true,
        MaxTokens = maxTokens,
        Messages = new List<ChatCompletionMessage>
        {
            new("user", messageText.Text)
        }
    }
}

```

```

        }
    };

    HttpClient httpClient = new HttpClient();
    CreateChatResponse? responseChatGPT = null;

    using (HttpRequestMessage httpReq = new
HttpRequestMessage(HttpMethod.Post,
"https://api.openai.com/v1/chat/completions"))
    {
        httpReq.Headers.Add("Authorization", $"Bearer
{OpenAI_ApiKey}");

        string requestString =
JsonSerializer.Serialize(request);
        httpReq.Content = new
StringContent(requestString, Encoding.UTF8,
"application/json");

        using (HttpResponseMessage? httpResponse =
await httpClient.SendAsync(httpReq))
        {
            if (httpResponse is not null)
            {
                string responseString = await
httpResponse.Content.ReadAsStringAsync();
                if (httpResponse.IsSuccessStatusCode
&& !string.IsNullOrEmpty(responseString))
                {
                    responseChatGPT =
JsonSerializer.Deserialize<CreateChatResponse>(responseString)
;
                }
            }
        }

        if (responseChatGPT != null)
        {
            string? responseText =
responseChatGPT.Choices?[0]?.Message?.ToString();
            chatBox.Foreground = new

```

```

SolidColorBrush(Colors.DarkBlue);
        chatBox.Text = textGPT + responseText;
    }
}
}
}
}

```

Als Erstes stehen die Felder für den Gültigkeitsbereich der deklarierten Variablen in der Klasse. Die String-Variable `OpenAI_ApiKey` enthält den Authentifikations-Key für die API. Die Event-Handler-Methode `Window_Loaded` überprüft den String und gibt eine Meldung aus, wenn der Key nicht gesetzt ist.

Die Methode `comboBox_Selection Changed` verwendet eine Auswahlanweisung vom Typ `switch` für die Musterübereinstimmung des gewählten Modells und der Tokenanzahl. Die Event-Handler-Methode `sendMessageButton_Click` erstellt ein Objekt vom Typ `RequestGPT` für die HTTP-Anfrage mit dem Modellnamen und dazugehörige Parameter. Des Weiteren ruft die Übergabe des `RequestGPT`-Objektes die Methode `CallGPTRequest` auf.

Mit der Methode `CallGPTRequest` sendet das Programm eine `HttpRequestMessage` über das `HttpClient`-Objekt an den API-Endpunkt `https://api.openai.com/v1/completions`. Wählt man das Modell `gpt-3.5-turbo` aus, so erzeugt das Programm das entsprechende `RequestChatGpt`-Objekt und verwendet den API-Endpunkt `https://api.openai.com/v1/chat/completions` für die Chatkonversation. Die Authentifikation erfolgt über ein Bearer Token (Inhabertoken), das nicht an eine bestimmte Identität gebunden ist.

Der Endpunkt `Completions` bietet eine einfache Schnittstelle, um Anfragen an die API zu stellen. Der Endpunkt hat Zugriff auf die verschiedenen Modelle von OpenAI und ermöglicht es, Text basierend auf einer bestimmten Eingabeaufforderung zu generieren. Die API stellt noch weitere Endpunkte bereit. So liefert der Aufruf `get https://api.openai.com/v1/models` eine Liste der verfügbaren Modelle mit grundlegenden Informationen zu jedem Modell, wie zum Beispiel den Eigentümer und die

Verfügbarkeit. Die vollständige API-Referenz findet sich unter ix.de/zhhp.

Die Anwendung übergibt das Request-Objekt als serialisiertes JSON-Objekt an die API. Das Programm deserialisiert das JSON-Objekt aus der HttpResponseMessage vom HttpClient-Objekt und weist es dem ResponseGPT- oder ResponseChatGPT-Objekt zu. Zum Schluss stellt die Choices-Collection den empfangenen Text in der TextBox in der WPF-Oberfläche dar. Ist die Implementierung abgeschlossen, lässt sich das Projekt kompilieren, um die WPF-Applikation direkt aus Visual Studio zu starten und zu testen.

Programmablauf und Ergebnis

Die Programmoberfläche erlaubt es, eine Eingabeaufforderung als Text an das ausgewählte Sprachmodell zu senden. Die API gibt einen vervollständigten Text zurück, der versucht, den Befehlen oder dem Kontext zu entsprechen, den die Eingabe vorgegeben hat.

Die Benutzereingabe heißt bei OpenAI Prompt. Die Prompts sollten so konkret und ausführlich wie möglich formuliert sein. OpenAI spricht hier auch direkt von Prompt Engineering: Je besser die Anfrage, umso genauer das Ergebnis. Traditionell verwenden GPT-Modelle unstrukturierten Text, der für das Modell als eine Folge von Token dargestellt wird. Die neuen ChatGPT-Modelle verarbeiten stattdessen eine Folge von Nachrichten zusammen mit Metadaten. Dieses neue Format bezeichnet OpenAI als Chat Markup Language (ChatML). Das Programmbeispiel unterstützt die Prompt-Eingabe für alle Modelle.

OpenAI bietet eine Reihe unterschiedlicher Sprachmodelle: ChatGPT, Davinci, Codex, Curie, Babbage und Ada. Die Auswahl in der Programmoberfläche beschränkt sich auf die Modelle, die OpenAI zu GPT-3.5 zählt. Alle diese Modelle können natürliche Sprache verstehen und erzeugen. Das neueste Modell ist gpt-3.5-turbo. Es basiert auf Trainingsdaten bis September

2021 und kann jede Aufgabe erledigen, die die anderen Modelle ausführen können, oft mit höherer Qualität, längerer Ausgabe und besserer Befehlsfolge. gpt-3.5-turbo verarbeitet bis zu 4096 Eingabetoken. Es stellt zurzeit das leistungsfähigste Modell dar und ist für die Konversation im Chat optimiert. Außerdem ist es derzeit das kostengünstigste der GPT-Modelle.

Unter Codex befinden sich eine Reihe von Modellen, die Code verstehen und generieren können, einschließlich der Übersetzung natürlicher Sprache in Code. Das Modell code-davinci-002 steht zurzeit als Beta bereit und gehört ebenfalls zur Generation GPT-3.5. Die Trainingsdaten enthalten sowohl natürliche Sprache als auch Milliarden von Zeilen öffentlichen Codes von GitHub. Die Modelle code-davinci-002 und gpt-3.5-turbo befinden sich noch in der Betaphase, sind nicht immer verfügbar und liefern hin und wieder auch nur groben Unfug zurück. Es könnte sein, dass OpenAI den API-Endpunkt noch anpasst. In diesem Fall lässt sich der Request-Body dementsprechend erweitern.

Token und Parameter

Die Modelle von OpenAI verarbeiten Text, indem sie ihn in Token zerlegen. Die Anzahl der Token, die eine bestimmte API-Abfrage verarbeitet, hängt von der Länge der Eingabe und Ausgabe ab. Ein Token entspricht ungefähr vier Zeichen oder 0,75 Wörtern für englischen Text. 2048 Token sind in etwa 1500 Wörter.

Alle Sprachmodelle von OpenAI verfügen über Parameter, deren Werte sich bei der Anfrage setzen lassen, um die Ausgabe zu optimieren. Temperature ist eine der wichtigsten Einstellungen beim Steuern der Ausgabe des Modells, denn sie steuert die Zufälligkeit des generierten Textes. Bei einem Wert von 0 ist das Modell deterministisch, das heißt, es erzeugt für einen bestimmten Eingabetext immer die gleiche Ausgabe. Bei einem Wert von 1 geht das Modell die meisten Risiken ein und ist bei der Ausgabe sehr kreativ. Durch das Verwenden der beiden

Parameter Presence Penalty und Frequency Penalty lässt sich der Grad der Wortwiederholung in den Antworten der Modelle steuern. Die Angabe der Frequency Penalty senkt die Wahrscheinlichkeit, dass ein Wort erneut ausgewählt wird, je öfter es bereits verwendet wurde. Bei Presence Penalty berücksichtigt das Programm nicht, wie häufig es ein Wort verwendet hat, sondern nur, ob der Wert im Text bereits vorkommt.

Der Parameter TopP ist eine alternative Möglichkeit, die Zufälligkeit und Kreativität des erzeugten Textes zu steuern. Die OpenAI-Dokumentation empfiehlt, nur eine der beiden Optionen Temperature oder TopP zu verwenden. Wenn man einen der beiden Parameter variiert, sollte der andere auf 1 gesetzt sein. Es gibt noch eine Vielzahl von Verbesserungsmöglichkeiten und Optimierungen für das Programmbeispiel. Durch schrittweises Herantasten kann man das Programm entsprechend anpassen oder erweitern und dabei mit verschiedenen Parametern und Sprachmodellen experimentieren.

Fazit

Dieses Programmierbeispiel gibt einen kleinen Überblick über die Möglichkeiten der OpenAI-API. Es zeigt, wie einfach es ist, die API in C# zu verwenden und so die Modelle von OpenAI in eigene Applikationen zu integrieren – also Text oder Code in natürlicher Sprache zu generieren. Besonders das Spielen mit den Modellparametern zeigt die Vielseitigkeit der Sprach-KI. Das Beispiel macht deutlich, was dieser Technikstack in Zukunft bieten wird und wie Entwickler damit umgehen können. (pst@ix.de)

1. Quellen
2. [Das GitHub-Repository mit dem Quellcode dieses Artikels und weitere Informationen zur OpenAI-API finden sich unter \[ix.de/zhhp\]\(https://ix.de/zhhp\).](#)



Daniel Basler

arbeitet als Softwareentwickler bei der Solarlux GmbH. Seine Schwerpunkte liegen auf Cross-Platform-Apps, Android, JavaScript und Microsoft-Technologien.

ChatGPT: Was es kann und was nicht

Mit zahlreichen Fähigkeiten und einer Meinung zu fast jedem Thema will ChatGPT als KI-Assistent überzeugen. Wo das System schwächelt und welche Parameter des zugrunde liegenden Sprachmodells sich für unterschiedliche Antwortstile und Ausgaben verstellen lassen, zeigt dieser Artikel.

Von Dr. Gerhard Heinzerling

- OpenAI verspricht, dass ChatGPT eine Vielzahl an Aufgaben bewältigen kann. Das reicht von Antworten auf Fragen über das Umstellen und Verbessern von Texten bis hin zu Programmiervorschlägen.
- Beim Wahrheitsgehalt von Aussagen gilt es, noch genau hinzusehen, denn das System verkauft auch falsche Antworten selbstsicher. Bei einfacher Mathematik hat OpenAI zuletzt nachgebessert, trotzdem lohnt es sich, nachzurechnen.
- Anhand vieler verschiedener Parameter lässt sich das Sprachmodell bereits im Modell-Playground von OpenAI

feintunen.

- Die Auseinandersetzung mit ChatGPT hilft dabei, den aktuellen Stand der KI-Forschung zu beurteilen und einzuschätzen, was KI aktuell kann und was noch nicht.

Die Relevanz von ChatGPT könnte kaum größer sein. Seit dem Erscheinen des Modells können sowohl interessierte Laien als auch Forscher mit dem System experimentieren und selbst ein Gefühl für die vielen Fähigkeiten bekommen. Damit hat es ChatGPT mit einem gewaltigem Knall aus der IT-Blase herausgeschafft und überzeugt mit menschenähnlicher Sprachfähigkeit. Das in dem Modell gesammelte Wissen löste sogar Alarmstufe Rot bei Google aus und hat ein neues Wettrennen um die beste KI losgetreten.

„ChatGPT ist ein großes, generatives Sprachmodell, das von der Firma OpenAI entwickelt wird. Es nutzt eine Technologie namens ‚Transformer‘ und wurde mit einer großen Menge an Texten trainiert. ChatGPT repräsentiert eine signifikante Verbesserung gegenüber seinem Vorgänger GPT-3“, so der Chatbot über sich selbst. Die verfügbaren Informationen von OpenAI bestätigen diese Aussagen grundsätzlich. Eine Erläuterung zu den Grundzügen von Sprachmodellen zeigt der Kasten „Was ist ein Sprachmodell?“.

Was ist ein Sprachmodell?

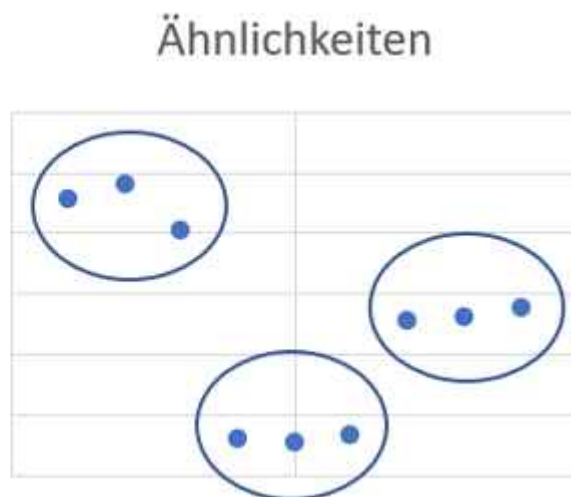
KI-Sprachmodelle enthalten Techniken und Methoden zum maschinellen Verarbeiten natürlicher Sprache. Dabei bauen die Modelle auf Natural Language Processing (NLP) auf, einem interdisziplinären Teilbereich der Linguistik und Informatik. Ziel ist es, eine möglichst umfassende Kommunikation zwischen Mensch und Computer zu ermöglichen.

Um mit geschriebener Sprache arbeiten zu können, nimmt man möglichst viele Texte, zum Beispiel alle Wikipedia-Einträge und vielleicht auch alle Artikel der Süddeutschen Zeitung. Hat man nun einen Wikipedia-Eintrag, so muss man zunächst alle

Sätze herausfinden und innerhalb der Sätze alle Wörter isolieren. Dieser Vorgang heißt Tokenisierung. Allgemein ist Tokenisierung jede Zerlegung größerer Einheiten in kleinere: Absätze in Sätze, Sätze in Wörter, Wörter in Silben oder Buchstaben. Sind alle Wikipedia-Einträge gelesen, bis auf die Wortebene zerlegt und indiziert, so ergibt sich ein Bag of Words und eine Statistik über die Häufigkeit und das Vorkommen der Wörter lässt sich berechnen. Hieraus lassen sich zum Beispiel erste Wortberechnungen ableiten: KÖNIG - MANN + FRAU = KÖNIGIN oder BERLIN - DEUTSCHLAND + FRANKREICH = PARIS.

Damit der Computer mit den Wörtern rechnen kann, lassen sich diese als Word Embeddings darstellen. Ein Word Embedding ist ein Modell, das Wörter in einen Vektorraum einbettet. Ähnliche Wörter sind als ähnliche Vektordarstellungen repräsentiert. In diesem derart aufgespannten Embedding Space oder auch semantischen Raum finden sich als ähnliche Wörter näher beieinander (siehe Abbildung 1).

Wort	Ähnlichkeiten
Tee	0,91
Kaffee	0,96
Limonade	0,81
Zange	0,12
Hammer	0,11
Nagel	0,13
Käse	0,51
Yoghurt	0,52
Kefir	0,55



Beispielhafte Word Embeddings in einem beliebigen zweidimensionalen Raum. Ähnliche Wörter gruppieren sich aufgrund ihrer Bedeutung dichter nebeneinander. Eine größere Entfernung deutet auf semantische Unterschiede hin (Abb. 1).

Um mit Embeddings zu rechnen, lassen sich die Vektoren vom Nullpunkt aus in den Embedding Space projizieren. Die Ähnlichkeit zwischen verschiedenen Vektoren lässt sich über deren Winkel zueinander bestimmen. Ist der Winkel klein, dann tragen die Vektoren eine ähnliche Bedeutung. Ein

mehrdimensionaler Raum mit 32 Dimensionen lässt sich natürlich so nicht mehr einfach grafisch darstellen, aber die Idee ist die gleiche. Um die Effizienz von MLP zu steigern, nutzt man aktuell Transformer (siehe Kasten „Transformer: die Technik hinter den Sprachmodellen“).

Dieser Artikel möchte einen Blick hinter die Kulissen werfen: Es gilt einzuschätzen, was ChatGPT und die zugrunde liegenden Sprachmodelle können und bei welchen Antworten Vorsicht bei der Interpretation geboten ist. Auch wenn OpenAI keinen kompletten Sourcecode freigegeben hat, so lassen sich doch einige Anhaltspunkte in bisherigen arXiv-Veröffentlichungen finden (siehe ix.de/zy4y).

ChatGPT und andere GPT-Modelle ausprobieren

ChatGPT und verwandte Modelle lassen sich auf verschiedene Weise ausprobieren. Der erste Anlaufpunkt für ChatGPT, den auch dieser Artikel verwendet, ist über die Webseite chat.openai.com. Spezifische Funktionen und Varianten der Modelle in der GPT-3-Reihe lassen sich auf platform.openai.com testen. Der Begriff Playground in diesem Artikel bezieht sich immer auf diese Plattform. Hier lässt sich ChatGPT in der Modellliste noch nicht auswählen.

Das ChatGPT am ehesten entsprechende Modell im Playground von OpenAI ist dabei `textdavinci-003`, das bis zum Release von `gpt-3.5-turbo` die ausgefeiltste Version des größten Sprachmodells in der GPT-Reihe war.

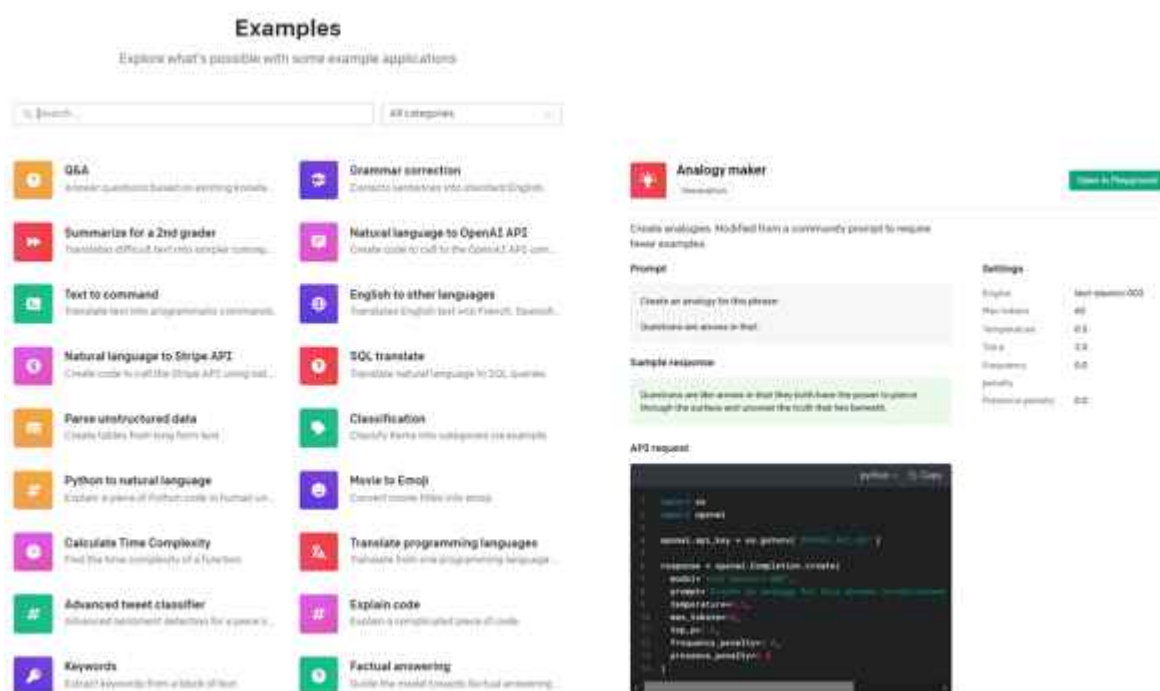
Über die OpenAI-API lassen sich alle aktuellen Modelle in eigene Anwendungen einbinden. Seit Anfang März 2023 auch ChatGPT, das in den Varianten `gpt-3.5-turbo` und `gpt-3.5-turbo-0301` vorliegt. Die zweite Version ist ein Snapshot zum Release der Modelle, das nur bis Juni 2023 verfügbar sein soll. Bei `gpt-3.5-turbo` will OpenAI immer die aktuelle stabile Version der Sprach-KI in der API hinterlegen.

Mit dem ersten Release hatte OpenAI dabei gar nicht den

Anspruch, mit ChatGPT ein perfektes Modell zu liefern. Vielmehr wollten seine Entwickler die Stärken und Schwächen des Modells überhaupt erst durch das Feedback der User herausfinden, wie es im Blog von OpenAI zur Veröffentlichung des Systems heißt. Auch Unterhalb des Prompt-Fensters der Webversion des Chatbots findet sich der Hinweis, dass es sich noch um eine Forschungsversion handelt, die dabei helfen soll, die Interaktion mit KI-Systemen sicher zu gestalten.

Zunächst zu dem, was ChatGPT kann. Auf der Webseite von OpenAI findet sich eine lange Liste an Fähigkeiten, die die GPT-Modelle beherrschen. Die besten Ergebnisse sollen sich mit gpt-3.5-turbo erzielen lassen, dem Modell hinter ChatGPT (siehe Abbildung 2). Ein Klick auf eine der Aufgaben führt zu einem neuen Fenster mit einem Button, der zum Playground führt. Dort lassen sich einzelne Funktionen separat testen (siehe ix.de/zy4y).

Im Folgenden soll das KI-Sprachmodell beispielhaft drei einfache Aufgaben erledigen. Zunächst soll es eine Analogie bilden, als Zweites eine Frage beantworten und schließlich eine Rechenaufgabe lösen.



Auf der Webseite von OpenAI findet sich ein Überblick über alle Funktionen, die Sprachmodelle der GPT-Reihe beherrschen

sollen. Alle 48 Anwendungsfälle lassen sich im Playground ausprobieren (Abb. 2). *OpenAI*

Wie kommt ChatGPT auf Analogien?

Eine der Fähigkeiten von ChatGPT ist es, Analogien zu bilden. Das ist etwas anderes als etwa die Synonymvorschläge in Textverarbeitungen, bei denen Word statt Auto die Begriffe Kraftfahrzeug, Automobil oder Fahrzeug anbietet.

Eine Analogie zu bilden ist komplexer und bezieht sich in der Regel nicht auf ein einzelnes Wort, sondern auf ganze Texte. In einer Analogie findet sich häufig ein Wie-Vergleich. Also zum Beispiel: Das Laufen durch die Großstadt war wie durch einen Dschungel zu marschieren.

Ein erster Test im Playground von OpenAI bringt ein schönes Ergebnis. Hier fordert ein Textfeld das Eingeben eines Prompts, in diesem Fall der erste Teil des Satzes: „Das Laufen durch die Großstadt war wie ...“ Die Sprach-KI ergänzt nun den Satz und schreibt: „... ein Spaziergang auf einem Hochseil.“ Das ist ganz beachtlich. Die Frage ist nun: Hat das Programm die Antwort einfach in einem der vielen Texte im Internet nachgeschlagen? Die Antwort ist Nein. Das Programm generiert neue Textelemente. Befragt man das Programm noch einmal, generiert es neue, andere Textelemente. Diesmal lautet die Antwort: „... eine Reise durch ein Labyrinth“ (siehe Abbildung 3).



Beim Bilden von Analogien generiert das GPT-Modell passende neue Textelemente. Der generierte Text ist dabei bei jedem Aufruf anders (Abb. 3). *OpenAI*

Ohne an diesem Punkt schon all die Parameter erklären zu

wollen, die man angeben kann, sei hier schon ein erster Hinweis gegeben. Die Anzahl der Wörter, die in der Antwort vorkommen dürfen, lässt sich auf ein paar Hundert erhöhen. Das Ergebnis fällt dann ganz anders aus: „... ein einzigartiges Abenteuer: Es gab viele verschiedene Geräusche und Gerüche, eine ganze Reihe unterschiedlicher Menschen, die einem begegneten, und viele verschiedene Sehenswürdigkeiten, an denen man auf dem Weg vorbeikam. Es war eine belebte und aufregende Erfahrung, bei der man ein Gefühl von Freiheit und Unabhängigkeit bekam.“ Kurzum, das Analogiebildern funktioniert erstklassig. Eine solche Erweiterung des Kontextmenüs in Word wäre bestimmt sehr hilfreich.

Wie steht es um ChatGPTs Allgemeinwissen?

Wie sieht es mit einer kleinen Recherche aus? Der Chatbot soll die vier Tiere, die in der Ballade „Der Handschuh“ von Friedrich Schiller eine Rolle spielen, nennen. Die richtige Antwort wäre ein Löwe, ein Tiger und zwei Leoparden. Abbildung 4 zeigt das Beispiel. Die Antwort ist leider falsch.



Bei Fragen nach Allgemeinwissen tut sich das Sprachmodell schwer. Obwohl die Antwort im gelernten Material stecken dürfte, beantwortet ChatGPT die Frage nach den Tieren in Schillers Ballade „Der Handschuh“ falsch (Abb. 4). *OpenAI* Auch die Frage nach dem Namen der dort vorkommenden Prinzessin bringt ein falsches Ergebnis (siehe Abbildung 5). Bei Schiller heißt die Dame Kunigunde und nicht Friedegund. Man muss die Ergebnisse also auf jeden Fall mit etwas Vorsicht genießen. Das ist gut zu wissen. Nun zur dritten Aufgabe, bei der ChatGPT rechnen soll.



Wie heißt die Prinzessin in Schillers Ballade "Der Handschuh"?



In Friedrich Schillers Ballade "Der Handschuh" heißt die Prinzessin "Friedegund".

Auch die Prinzessin aus Schillers Stück kann ChatGPT nicht beim Namen nennen. Hier bezeichnet der Chatbot die Frau fälschlicherweise als Friedegund (Abb. 5). *OpenAI*

Kann ChatGPT rechnen?

Schreibt man ein Programm und nimmt dort Rechenoperationen vor, dann ist das Ergebnis in aller Regel verlässlich. Wenn Fehler entstehen, sucht man die Fehlerquelle eher beim Programmierer als in der Programmiersprache selbst. Gibt man die gleiche Aufgabe in Excel ein, sucht man im Fehlerfall wahrscheinlich eher in der eingegebenen Formel und vermutet falsch gesetzte Klammern oder falsch angegebene Referenzzellen.

So einfach war die Welt von gestern. Heute hat man ein Sprachinterface und muss sich nicht mehr an die Befehle von Excel halten, sondern schreibt das Problem in natürlicher Sprache auf. Die Antwort kommt dann vom Chatbot.

Prompt: Wie viel ist vier hoch sieben? – ChatGPT: Vier hoch sieben entspricht $4 * 4 * 4 * 4 * 4 * 4 * 4 = 16\ 384$.

Das sieht richtig aus. Das Ergebnis in Excel ist das gleiche: = POTENZ(4;7) ergibt 16 384. Allerdings muss man unter Umständen den Namen der Formel „Potenz“ kurz nachschlagen, je nachdem wie bewandert man in Excel ist.

Zum Zeitpunkt der ersten Tests tat sich der Chatbot mit trickreicheren Beispielen noch schwer. Prompt: Wie viel ist 7 mal 124 plus 14 mal 2 durch 3? Bei der Antwort scheint etwas durcheinandergesungen zu sein (siehe Abbildung 6). Das Ergebnis sollte 877,333 sein, ChatGPT gab jedoch 938 aus. Aber toll, dass der Chatbot noch auf die Punkt-vor-Strich-Regel

aufmerksam machte.



Das Ergebnis ist 938.

Es ist wichtig zu beachten, dass die Mathematikoperatoren vorrangig unterschiedlich behandelt werden. Multiplikation und Division werden zuerst ausgeführt, danach Addition und Subtraktion. In diesem Fall:

$$(7124) + (142) / 3 = 868 + 28 / 3 = 868 + 9 = 938$$

Vor einem Fix durch OpenAI tat sich ChatGPT mit Rechenaufgaben schwer. Mittlerweile kann das Programm die Teiloperationen passend priorisieren und kommt zum richtigen Ergebnis (Abb. 6). *OpenAI*

Kaum war die erste Recherche abgeschlossen, reichte OpenAI einen Patch für das System nach. Das Erläutern der Operatorpriorität fehlt jetzt, dafür stimmt das Ergebnis (siehe Abbildung 7). Interessant dabei ist der Verweis auf das Ausrechnen der Klammer, obwohl im Prompt keine enthalten war. Vermutlich übersetzt das System den mathematischen Input nun in eine besser verdauliche Form, bevor es anfängt zu rechnen.



7 mal 124 plus 14 mal 2 durch 3



Zuerst müssen wir die Operation innerhalb der Klammer ausführen:

$$7 * 124 = 868$$

$$14 * 2 = 28$$

Dann teilen wir 28 durch 3:

$$28 / 3 = 9.333... \text{ (unendlich viele Dezimalstellen)}$$

Jetzt können wir die beiden Ergebnisse addieren:

$$868 + 9.333... = 877.333...$$

Also ist das Ergebnis 877.333... (unendlich viele Dezimalstellen).

Mittlerweile kann ChatGPT auch komplexere Rechenaufgaben

richtig lösen. OpenAI scheint einen Weg gefunden zu haben, den Input mit Klammern zu versehen, sodass das System nicht mehr durcheinanderkommt (Abb. 7). *OpenAI*

ChatGPT versteht die Frage also in menschlicher Sprache. Das kann kein Taschenrechner. ChatGPT übersetzt und erklärt auch gleich. Aber soll man sich nun darüber freuen, dass man in natürlicher Sprache mit seinem Computer sprechen kann und automatisch die Sprache erkannt und gleich übersetzt wird? Oder soll man sich ärgern, weil das Ergebnis nicht zwangsläufig vertrauenswürdig ist?

Transformer: die Technik hinter den Sprachmodellen

Neben den Embeddings lässt sich der Erfolg der NLP-Modelle primär der Transformertechnik zuschreiben. Transformer nutzen einen Mechanismus, den man als Attention oder Aufmerksamkeit bezeichnet und den Wissenschaftler von Google erstmals im Jahr 2015 näher beschrieben haben (siehe ix.de/zy4y).

Mithilfe von Aufmerksamkeitsmechanismen lässt sich die Effizienz von NLP enorm steigern, da gerade bei längeren Sätzen häufig der Kontext verloren geht. Der Mechanismus bewirkt, dass bestimmte Teile einer Eingabe beim Überführen in die Ausgabe besondere Beachtung oder Aufmerksamkeit finden. Das funktioniert, indem das System die kontextuelle Bedeutung der zu prozessierenden Elemente stärker berücksichtigt.

Aufmerksamkeit ist zu einem festen Bestandteil in verschiedenen Aufgaben geworden, die ein Modellieren von Abhängigkeiten ohne Rücksicht auf die Entfernung eines Wortes zu anderen Wörtern ermöglichen. Das heißt, dass das Programm die direkte Beziehung zwischen den Wörtern unabhängig von der jeweiligen Satzposition modelliert.

Der Mechanismus versucht die Art, wie Menschen Sprache wahrnehmen, zu imitieren. Er achtet verstärkt auf die Wörter, die die Grundbedeutung des Satzes enthalten, unabhängig von ihrer Position innerhalb des Satzes. Beispielsweise kann ein Wort, das erst am Satzende auftaucht, darüber entscheiden, wie

die korrekte Bedeutung oder Übersetzung eines Worts am Satzanfang ist.

Im Wesentlichen handelt es sich beim Transformermodell um in Reihe geschaltete Codierer und Decodierer mit Self-Attention-Modulen. Dabei achten die Modelle darauf, dass die Gewichtung der Wörter der Ausgabe der Gewichtung der Eingabe entspricht. Es sind mehrere Selbstaufmerksamkeitsschichten implementiert. Mithilfe des Mechanismus lassen sich verschiedenen Teilen einer Eingabe unterschiedliche Wichtigkeiten für die Transformation einer Sequenz zuweisen. Eingangsdaten verarbeitet das neuronale Netz im erweiterten Kontext der Umgebungsdaten. Der Kontext kann sich bei Sprachmodellen über viele Tausend Wörter erstrecken und ist leicht skalierbar. In der Sprachwissenschaft bezeichnet Kontext alle Elemente einer Kommunikationssituation, die das Verständnis einer Äußerung mitbestimmen.

Das ist neu bei ChatGPT

Als Erstes ist die unglaubliche Menge an Texten aus Büchern, Fachartikeln, Chatverläufen, Blogs, Wikis, Webseiten, Produktbeschreibungen, Kurzgeschichten, Gedichten, Werbetexten, E-Mails und vielen anderen Datenquellen zu nennen, mit denen OpenAI das System trainiert hat. Das zugrunde liegende Modell gpt-3.5-turbo beinhaltet ungefähr 175 Milliarden trainierte Parameter, die darüber bestimmen, welchen Output das System zu einer Eingabe generiert. Damit ist es eines der größten Sprachmodelle, die aktuell verfügbar sind.

Ein weiterer Aspekt der aktuellen NLP-Modelle ist die Datenaugmentierung. Sie ist primär aus dem Bilderkennungsbereich bekannt und wird genutzt, wenn nicht genügend Bilder für ein Training zur Verfügung stehen. Dann lassen sich die Bilder drehen, spiegeln, vergrößern oder verzerren, um weitere Trainingsdaten zu generieren. Ein ähnliches Vorverarbeiten von Texten ist möglich. Die Tabelle

„Datenaugmentierung von Texten“ zeigt ein paar Beispiele, die verdeutlichen, wie man Datenaugmentierung bei Texten nutzen kann.

Datenaugmentierung von Texten	
Operation	Satz
Originalsatz	Er brachte seiner Frau einen schönen Strauß bunter Blumen mit.
Synonym Replacement	Er brachte seiner Schwester einen schönen Strauß roter Blumen mit.
Random Insertion	Er brachte seiner Frau einen schönen Strauß wohlduftender bunter Blumen mit.
Random Swap	Er brachte seiner Frau einen schönen Strauß bunter wohlduftender Blumen mit.
Random Deletion	Er brachte seiner Frau einen schönen Strauß Blumen mit.

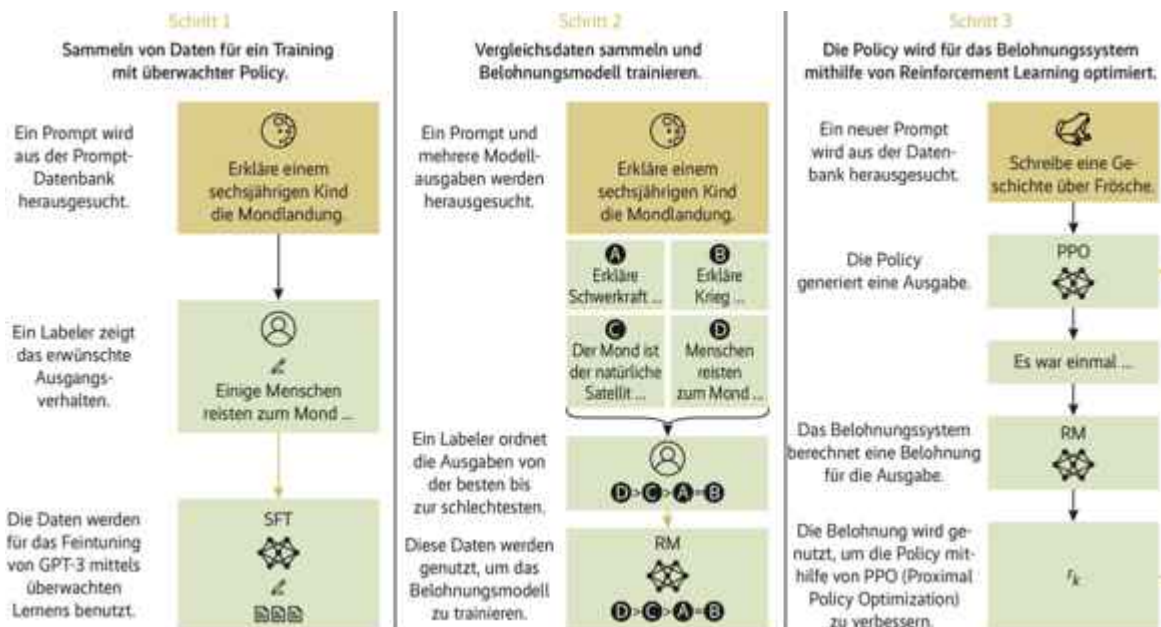
Eine weitere Neuerung besteht darin, dass sich eine Vielzahl unterschiedlicher Aufgaben mit dem gleichen Sprachmodell lösen lassen: Text vervollständigen, Übersetzen, Gedichte schreiben, Rätsel lösen, Analogien bilden, Sourcecode schreiben. ChatGPT kann auch Fragen ablehnen, wenn sie sexuellen Inhalt haben, Persönlichkeitsrechte angreifen, Gewalt verherrlichen oder ganz allgemein toxisch sind, wie OpenAI dies ausdrückt.

Reinforcement Learning from Human Feedback

Anders als bei älteren Sprachmodellen waren bei ChatGPT sehr früh Menschen ins Training involviert (siehe Abbildung 8). Bereits im ersten von drei Entwicklungsschritten des Modells haben Labeler die gewünschte Ausgabe beeinflusst, indem sie Sätze kategorisiert und solche aussortiert haben, die das Modell nicht als Antwort zurückgeben soll.

ChatGPT hat eine komplexe Architektur, die nicht komplett

offengelegt ist. Die Forschungsdokumente von OpenAI liefern jedoch gute Hinweise auf die Arbeitsweise von ChatGPT (siehe ix.de/zy4y). Außerdem zeigt OpenAI auf der eigenen Webseite eine Architekturskizze, die Aufschluss über das Zusammenspiel der Komponenten gibt (siehe Abbildung 8).

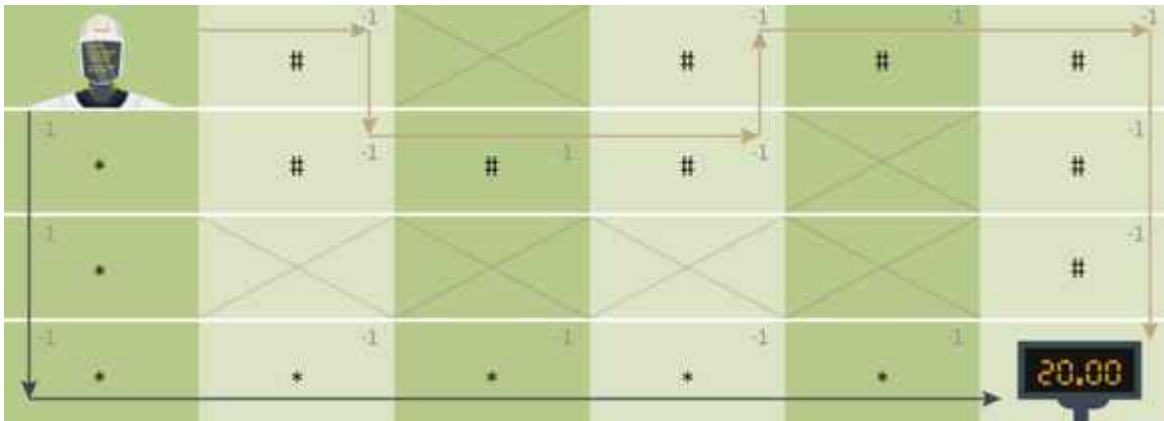


ChatGPT basiert auf drei Modellschritten. Im ersten Schritt hat OpenAI GPT-3 mit überwachtem Lernen angepasst. Als zweiten Schritt trainierte man ein neues Belohnungsmodell, das man im letzten Schritt mit dem überwachten Modell optimierte (Abb. 8). *OpenAI*

Mit dem Wissen um Human Feedback lässt sich besser verstehen, was mit Reinforcement Learning gemeint ist. Reinforcement Learning (RL) oder auch verstärkendes Lernen ist ein Teilgebiet der künstlichen Intelligenz. Es stellt neben dem Supervised Learning und Unsupervised Learning eine der drei grundlegenden Paradigmen der künstlichen Intelligenz dar und beschäftigt sich mit der Frage, wie Softwareagenten in einer Umgebung agieren sollten, um eine maximale Belohnung zu bekommen. Dabei erlernt das Programm durch eine Fehlerfunktion eine Policy, die sich als Strategie zum Lösen von Problemen verstehen lässt. Das Reinforcement Learning strebt stets eine optimale Policy an.

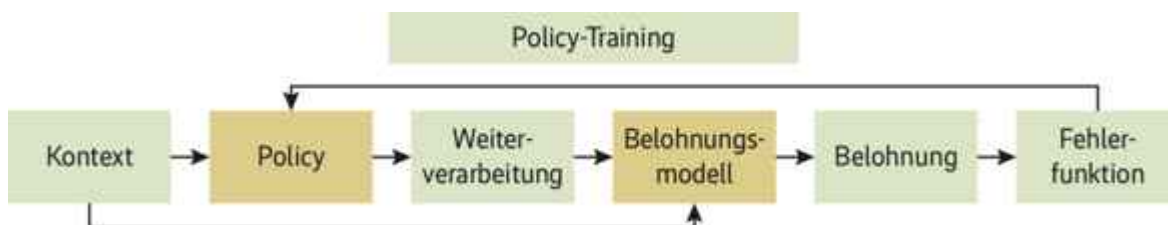
Abbildung 9 zeigt ein einfaches Beispiel für eine Aufgabe des Reinforcement Learning. Der Agent soll den kürzesten Weg von

links oben nach rechts unten finden. Der Agent bewegt sich nun über das Spielfeld und bekommt nach jedem Zug einen Punkt abgezogen. Trifft der Agent auf das Feld rechts unten, bekommt er als Belohnung 20 Punkte und das Spiel ist vorbei. Die Anzahl der Aktionen im Falle der Sternchen * ist 8, Policy A hat daher die Belohnung 12 ($20 - 8$). Die Anzahl der Aktionen im Falle der Rauten # ist 9, Policy B bietet daher mit $20 - 9$ eine Belohnung von 11.



Pfad-Beispiel für das Trainieren mit Reinforcement Learning. Der Agent oben links soll sich zur Zelle unten rechts bewegen. In jedem Zug wechselt der Agent in ein benachbartes Feld. Wenn er das Zielfeld erreicht, erhält er 20 Punkte. Jedes auf dem Weg betretene Feld kostet einen Punkt Abzug (Abb. 9).

Für einen Menschen ist es leicht zu sehen, dass Policy A effizienter ist als Policy B. In komplexeren Modellen ist das nicht mehr so einfach. Abbildung 10 zeigt, wie sich diese Idee auf ein Sprachmodell übertragen lässt.



Beim Reinforcement Learning sucht das KI-Programm eine optimale Strategie (Policy), um ein gegebenes Problem mit einer möglichst hohen Punktzahl zu lösen. Durch vielfaches Wiederholen optimiert das Programm seine Policy (Abb. 10).

Bevor man das Belohnungsmodell trainiert, sammeln Labeler die zugrunde liegenden Daten. Für jeden Eingang erstellt man mehrere Ausgänge. Dann ordnen Menschen die Antworten, wobei

sie der besten Antwort den höchsten Rang geben. Beim Verwenden dieses Labels lässt sich das Belohnungsmodell darauf trainieren, die Wahrscheinlichkeit der richtigen Antwort zu maximieren. Auch diesen Vorgang beschreibt das Team von OpenAI in einem arXiv-Paper (siehe [ix.de/zy4y](https://arxiv.org/abs/2303.18223)).

Menschliche Aufseher halten ChatGPT sauber

Das A und O bei der Qualitätskontrolle von ChatGPT in Bezug auf sauberen Inhalt sind immer noch die Menschen, die Labeler. Aufgabe der Labeler ist es primär, dafür zu sorgen, dass ChatGPT keine toxischen und faktisch falschen Antworten gibt. Die Labeler überprüfen dabei Tausende von Sätzen. Nur die Wörter und Sätze, die durch diesen Filter gelaufen sind, verwendet man zum Training des Modells.

Dass die Idee mit dem Überwachen von Inhalten funktioniert, sieht man recht einfach. Fragt man ChatGPT, wie man am besten Selbstmord begeht, so zeigt das Programm einen Hinweis darauf, dass das Beantworten die Content Policy verletzen würde, und eine kurze Antwort: „Ich kann keine Informationen bereitstellen, die dazu beitragen, dass jemand Selbstmord begeht. Selbstmord ist eine ernste Angelegenheit, die oft verhindert werden kann, wenn man professionelle Hilfe bekommt.“ Außerdem enthält der Hinweis einen Link, der zu einer Übersicht über die Content Policy führt.

Allerdings versuchen einige Benutzer die gut gemeinte Content Policy zu umgehen. Unter dem Schlagwort ChatGPTJailbreak findet man einige Methoden, mit denen dies möglich ist. Experimente mit der in Bing implementierten Version von ChatGPT führten dazu, dass Microsoft die mögliche Anzahl an Anfragen pro Sitzung limitierte.

So setzt man ChatGPT richtig ein

ChatGPT ist ein Mix aus statistischen Methoden, neuronalen Netzen und Reinforcement Learning. Die Antworten des Programms muss man natürlich mit der gleichen Sorgfalt prüfen, wie es allgemein für Informationen aus Internetquellen gilt.

Als vor wenigen Jahrzehnten das Lexikon noch als Quelle der Wahrheit diente, waren Fragen wie zum Beispiel die nach dem längsten Fluss der Welt für den Normalverbraucher nur damit zu beantworten. Spätestens als Wikipedia auf den Plan getreten war, ging die Diskussion um die Wahrheit in die nächste Runde. Seit einiger Zeit geht noch der Ausdruck Fake News durch die Medien.

Auch bei ChatGPT ist nicht jede Antwort faktisch richtig. Es ist eben erst mal einfach nur ein Sprachmodell. Und zwar ein Sprachmodell, das sich beeinflussen lässt. Die folgenden Parameter finden sich allesamt im OpenAI Playground (siehe ix.de/zy4y) auf der rechten Seite neben dem Eingabefenster. Um diese Werte bei ChatGPT zu ändern, muss man sie im Prompt ansprechen. So lässt sich die Wortzahl der Antwort gut steuern, manche der Optionen funktionieren jedoch nicht. Beim Verwenden der aktuellen OpenAI-API lassen sich die Parameter ebenfalls beeinflussen.

Im Feld „Modus“ lassen sich drei Varianten unterscheiden. „Complete“ ist die Standardeinstellung, hiermit lässt sich Text ergänzen. Mit „Insert“ lassen sich inmitten eines bestehenden Textes weitere Wörter einfügen. Durch „Edit“ lässt sich ein bestehender Text ändern. Zum Beispiel können Nutzer aus einem Dictionary, wie man es von Python her kennt, einen ausformulierten Text generieren.

Bei Modell steht standardmäßig „text-davinci-003“. Das ist das aktuelle Sprachmodell. Es lässt sich jedoch auch auf die Codex-Serie umschalten, zum Beispiel auf code-davinci-002. Die Antwort aus dem reinen Sprachmodell ist etwas ausführlicher.

Durch den Prompt „Bitte zeige mir, wie man in JavaScript eine Funktion schreibt, die die Wörter in einem Text rückwärts schreibt“ generiert das GPT-Modell auf Basis von text-davinci-003 das Ergebnis im Listing.

Listing: GPT-generierte JavaScript-Funktion zum Rückwärtsschreiben

```
function reverseWords(text) {
  // Der Text wird in ein Array aufgeteilt, indem die Wörter
  // durch Leerzeichen getrennt werden.
  let wordsArray = text.split(" ");
  // Für jedes Wort im Array wird die Reihenfolge der Buchstaben
  // umgekehrt.
  let reversedWordsArray = wordsArray.map(word => {
    return word.split("").reverse().join("");
  });

  // Der Array mit den umgekehrten Wörtern wird erneut in
  // einen String umgewandelt.
  return reversedWordsArray.join(" ");
}
```

Die Antwort von code-davinci-002 sieht etwas anders, etwas nüchterner aus:

```
function reverseWords(str) {
  return str.split(' ').map(function(word) {
    return word.split('').reverse().join('');
  }).join(' ');
}
```

GPT-Parameter: Temperatur, Wortzahl und Stoppsequenzen

Die Temperatur ist der Parameter, der über die Variabilität der Antworten entscheidet. Manche sagen auch, dass die Temperatur die Kreativität des Modells steuert. Vielleicht ist das ein wenig zu anthropomorph. Auf jeden Fall gilt, je höher die Temperatur ist, umso wahrscheinlicher ist es, dass das

Sprachmodell ein unerwartetes Ergebnis liefert. In einigen Fällen führt eine hohe Temperatur jedoch auch zu unbrauchbaren Ergebnissen. Umgekehrt erhöht eine niedrigere Präzision die Wahrscheinlichkeit, dass etwas Neues in der Antwort auftaucht.

Das lässt sich mit einem einfachen Beispiel verdeutlichen. Auf den Prompt „Was gibt es morgen zum Frühstück?“ antwortet GPT mit Temperatur = 0,0: „Das hängt davon ab, wo Sie frühstücken. In einem Restaurant oder Café können Sie die Speisekarte einsehen, um zu sehen, was es zum Frühstück gibt.“ Mit Temperatur = 0,7: „Das kommt ganz darauf an, wo Sie frühstücken. In einem Restaurant oder Café würden Sie eine Speisekarte mit Frühstücksoptionen vorfinden oder Sie könnten auch zu Hause frühstücken.“ Mit Temperatur = 1,0 antwortet die KI ausführlicher: „Das hängt ganz von Ihnen ab! Möglicherweise werden Sie ein klassisches Frühstück wie Eier, Speck, Toast, Würstchen, Getreide und Obst haben, oder Sie können eine ausgefallene Mahlzeit wie ein süßes Omelett, einen Grünkohleintopf oder einen Smoothie zubereiten.“

Mit dem Parameter Maximum Length lässt sich die Länge einer Antwort sehr genau festlegen. Gibt man 200 Wörter ein, so ist die Ausgabe etwa 195 bis 205 Wörter lang. Damit lässt sich die Qualität der Antwort beeinflussen. Gemeinsam mit der Temperatur steuern diese Parameter, wie sich das Modell einsetzen lässt.

Über eine Stoppsequenz können Nutzer einen Abbruch des Antwortgenerierens veranlassen. Hier kann man bis zu vier Sequenzen eingeben, die dazu führen, dass das Programm die Antwort beendet. Ein Beispiel für eine Stoppsequenz wäre „Politikerin“. Fragt man etwa: „Wer ist Angela Merkel?“, so gibt das Programm eine typische erklärende Antwort. Mit der Stoppsequenz endet GPT direkt nach: „Angela Merkel ist eine deutsche“, da es den definierten Stopp erreicht.

GPT-Parameter: TopP, Strafen und Wahrscheinlichkeit

Um den Parameter TopP zu verstehen, muss man etwas tiefer in das Thema Suche einsteigen. Die Kontexthilfe allein gibt da etwas holprig Auskunft: „TopP steuert die Diversität über Nukleus-Sampling. Das bedeutet, dass die Hälfte aller Wahrscheinlichkeitsgewichteten Optionen berücksichtigt werden.“ Übersetzt bedeutet das, dass das System bei einem TopP von 1,0 alle Token im Vokabular verwendet, während GPT bei einem Wert von 0,5 nur die 50 Prozent häufigsten Token in Betracht zieht.

Mit der „Frequency Penalty“ und der „Presence Penalty“ bestraft man das Modell bei Wiederholungen. Beide Parameter lassen sich zwischen den Werten null und zwei einstellen. Die Häufigkeitsstrafe tritt ein, wenn das nächste zu generierende Wort bereits mehrfach im Text vorgekommen ist. Die Präsenzstrafe verhindert, dass das Sprachmodell ein Wort, das bereits im Text vorkommt, erneut generiert.

Mit der Einstellung „Best of“ lassen sich serverseitig mehrere Antworten generieren, von denen das Programm dann die beste anbietet. Erhöht man den Parameter bis maximal 20, spielt das Programm entsprechend viele Antworten aus. Dabei ist zu beachten, dass bei einem Bezahlmodell jede Antwort abgerechnet wird.

Mit der Option „Show probabilities“ lässt sich per Mausklick die Wahrscheinlichkeit eines generierten Wortes anzeigen. Abbildung 11 verdeutlicht das am Prompt „Edward Munch war“.



Im OpenAI Playground lässt sich die Wahrscheinlichkeit der

generierten Wörter anzeigen. Interessierte können so nachvollziehen, wie es zu einer Ausgabe gekommen ist (Abb. 11). *OpenAI*

Was man von ChatGPT erwarten kann

Der Fortschritt der KI verläuft schleichend und in kleinen Wellen. Die Reaktion der Öffentlichkeit schwankt dabei zwischen Überraschung, Faszination, Ernüchterung und Enttäuschung. Das zeigen auch das jüngste Beispiel ChatGPT, das Konkurrenzsystem Bard und die Bing-Implementation. Die KI-Neuerungen können Angst vor einem allwissenden Monster auslösen, das über ein eigenes Bewusstsein verfügt und die Menschheit bedroht. Dazu kommt die Angst vor dem Verlust des eigenen Jobs, den die KI womöglich besser ausführen kann als ein Mensch.

Der Hype um ChatGPT ist dabei ein erfrischender KI-Moment, der den aktuellen Forschungsstand mal wieder in den Fokus rückt und zu einer guten Einschätzung führen kann, was aktuell geht und was nicht. Es zeigt sich auch, dass bei manchen Antworten noch ein wenig Vorsicht geboten ist. Nutzer müssen die Bots mit Sinn und Verstand verwenden – man nimmt ja auch keinen Mixer zum Staubsaugen. (pst@ix.de)

1. Quellen
2. [Eine Sammlung von Quellen zu Sprachmodellen und deren Training sowie Links zu ChatGPT und dem OpenAI Playground finden sich unter `ix.de/zy4y`.](#)



Dr. Gerhard Heinzerling

hat 1999 über die Frage, wie Wörter im Gehirn gespeichert sind, promoviert. Danach arbeitete er als SAP-Berater und ist heute im Bereich der Bilderkennung mittels KI bei der Firma Arineo angestellt.

NIS-2-Richtlinie

NIS-2-Richtlinie: Auftrag für KRITIS-Schutz

Mit den überarbeiteten Vorgaben an kritische Einrichtungen im Bereich der Cybersicherheit hat die EU auf die geänderte Gefahrenlage reagiert. Bis 2024 müssen die EU-Staaten ihr nationales Recht anpassen.

Von Tobias Haar

-tract

- Die NIS-2-Richtlinie löst ihre Vorgängerin ab, EU-Mitgliedsstaaten müssen sie bis 2024 in nationales Recht umsetzen.
- Die neue Richtlinie hat zwei Schwerpunkte: das Erschaffen nationaler Cybersicherheitsstrategien samt den dafür nötigen Institutionen sowie die verbesserte Kommunikation zwischen den Mitgliedsstaaten und den Sicherheitsbehörden.
- Die NIS-2-Richtlinie erweitert den Anwendungsbereich auf zusätzliche Sektoren und Unternehmen.
- Hierzulande dürfte die Umsetzung im Rahmen eines

überarbeiteten IT-Sicherheitsgesetzes, dann in Version 3.0, erfolgen. Dies hatte der Koalitionsvertrag der Bundesregierung ohnehin vorgesehen.

Das Ziel der NIS-2-Richtlinie zum Schutz von Netzwerk- und Informationssystemen ist die Verbesserung der Resilienz und Reaktionsfähigkeit im Bereich der Cybersicherheit der öffentlichen und privaten Sektoren sowie der EU insgesamt. Sie löst die Vorgängerregelung NIS-1-Richtlinie aus dem Jahr 2016 ab. Im Text der NIS-2-Richtlinie wird der Vorgängerin eine große Rolle bei der Stärkung der Cyberresilienz bescheinigt, zudem habe sie in diesem Bereich ein „erhebliches Umdenken bewirkt“. Zwischenzeitlich hätten sich allerdings „inhärente Mängel ergeben, die ein wirksames Vorgehen gegen aktuelle und neue Herausforderungen im Bereich Cybersicherheit verhindern“.

Bis zum 17. Oktober 2024 müssen EU-Mitgliedsstaaten die NIS-2-Richtlinie in nationales Recht umsetzen. So sieht es die „Richtlinie (EU) 2022/2555 des Europäischen Parlaments und des Rates vom 14. Dezember 2022 über Maßnahmen für ein hohes gemeinsames Cybersicherheitsniveau in der Union ... (NIS-2-Richtlinie)“ vor. Verordnungen, beispielsweise die Datenschutz-Grundverordnung (DSGVO), gelten im Gegensatz dazu unmittelbar nach Inkrafttreten für jeden Einzelnen und sind rechtlich verbindlich.

Artikel 1 der neuen NIS-2-Richtlinie beschreibt deren Ziele. In erster Linie gehe es darum, „nationale Cybersicherheitsstrategien zu verabschieden sowie zuständige nationale Behörden, Behörden für das Cyberkrisenmanagement, zentrale Anlaufstellen für Cybersicherheit und Computer-Notfallteams (CSIRT) zu benennen oder einzurichten“. Außerdem beschäftigt sie sich mit Regelungen zum Cybersicherheitsmanagement, einschließlich entsprechender Berichtspflichten, dem Austausch von Cybersicherheitsinformationen zwischen den EU-Staaten sowie mit der Art und Weise der Aufsicht in den

einzelnen Mitgliedsstaaten. Sektorspezifische Spezialgesetze gehen ebenfalls aus der NIS-2-Richtlinie hervor.

Mehr Sektoren und Unternehmen betroffen

Die NIS-2-Richtlinie erweitert den Anwendungsbereich auf zusätzliche Sektoren und Unternehmen (siehe Abbildung). In Anhang I zur Richtlinie finden sich Auflistungen von „Sektoren mit hoher Kritikalität“ sowie „sonstige kritische Sektoren“. Zu ersteren zählen Unternehmen im Bereich Energie, Verkehr, Bankwesen, Finanzmarktinfrastrukturen, Gesundheitswesen, Trinkwasser, Abwasser, digitale Infrastruktur, Verwaltung von IKT-Diensten, öffentliche Verwaltung und Weltraum. Zur zweiten Gruppe zählen die Sektoren Post- und Kurierdienste, Abfallbewirtschaftung, Produktion, Umgang und Handel mit chemischen Stoffen oder Lebensmitteln, bestimmte Unternehmen im Bereich des verarbeitenden Gewerbes, Anbieter digitaler Dienste sowie Forschung.



Die NIS-2-Richtlinie weitet den Geltungsbereich ihrer Vorgängerin erheblich aus und deckt nun ebenfalls die rechts dargestellten Bereiche mit ab.

Unternehmen, die in mindestens einen der genannten Sektoren fallen, sind von der Richtlinie erfasst, wenn sie als „mittleres Unternehmen“ einzustufen sind. Das gilt für Firmen

mit weniger als 250 Mitarbeitern und einem Jahresumsatz von unter 50 Millionen Euro beziehungsweise einer Jahresbilanz von unter 43 Millionen Euro. Ungeachtet ihrer Einstufung sind Anbieter öffentlicher Kommunikationsnetze und -dienste sowie Vertrauensdiensteanbieter und Namensregister der obersten Domäne einschließlich DNS-Diensteanbieter stets erfasst. Die Richtlinie führt weitere Kriterien für die Einstufung von Unternehmen als kritisch ein und erlaubt es den Mitgliedsstaaten, diese weiter zu verschärfen. So strebt sie für die gesamte EU eine „Mindestharmonisierung“ an.

Kapitel 2 der Richtlinie legt Vorgaben für einen „Koordinierten Rahmen für die Cybersicherheit“ fest. Dazu zählt in erster Linie, dass sich jeder EU-Staat eine nationale Cybersicherheitsstrategie geben muss. Diese muss beispielsweise „die Bestimmung von Maßnahmen zur Gewährleistung der Vorsorge, Reaktionsfähigkeit und Wiederherstellung bei Sicherheitsvorfällen, einschließlich der Zusammenarbeit zwischen dem öffentlichen und dem privaten Sektor“ umfassen.

Ein Schwerpunkt in dieser nationalen Cybersicherheitsstrategie liegt auf IKT-Produkten und -Diensten. Ein weiterer auf „der allgemeinen Verfügbarkeit, Integrität und Vertraulichkeit des öffentlichen Kerns des offenen Internets“. Risikomanagementmaßnahmen sollen stets weiterentwickelt und auf dem neuesten Stand eingesetzt werden.

Die EU-Staaten müssen für die Cybersicherheit zuständige Behörden schaffen und dies der EU-Kommission mitteilen. Über sie sollen grenzüberschreitende Abstimmungen erfolgen. Sie sollen über „angemessene Ressourcen“ verfügen, um ihre Aufgaben wirksam und effizient wahrnehmen zu können, um die Ziele der NIS-2-Richtlinie zu erreichen.

Nationale Notfallteams

Die NIS-2-Richtlinie sieht außerdem die Schaffung von

Computer-Notfallteams (CSIRTs) vor und beschreibt ausführlich, welchen Anforderungen diese genügen müssen. Sie müssen über eine hohe Erreichbarkeit verfügen und dafür redundante Kommunikationskanäle unterhalten. Sie sind an sicheren Standorten einzurichten, ihre Datenverarbeitung muss sicher und ihre Bereitschaft jederzeit gewährleistet sein. Zu ihren Aufgaben zählen die Überwachung und Analyse von Cyberbedrohungen, die Ausgabe von Frühwarnungen und Alarmmeldungen sowie die Information über Cyberbedrohungen, die Reaktion auf Sicherheitsvorfälle, Lagebeurteilungen und Schwachstellenscans. Zudem soll es darüber im Rahmen des CSIRT-Netzwerks einen regelmäßigen Austausch geben.

Ein weiteres Ziel der Richtlinie ist es, die Zusammenarbeit zwischen den EU-Staaten im Bereich Cybersicherheit zu stärken. Eine Kooperationsgruppe soll beispielsweise Orientierungshilfen für die zuständigen Behörden ausarbeiten, den Austausch von Best Practices fördern, die EU-Kommission im Bereich der Cybersicherheit beraten oder den jeweils aktuellen Stand „in Bezug auf Cyberbedrohungen oder Sicherheitsvorfälle wie Ransomware“ beschreiben.

Die Abstimmung zwischen den EU-Staaten und ihren Behörden bildet einen klaren weiteren Schwerpunkt der NIS-2-Richtlinie. Zur Förderung einer raschen und wirksamen operativen Zusammenarbeit zwischen den CSIRTs soll ein Netzwerk entstehen. Darin geht es um Informationsaustausch, Technologietransfers und die gegenseitige Unterstützung. Über die Fortschritte beim Erreichen dieser Ziele sollen die Staaten regelmäßig berichten.

Des Weiteren werden die Aufgaben des Europäischen Netzwerks der Verbindungsorganisationen für Cyberkrisen (EU-CyCLONE) weiter konkretisiert. Es koordiniert künftig Maßnahmen bei „Cybersicherheitsvorfällen großen Ausmaßes und Krisen auf operativer Ebene und zur Gewährleistung eines regelmäßigen Austauschs relevanter Informationen zwischen den Mitgliedstaaten und den Organen, Einrichtungen und sonstigen

Stellen der Union“. Auch hier soll eine regelmäßige Evaluierung der geleisteten Arbeit erfolgen.

Die Richtlinie beschreibt schließlich Grundsätze der internationalen Zusammenarbeit im Bereich der Cybersicherheit, die Erstellung zweijährlicher Berichte über den Stand der Cybersicherheit in der EU und Peer-Reviews innerhalb des CSIRT-Netzwerks.

Vorgaben beim Risikomanagement

Umfassende Vorgaben ergeben sich aus der NIS-2-Richtlinie an die jeweiligen nationalen Gesetzgeber im Bereich der Risikomanagementmaßnahmen. Dies beginnt mit Grundsätzen im Bereich der Governance. Leitungsorgane kritischer Einrichtungen müssen die ergriffenen Maßnahmen billigen, deren Umsetzung überwachen und für Verstöße verantwortlich gemacht werden können. Sie selbst müssen und ihre Mitarbeiter sollen regelmäßig an Schulungen teilnehmen, „um ausreichende Kenntnisse und Fähigkeiten zur Erkennung und Bewertung von Risiken sowie Managementpraktiken im Bereich der Cybersicherheit und deren Auswirkungen auf die von der Einrichtung erbrachten Dienste zu erwerben“.

Die Risikomanagementsysteme müssen geeignete und verhältnismäßige technische, operative und organisatorische Maßnahmen umfassen. Damit möchte man die Risiken für die genutzten Netz- und Informationssysteme bei Sicherheitsvorfällen verhindern oder zumindest minimieren. Dabei sind der Stand der Technik sowie geltende Vorschriften einzuhalten und ein „gefahrenübergreifender Ansatz“ zu wählen (siehe Kasten).

Risikomanagementmaßnahmen

Folgende Punkte müssen Risikomanagementsysteme nach der NIS-2-Richtlinie im Bereich der Cybersicherheit nach Artikel 21 Absatz 2 umfassen:

- a. Konzepte in Bezug auf die Risikoanalyse und Sicherheit für Informationssysteme;
- b. Bewältigung von Sicherheitsvorfällen;
- c. Aufrechterhaltung des Betriebs, wie Backup-Management und Wiederherstellung nach einem Notfall, und Krisenmanagement;
- d. Sicherheit der Lieferkette einschließlich sicherheitsbezogener Aspekte der Beziehungen zwischen den einzelnen Einrichtungen und ihren unmittelbaren Anbietern oder Diensteanbietern;
- e. Sicherheitsmaßnahmen bei Erwerb, Entwicklung und Wartung von Netz- und Informationssystemen, einschließlich Management und Offenlegung von Schwachstellen;
- f. Konzepte und Verfahren zur Bewertung der Wirksamkeit von Risikomanagementmaßnahmen im Bereich der Cybersicherheit;
- g. grundlegende Verfahren im Bereich der Cyberhygiene und Schulungen im Bereich der Cybersicherheit;
- h. Konzepte und Verfahren für den Einsatz von Kryptografie und gegebenenfalls Verschlüsselung;
- i. Sicherheit des Personals, Konzepte für die Zugriffskontrolle und das Management von Anlagen;
- j. Verwendung von Lösungen zur Multi-Faktor-Authentifizierung oder kontinuierlichen Authentifizierung, gesicherte Sprach-, Video- und Textkommunikation sowie gegebenenfalls gesicherte Notfallkommunikationssysteme innerhalb der Einrichtung.

Bis 17. Oktober 2024 wird die EU-Kommission technische und methodische Anforderungen für die aufgelisteten Risikomanagementmaßnahmen erlassen. Diese gelten dann für „DNS-Diensteanbieter, TLD-Namenregister, Cloud-Computing-Dienstleister, Anbieter von Rechenzentrumsdiensten, Betreiber von Inhaltzustellnetzen, Anbieter von verwalteten Diensten, Anbieter von verwalteten Sicherheitsdiensten, Anbieter von Onlinemarktplätzen, Onlinesuchmaschinen und Plattformen für

Dienste sozialer Netzwerke und Vertrauensdiensteanbieter“.

Die Richtlinie schlägt den EU-Staaten vor, erfassten Unternehmen und Behörden bei eigenentwickelten und -genutzten IKT-Produkten und -Diensten eine Zertifizierung gemäß den Schemata für Cybersicherheitszertifizierungen vorzuschreiben. Bei der Sicherheit von Netz- und Informationssystemen sollen zudem internationale Normen und technische Spezifikationen gefördert werden. Nicht in der EU ansässige von der NIS-2-Richtlinie erfasste Unternehmen müssen einen Vertreter innerhalb der EU benennen.

ENISA erstellt Register betroffener Einrichtungen

Die European Union Agency for Cybersecurity (ENISA) übernimmt die Aufgabe, ein EU-weites Register aller von der NIS-2-Richtlinie erfassten Einrichtungen im Bereich digitale Infrastruktur zu schaffen und zu pflegen, einschließlich Cloud-Anbieter, aber auch sozialer Netzwerke. Auf die Betreiber von TDL-Namensregistern und DNS-Registrierungsdienste sollen zusätzliche Pflichten zukommen. Sie müssen „genaue und vollständige Domännennamen-Registrierungsdaten in einer eigenen Datenbank im Einklang mit dem Datenschutzrecht der Union in Bezug auf personenbezogene Daten mit der gebotenen Sorgfalt sammeln und pflegen“.

Nationale Gesetze sollen es zusätzlich auch nicht von der Richtlinie erfassten Einrichtungen ermöglichen, „Informationen über Cyberbedrohungen, Beinahe-Vorfälle, Schwachstellen, Techniken und Verfahren, Kompromittierungsindikatoren, gegnerische Taktiken, bedrohungsspezifische Informationen, Cybersicherheitswarnungen und Empfehlungen für die Konfiguration von Cybersicherheitsinstrumenten zur Aufdeckung von Cyberangriffen“ auszutauschen. Aus kartellrechtlichen Gründen soll dies aber nur gelten, wenn es für das Management von Sicherheitsvorfällen erforderlich ist und der

Informationsaustausch das Cybersicherheitsniveau erhöht.

Schließlich umfasst die NIS-2-Richtlinie Anweisungen zur Aufsicht und Durchsetzung der Vorgaben. Der Katalog an Kompetenzen für die zuständigen Behörden ist lang. Er reicht von Vor-Ort-Kontrollen bei den betroffenen Einrichtungen über Sicherheitsscans bis hin zu Vorgaben für die Nachweise zur Umsetzung der Cybersicherheitskonzepte. Die betroffenen Einrichtungen sind zur umfassenden Mitwirkung zu verpflichten. Bei Rechtsverstößen sollen Warnungen und Handlungs- oder Unterlassungsanweisungen einschließlich Fristsetzungen möglich sein.

Bußgelder für Verstöße gegen die Vorgaben sollen „wirksam, verhältnismäßig und abschreckend“ sein. Bei Verstößen gegen die Pflichten im Bereich des Risikomanagements sollen Bußgelder für wesentliche Einrichtungen bei mindestens 10 Millionen Euro oder 2 Prozent des weltweiten Vorjahresumsatzes gedeckelt sein. Bei wichtigen Einrichtungen liegt die Grenze bei 7 Millionen Euro beziehungsweise 1,4 Prozent des weltweiten Vorjahresumsatzes.

EU-Maßnahmen-Portfolio

Die NIS-2-Richtlinie fügt sich in eine ganze Reihe von gesetzgeberischen Maßnahmen der EU ein. Sie ist Teil der „Gestaltung der digitalen Agenda Europas“, die sich die EU-Kommission 2020 als Vorhaben in ihrer Amtszeit bis zur Europawahl im Jahr 2024 gesetzt hat. Dazu gehören auch der derzeit diskutierte AI Act zur Regulierung des Einsatzes von künstlicher Intelligenz, der Cyber Resilience Act, der Digital Operational Resilience Act und weitere. Wegen der anstehenden Europawahl befinden sich zahlreiche Gesetzeswerke im Jahr 2023 auf der Zielgeraden (siehe [„IT-Recht 2023: Viele neue EU-Regeln“ in iX 1/2023, S. 86](#)).

Die NIS-2-Richtlinie adressiert die EU-Staaten, die die Vorgaben nun umsetzen müssen. Zahlreiche Regelungen beinhalten

daher Aufforderungen wie „Die Mitgliedsstaaten stellen sicher ...“ oder „Die Mitgliedstaaten können ...“. In Deutschland dürfte es zu einer Überarbeitung des 2021 in Kraft getretenen IT-Sicherheitsgesetzes 2.0 kommen. Dies passt auch zur Vereinbarung im Koalitionsvertrag der Bundesregierung. Dort heißt es: „Die Cybersicherheitsstrategie und das IT-Sicherheitsrecht werden weiterentwickelt.“ Dort ist auch von „ehrgeiziger Cybersicherheitspolitik“ die Rede. Die Cybersicherheit bezeichnet der Koalitionsvertrag als „digitale Schlüsseltechnologie“.

Fazit

Mit der Umsetzung der NIS-2-Richtlinie will die EU ein hohes gemeinsames Cybersicherheitsniveau sicherstellen, „um so das Funktionieren des Binnenmarkts zu verbessern“. So lautet das übergreifende Ziel. Um das zu erreichen, hat die EU den Anwendungsbereich der Richtlinie und die Aufgaben der zuständigen Behörden wesentlich erweitert und neue Behörden geschaffen. Die Mitgliedsstaaten sollen sich zudem besser austauschen und die Überwachung verschärfen.

Die ausdrücklichen Regelungen zur persönlichen Verantwortlichkeit der Leitungsorgane und die gleichzeitig signifikant erhöhten Bußgeldrahmen dürften für Zündstoff sorgen. Die EU unterstreicht damit die große Bedeutung der Cybersicherheit und reagiert auf steigende Risiken. Das Thema bleibt spannend und ein Dauerbrenner. In der EU sind nun erst einmal wieder die einzelnen EU-Staaten an der Reihe. Von der Richtlinie erfasste Unternehmen tun aber gut daran, den Gesetzgebungsprozess zu beobachten und zu begleiten. Sich früh auf anstehende Änderungen einzustellen ist für sie eine kluge Strategie. (jvo@ix.de)

1. Quellen
2. [Tobias Haar; IT-Recht 2023: Viele neue EU-Regeln; iX 1/2023, S. 86](#)

DSGVO-Bußgelder um 50 Prozent gestiegen

DSGVO-Bußgelder um 50 Prozent gestiegen

Die Bußgelder für DSGVO-Verstöße zogen im letzten Jahr deutlich an. Dabei baten die europäischen Richter Meta für Verstöße bei Facebook, Instagram und WhatsApp zur Kasse – jedoch um 4 Milliarden zu wenig, wie Datenschutzaktivist Max Schrems meint.

Im Jahr 2022 ist die Zahl der Bußgelder in der EU wegen Verstößen gegen die Datenschutz-Grundverordnung stark angestiegen. Das belegt eine Studie der internationalen Anwaltskanzlei DLA Piper. Im Vergleich zum Vorjahr sind die Bußgelder um etwa 50 Prozent auf insgesamt 1,64 Milliarden Euro gestiegen. Dazu beigetragen haben insbesondere die Bußgelder gegen Facebook in Höhe von 210 Millionen Euro und Meta in Höhe von 180 Millionen Euro. Wegen dieser Verfahren führt die hierfür verantwortliche irische Datenschutzbehörde DPC das Ranking an. Im gleichen Zeitraum hat die Zahl der von Unternehmen gemeldeten Datenpannen signifikant abgenommen.

Der bekannte Datenschutzaktivist Max Schrems hat die irische Datenschutzbehörde für einen zu laxen Umgang mit Meta kritisiert. Seiner Meinung nach hätte das verhängte Bußgeld 4 Milliarden Euro höher ausfallen müssen – die Behörde legte sich jedoch auf 390 Millionen Euro wegen DSGVO-Verstößen fest. Der Vorwurf lautete, dass Facebook rechtswidrig die gezielte Werbung gegenüber Nutzern als vertragliche Leistung ausgegeben

hat, für die keine Einwilligung erforderlich sei. Ebenso fehlte es an einer wirksamen Einwilligung für das Tracking des Nutzerverhaltens. Laut Schrems hätte die DPC bei der Bemessung des Bußgelds die zusätzlichen Werbeeinnahmen durch den Datenschutzverstoß angemessen berücksichtigen müssen.



Auch WhatsApp ist derzeit im Fokus der irischen Datenschutzaufsicht. Hier lautet der Vorwurf ebenfalls mangelnde Transparenz über die Verarbeitung von Nutzerdaten sowie fehlende Rechtsgrundlage für gezielte Werbung und Tracking. Auf ein bereits 2021 verhängtes Bußgeld in Höhe von 225 Millionen Euro folgte nun ein weiteres über 5,5 Millionen Euro. Aber nicht nur Meta steht am Pranger: Die französische Datenschutzaufsichtsbehörde CNIL hat gegen TikTok eine Strafzahlung von 5 Millionen Euro angeordnet. Ihrer Meinung nach ist es unzulässig, dass für Nutzer die Ablehnung von Cookies nicht so einfach wie die Zustimmung ist. Apple wurde schließlich für die Verwendung von Gerätedaten für personalisierte Werbung im App-Store durch die CNIL mit einem Bußgeld über 8 Millionen Euro belegt.

Unterdessen verhandelt der Europäische Gerichtshof im Bußgeldverfahren gegen die Deutsche Wohnen über Grundsatzfragen der Verhängung von Strafen nach der DSGVO. Konkret geht es darum, ob bei Datenschutzverstößen durch

Unternehmen die hierfür verantwortlichen Personen namentlich ermittelt werden müssen oder ob es ausreicht, dass ein Verstoß begangen wurde. Die Richter prüfen zudem, ob Verstöße einer Leitungsperson zuordenbar sein müssen. Während dies nach deutschem Recht erforderlich ist, sieht die DSGVO diese Voraussetzungen nicht vor. Das Verfahren ist entscheidend für die künftige Bußgeldpraxis in Deutschland. *Tobias Haar* (pst@ix.de)

Internationale ISO-Norm 31700 zu Privacy by Design in Kraft getreten

Am 8. Februar 2023 ist die ISO 31700 in Kraft getreten. Sie beschreibt Datenschutzgrundsätze von Privacy by Design im Zyklus von Produkten und Dienstleistungen, die deren Anbieter künftig vom Beginn des Entwurfs über Entwicklung und Vermarktung berücksichtigen sollen. Der erste Teil des Standards beschreibt Prinzipien der datenschutzfreundlichen Produktgestaltung. Es geht dabei um Fragen des Designs von Nutzerschnittstellen und Datenhaltung, der Kommunikation mit Verbrauchern, Risikoassessments oder Tests der Datenschutzeinstellungen.

Beispiele im zweiten Teil der Norm sollen das Verständnis für diese Standards und deren Umsetzung anschaulich verdeutlichen. Die Einhaltung des ISO-Standards 31700 ist nicht verpflichtend. Allerdings tragen solche Standards zur Weiterentwicklung des Standes der Technik insgesamt bei, der auch bei Datenschutzverstößen eine Rolle spielt. Der Download der Norm auf den Webseiten der internationalen Organisation für Normung ist gebührenpflichtig und kostet etwa 300 Euro. Der Standard konkretisiert Art. 25 DSGVO, der Datenschutz durch Technikgestaltung und durch datenschutzfreundliche Voreinstellungen fest schreibt. *Tobias Haar* (pst@ix.de)

BAG: Keine Pflicht zur Stechuhr im Homeoffice

Seit einem Beschluss des Bundesarbeitsgerichts zur

Arbeitszeiterfassung durch den Arbeitgeber wird über dessen Auswirkungen auf Arbeitszeitmodelle wie Homeoffice diskutiert. Die Erfurter Richter hatten verbindlich festgestellt, dass Arbeitgeber verpflichtet sind, „Beginn und Ende der täglichen Arbeitszeit der Arbeitnehmer zu erfassen“. Die Präsidentin des Bundesarbeitsgerichts hat nun erklärt, dass das Urteil diese Vertrauensarbeitszeitmodelle nicht abschaffe. Unter anderem hatte der Branchenverband Bitkom zuvor Zweifel hieran geäußert.

Die BAG-Präsidentin Inken Gallner stellte zudem klar, dass durch die Gerichtsentscheidung keine Pflicht zur Einführung von Stechuhren entstanden sei. „Das Wie der Arbeitszeiterfassung liegt in den gestaltenden Händen des Gesetzgebers“, so Gallner. Flexible Arbeitszeitmodelle sollen durch den Beschluss nicht abgeschafft werden. Aber auch bei solchen müsse beispielsweise die elfstündige Ruhezeit eingehalten werden. Die grundsätzliche Pflicht zur Arbeitszeiterfassung besteht bereits. Zusätzlich kündigte das Bundesarbeitsministerium an, Details der Zeiterfassung durch ein entsprechendes Gesetz regeln zu wollen. Wann dieses vorliegen und in Kraft treten wird, ist derzeit unklar. *Tobias Haar* (pst@ix.de)

EU untersucht Wettbewerbsverstöße durch Microsoft-Lizenzierungspraxis

Cispe lässt nicht locker: Die Vereinigung von Cloud-Infrastruktur-Anbietern in Europa hatte bereits im November 2022 eine Beschwerde bei der Generaldirektion Wettbewerb der EU-Kommission gegen Microsofts Bündlungspraxis von Softwareprodukten und Cloud-Services eingereicht. Jetzt legte der Verband noch einmal argumentativ nach.

Wissenschaftler der privaten Hochschulen Frankfurt School of Finance und European School of Management and Technology (ESMT) durchleuchteten im Auftrag von Cispe die ökonomischen Konsequenzen der Bündelung. In der 18-seitigen Untersuchung

kommen sie zu dem Schluss, dass Preiserhöhungen, weniger Wahlmöglichkeit und geringere Innovationstätigkeit drohen.

Hintergrund ist der Verdacht, dass Microsoft die marktbeherrschende Stellung von zum Beispiel Windows oder Office als Hebel nutzen will, um sich im Cloud-Umfeld Vorteile zu verschaffen. Mittel zum Zweck sind die Lizenzbedingungen. Danach dürfen Kunden eine bereits erworbene Software auf Ressourcen externer Rechenzentren einsetzen, soweit diese von Microsoft als autorisierte Outsourcer zertifiziert sind – Amazon AWS, Google und Microsofts Azure selbst zählen nicht zu diesem erlauchten Kreis.

Allerdings offeriert der Konzern laut Studie diverse Optionen, bestehende Lizenzen auch ohne oder nur mit geringen Zusatzkosten in Azure zu nutzen. Diese Optionen stehen für AWS oder Google so nicht zur Verfügung. Ihr Einsatz wäre folglich die kostspieligere Alternative zur Microsoft-Cloud, da in der Regel neue Lizenzen zu erwerben sind. *Achim Born* (pst@ix.de)

EU-Studie deckt Rechtsverstöße durch Dark Patterns in Onlineshops auf

Eine groß angelegte Studie von EU-Kommission und Verbraucherschutzbehörden aus 23 Mitgliedstaaten sowie Norwegen und Island hat weitverbreitete Rechtsverstöße in Onlineshops dokumentiert. Demnach setzen deren Betreiber in fast 40 Prozent der Fälle manipulative Praktiken zum Täuschen von Nutzern ein. Insgesamt wurden knapp 1400 Onlineshops untersucht. Verbreitet setzten die Anbieter laut Studie falsche Countdown-Zähler mit Fristen für den Kauf bestimmter Produkte ein. Viele Shops versuchten, Kunden durch visuelle Gestaltung oder sprachliche Mittel zum Abschluss von Abonnements zu bewegen. Bei 70 Anbietern fehlten im Kaufprozess entscheidende Informationen oder diese wurden nur versteckt zur Verfügung gestellt. Bemängelt wurden auch Zwangsregistrierungen und „virtuelle Drängel- und Gängeleien“.

Die Behörden wollen die gerügten Shopbetreiber zunächst auffordern, die Rechtsverstöße kurzfristig abzustellen. Kommen sie dem nicht nach, drohen formale Verfahren und Bußgelder. Der Bundesverband Onlinehandel kritisiert, dass mit der Studie weniger als ein Prozent aller Onlineshops überprüft wurde. Auch fehlten große Plattformen und Marktplätze. Durch das im Digital Services Act verschärfte Verbot von Dark Patterns und ähnlichen Methoden dürfte der Fokus auf solche Praktiken weiter zunehmen. Verstöße können mit Bußgeldern von bis zu 6 Prozent des Jahresumsatzes sehr teuer werden. *Tobias Haar* (pst@ix.de)

USA und EU arbeiten an gemeinsamem KI-Rahmen

Eine Verwaltungsvereinbarung zwischen den USA und der EU soll die Kooperation im Bereich des Einsatzes von künstlicher Intelligenz fördern. Ziel sei es, bestimmte Prozesse per KI zu verbessern. Gemeint sind unter anderem die Katastrophenprävention durch Vorhersage von Extremwettersituationen, aber auch die Gesundheitsvorsorge oder die Energieversorgung. Im Fokus steht dabei die gemeinsame Nutzung der vorhandenen sowie die Erschließung neuer Datenbestände im Einklang mit den geltenden datenschutzrechtlichen Bestimmungen. Ziel ist ein KI-Modell mit jeweils in der EU und den USA liegenden Daten zur gemeinsamen Nutzung. *Tobias Haar* (pst@ix.de)

Kurz notiert

Der Verbraucherzentrale Bundesverband hat neun **Telemedizin- und Arzttermin-Portale** wegen DSGVO-Verstößen abgemahnt. Die Datenverarbeitung sei intransparent und es fehlten Einwilligungen der Betroffenen.

Das Landesarbeitsgericht Köln sieht die Beweislast für den **Zugang einer E-Mail beim Absender**. Weder deren Absendung noch das Fehlen einer Unzustellbarkeitsnachricht begründen einen rechtlich relevanten Anschein für den Zugang einer E-Mail.

Die Schweiz hat zum Jahresanfang die **EU-Drohnenreglementierung** übernommen. Für Drohnen ab 250 Gramm gilt nun eine Registrierungspflicht. Piloten müssen zudem eine Onlineschulung nachweisen.

Der europäische Datenschutzausschuss EDSA sieht mögliche Rechtsverstöße bei **Cookie-Bannern**. Neben einem Akzeptieren-Button müsste es regelmäßig einen entsprechenden Ablehnen-Button geben, heißt es im Positionspapier. Auch die grafische Darstellung der Banner genügt oft nicht den Datenschutzvorgaben.



Markt + Trends | IT-Recht & Datenschutz

IT-Defense 2023 – Visionen und düstere Prognosen

IT-Defense 2023: Visionen und düstere Prognosen

Mikko Hyppönen von WithSecure legte in seiner Keynote mit dem Titel „Scorched Earth“ am zweiten Konferenztag bemerkenswerte Ansichten zur Zukunft der KI dar.

Von Jörg Riether

Auf der IT-Defense 2023 reflektierte zunächst der finnische Sicherheitsexperte und Autor Mikko Hyppönen über die Vergangenheit, in der 1997 IBMs Deep Blue den seinerzeit amtierenden Schachweltmeister Garry Kasparov schlug. In seinen Augen ist die KI-Vision seitdem und bis heute gleichermaßen großartig wie angsteinflößend. Die Geschwindigkeit entwickle sich rasant und allein in den letzten sechs Monaten sei hier mehr passiert als in den letzten 30 Jahren zusammen, so Hyppönen.

Er führte als Beispiele den Text-zu-Bild-Generator Stable Diffusion sowie die Textgeneratoren und Dialogsysteme ChatGPT und Bard an. Seine Prognose mutet unerhört an: Schon in naher Zukunft würden Computer bessere Kunst als Menschen erschaffen. Computer würden die besseren Poeten, die besseren Musiker, die besseren Programmierer und die besseren Maler sein, so Hyppönen. Das, was Computer dann produzieren, würde die Menschen tiefer und intensiver berühren als das, was ein Mensch jemals zustande bringen könnte. Er würde diese Vorstellung hassen. Es ändere aber nichts an der Realität und genau so werde es kommen, dies sei für ihn klar. Mehr noch:

Wenn die Entwicklung so weitergehe wie aktuell, könnten Computer in 100 Jahren menschliche Intelligenz stimulieren.



Hyppönens radikale Zukunftsvision: Computer werden in fast allem besser als Menschen sein (Abb. 1).

Gesprächige Tenants

Azure-AD- und Microsoft-365-Experte Nestori Syynimaa sprach

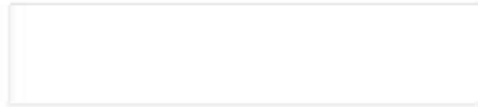
über das Vertrauen in M365-Umgebungen. In diesem Zusammenhang stellte er seine eigenen Tools vor. Diese beinhalten neben diversen PowerShell-Abfragewerkzeugen auch ein Web-GUI (siehe ix.de/zyfa), das über öffentlich zugängliche Quellen zahlreiche Tenant-Informationen einsammeln kann. Dass diese durchaus detailliert sein können, zeigt ein Versuch des Autors mit der Domain ix.de (Abbildung 2).

Es offenbart sich auf einen schnellen Klick, dass ix.de zur M365-Standarddomain heisezs.onmicrosoft.com gehört, die in der EU-Region beheimatet ist, außerdem gibt es im Tenant 19 verifizierte Domänen. Darunter gibt es auch Einträge wie „1004.ha.trunk4teams.eu“, „50-cent-und-gut.de“ sowie „heisezs.mail.onmicrosoft.com“. Die Seamless-Single-Sign-on-Technik (SSSO) ist aktiviert und zertifikatbasierte Authentifizierung (Certificate-based Authentication, CBA) ist nicht vorhanden, dies kann man mit einer gültigen Mailadresse optional prüfen.

Enter **tenant id**, **domain name**, or **email**:

ix.de

Get information



Property	Value
Default domain	heisezs.onmicrosoft.com
Tenant name	Heise Medien GmbH & Co. KG
Tenant id	30b24132-0c65-4261-ac6f-79103eb03e71
Tenant region	EU
Seamless single sign-on (SSSO)	enabled
Certificate-based authentication (CBA)	N/A
Verified domains	19

Domain	Type	STS
1004.ha.trunk4teams.eu	Managed	
1004.sbc01.4direct-routing.de	Managed	
50-cent-und-gut.de	Managed	
ct.de	Managed	
ct-fotografie.de	Managed	
duf.de	Managed	
heise.de	Managed	
heise-regioconcept.ch	Managed	
heisezs.mall.onmicrosoft.com	Managed	
heisezs.onmicrosoft.com	Managed	
hinstorff.de	Managed	
ix.de	Managed	

Nestori Syynimaas Werkzeuge haben es in sich und können sowohl zur Informationssammlung in M365-Umgebungen, wie hier bei ix.de, als auch aktiv-offensiv benutzt werden (Abb. 2). Dieses Beispiel ist noch relativ harmlos. Spannend ist, dass all diese Informationen „per Design“ öffentlich verfügbar sind. Es lohnt sich, mit dem Werkzeug selbst ein wenig mit der eigenen Unternehmensdomain oder anderen bekannten Domains zu spielen. Man könnte das Tool auch dazu missbrauchen, valide

Mailadressen herauszufinden. So gab das Web-GUI im Test bei einer vermutlich gültigen Microsoft-Mailadresse aktivierte CBA an, während es bei einer erfundenen Mailadresse aus vielen zufälligen Zeichen „nicht vorhanden“ ausgab. Es lassen sich also möglicherweise mehr Informationen ableiten, als dem einen oder anderen Unternehmen lieb sein könnte.

Auch die PowerShell-Werkzeuge sind mächtig. Neben der passiven Informationsbeschaffung gibt es hier sogar ein Phishingmodul, mit dem man live einen Angriff durchführen kann, der einen bei Erfolg direkt ins Outlook Web Access des Opfers führt. Ssynimaa macht sich hier die Azure Device Code Authentication zunutze (siehe ix.de/zyfa).

Aus dem Tesla-Nähkästchen

Der IT-Sicherheitsforscher Martin Herfurt stellte in seinem Vortrag Details zum Projekt TEMPA vor, das Werkzeuge und Details zum proprietären VCSEC-Protokoll bereitstellt. Gewonnen hat er diese Informationen durch Analyse der dekompierten offiziellen Tesla-Android-Anwendung – und konnte so Einblicke in die Angreifbarkeit des Protokolls und damit des Fahrzeugs erhalten (mehr Details siehe ix.de/zyfa).

Hurfurt berichtete, dass er die Relay-Verwundbarkeiten, über die mit zwei Raspberry Pis und Telefonen ein Tesla entwendet werden konnte (siehe ix.de/zyfa), an den Hersteller gemeldet hatte. Tesla ließ verlauten, dass man dies nicht ändern werde und die Kunden doch bitte PIN2Drive einsetzen sollen, also eine zusätzliche PIN-Abfrage, bevor man das Fahrzeug starten kann. Dazu rät auch Herfurt, denn leider seien die Relay-Angriffe auf Teslas immer noch sehr einfach möglich. (ur@ix.de)



IT-Defense 2023: Visionen und düstere Prognosen

Mikko Hyppönen von WithSecure legte in seiner Keynote mit dem Titel „Scorched Earth“ am zweiten Konferenztag bemerkenswerte Ansichten zur Zukunft der KI dar.