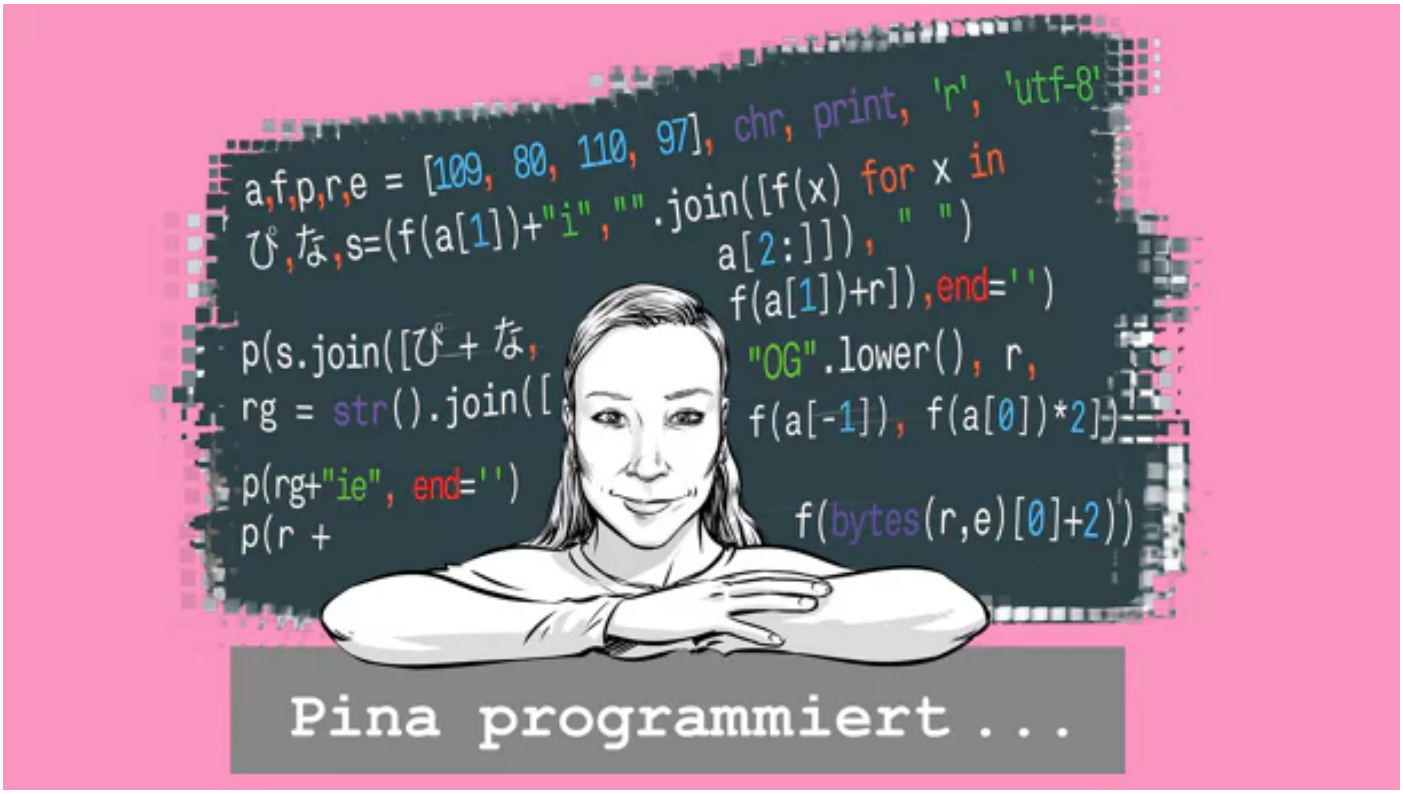
heise+

Pair-Programming mit ChatGPT: Wie KI beim Programmieren hilft

Sprach-KI kann coden. Aber programmiert sie gut genug für Software, die mehr ist als "Hallo Welt"? ChatGPT hilft bei einem Patch – mit überraschendem Ergebnis.



05.05.2023 15:00 Uhr | c't Magazin

Von Pina Merkert

INHALTSVERZEICHNIS

Der Barmann wischt durchs Glas, sein Kopf trägt das Gesicht von Michael Sheen, sein Oberkörper balanciert auf einem mechanischen Einrad. Im Film "Passengers" aus dem Jahr 2016 ist Chris Pratt als einziger Mensch auf einem Raumschiff aufgewacht, das jahrzehntelang durch die unendliche Leere des Alls gleitet. Der Barmann ist kein Mensch, sondern ein Roboter mit Gesicht, der ganz und gar menschlich daherredet. Als Zuschauer vergisst man stellenweise, dass der Roboter eine Maschine ist und tröstet sich ob der Verwechslung, dass ja menschliche Drehbuchautoren die Worte des Roboters geschrieben haben.

Zeitsprung ins Jahr 1806: Heinrich von Kleist schreibt an Otto August Rühle von Lilienstern. Der General kommt mit Meditation nicht voran und Kleist rät, mit anderen über das Thema zu sprechen. Dem Dichter geht es um "die allmähliche Verfertigung der Gedanken beim Reden": "Und siehe da, wenn ich mit meiner Schwester davon rede, welche hinter mir sitzt, und arbeitet, so erfahre ich, was ich durch ein vielleicht stundenlanges Brüten nicht herausgebracht haben würde. Nicht, als ob sie es mir im eigentlichen Sinne sagte; [...] Auch nicht, als ob sie mich durch geschickte Fragen auf den Punkt hinführte, auf welchen es ankommt, wenn schon dies letzte häufig der Fall sein mag." Das Gespräch mit der zum Thema ahnungslosen Schwester zwingt Kleist, seine Gedanken zu ordnen.

Zurück ins Jahr 2023: Ich spreche mit meinem Kollegen Jan Mahn über mein Programmierprojekt AssetStorm und wie die Software Änderungen an Texten in der Datenbank speichern könnte. Er gibt mir recht, kann beim Programmieren aber nicht helfen, weil sich zu viele andere Projekte auf seinem Schreibtisch stapeln. Ich stehe alleine da mit einer guten Idee, hätte aber schon gern Hilfe gehabt. Doch dann kommt mir eine Idee: Wenn mir kein Mensch helfen will, dann lasse ich mir eben von keinem Menschen helfen!

MEHR ZU KI-TOOLS

Ich logge mich bei OpenAI ein und fange an, mit ChatGPT zu schreiben. Die Text-KI kann kleine Programmierprobleme lösen – davon habe ich schon viele Beispiele gesehen. Aber kann sie auch helfen, wenn es schon eine umfangreiche Codebasis gibt? Den konkreten Code kennt sie nicht, also muss ich erklären, was ich schon programmiert habe. Geduldig beschreibe ich, was AssetStorm tut, wie die Software aufgebaut ist und was ich erreichen will. Das dauert ganz schön lang, weil ich meinen eigenen

Code lesen muss, um der KI präzise erklären zu können, was ich von ihr will. Den kompletten Chat in englischer Sprache gibt es als Download auf GitHub.

Dann suche ich die richtige Stelle heraus, um mit der Implementierung des neuen Features zu beginnen und füge den Code der Datei ins Chatfenster ein, markiert mit Backticks, damit ChatGPT versteht, dass das mein Quelltext ist. Nach einer Stunde tippe ich endlich eine Frage unter meine Beschreibung und schicke die Aufgabe ab: "Wie sollte ich den Python-Code verändern, um atomare Änderungs-Objekte zu bekommen?"

Sprachmodell

Unseren Test starteten wir, als das auf GPT3.5 basierende ChatGPT aktuell war. Inzwischen ist GPT4 erschienen und GitHub hat das auf dem neueren Sprachmodell basierende Copilot X veröffentlicht. Einen Chat, wie er hier beschrieben ist, würden wir heute lieber mit Copilot X in einem Fenster in der IDE Visual Studio Code führen.

Mit einem besseren Sprachmodell sind auch raffiniertere Antworten der KI möglich, sodass am vorgeschlagenen Code auch immer weniger Anpassungen nötig werden. Die Entwicklung ging bei Sprach-KI-Tools zuletzt sehr schnell. Das sind gute Nachrichten für Entwickler, weil die KI immer mehr langweilige Aufgaben übernehmen kann. Es ist aber nicht zu erwarten, dass KI-Modelle bald raffinierteren oder ästhetischeren Code schreiben können als Menschen mit Talent und Erfahrung, denn die Trainingsdaten sind weiter von Menschen geschriebene Programme.

KI-Vorschlag

Auszug aus dem Chatverlauf: Als Markdown ausgezeichneten Code hebt das Webinterface nicht hervor, ChatGPT schien den Code aber immer korrekt "verstanden" zu haben. Code in den Antworten zeigt das Chatbot-Interface mit schönem Syntax-Highlighting an.

Die KI macht sich ans Werk und schmiert mir erst mal Honig ums Maul. Ich hätte einen guten Überblick über meine Software und ich wüsste, was ich will. Ich ertappe mich dabei, dass ich mich ein wenig über die Bestätigung freue. Nach einer innerlichen Notiz, was für eine primitive Maschine ich selbst oft bin, schaue ich zu, wie sich weiterer Text in die Antwort ergießt.

In der Antwort erscheint Python-Code mit einer zusätzlichen Klasse für Datenbank-Änderungen. So zum Beispiel könne man das machen, erklärt ChatGPT und beendet die Antwort. Ich bin nicht zufrieden. Die Klasse passt nicht in mein Namensschema. Aber mich darüber zu ärgern, ist Unsinn, weil die Umbenenn-Funktion der IDE das Problem in Sekunden gelöst hätte. Aber mich stört noch mehr: Der Code ist nicht raffiniert. In der Stunde Code-Lesen für meine Frage war mir aufgefallen, wie ästhetisch mir mein eigener Code vorkam. Das Meiste war eine stilistische Präferenz, aber die hat großen Einfluss darauf, wie gern ich mit dem Code weiterarbeite. Der KI-Vorschlag sah dagegen wie eine leicht angepasste Version eines Beispiels aus einer Doku aus. So viel Code-Ästhetik wollte ich keinesfalls opfern.

Kampf gegen den Transformer

Mit dem enttäuschenden KI-Vorschlag packte mich der Ehrgeiz: Dann zeige ich jetzt der dummen KI, wie ich, ein Mensch, schöneren, schnelleren und besseren Code schreiben kann. Ich programmiere eine Alternative, verknüpfe sie mit den bisherigen Klassen und verbringe eine weitere Stunde, ohne ChatGPT irgendetwas Neues zu schreiben. Dann beschreibe ich meine Änderung, füge den Code ein und maßregele die KI, dass ich mir eigentlich eher so einen Code erwartet hätte.

ChatGPT antwortet diplomatisch, zitiert einen Teil meiner Aussagen, fasst sie ein wenig zusammen. Bei mir kommt wirklich ein bisschen das Gefühl auf, mit einer Person mit Ahnung vom Programmieren zu chatten. Es geht hin und her, ich nehme mir immer wieder Zeit, um etwas mehr meiner Idee zu implementieren. Stunden vergehen, in denen ich programmiere und mich anstrenge, um ja keine Zeile des KI-Codes übernehmen zu müssen.

Dabei fühle ich mich ChatGPT stets überlegen, aber nie so viel, dass mich die Antworten der KI nicht anspornen. Nach jeder neuen Nachricht denke ich, dass die nächste vielleicht doch schöneren oder besseren Code als meinen enthalten könnte. Eigentlich trägt die KI nichts bei, aber bis ich mich selbst davon überzeugt habe, sind zwei produktive Tage vergangen. Ohne KI wäre es sicherlich nicht schneller gegangen.

Nach zwei Tagen ist der größte Teil der Änderungen implementiert, erste Unittests laufen schon wieder. Mit dem Fortschritt bin ich zufrieden, aber mit ChatGPT nicht. Die KI hat für mich die Rolle eingenommen, die Kleist seiner Schwester zugesprochen hatte: Im Grunde kam von ihr weder eine zündende Idee noch eine wichtige Information. Aber der Austausch hat mich gezwungen, meine eigenen Gedanken zu ordnen. Wäre ich in der Lage, einer Gummiente die gleichen Informationen in gleicher Qualität zu erzählen, hätte sie einen ähnlich positiven Einfluss auf meinen Denkprozess gehabt. Aber ähnlich wie Chris Pratt in Passengers lieber mit dem Robo-Barmann redet als mit einem leeren Sessel, war es wichtig für mich, dass ChatGPT mit dem gleichen Mittel auf mich eingegangen ist, das auch ein Mensch nutzen würde: natürliche Sprache.

Codiertes Wissen

Einen Versuch wollte ich noch wagen und erzählte ChatGPT von einem Problem, auf das ich gestoßen war: Änderungen zwischen zwei Textversionen so darzustellen, dass eine sinnvolle Liste aus Einzeländerungen entsteht. Und statt eines weiteren unästhetischen Python-Blocks verwies mich die KI auf eine Bibliothek, die Google-Entwickler für Docs geschrieben und als Open Source veröffentlicht hatten. Ich kannte sie nicht und befürchtete zunächst, dass die KI sich mit einer erfundenen Bibliothek ganz billig aus der Affäre ziehen wollte.

Eine kurze Websuche nach dem Namen zeigt: Die Bibliothek gibt es wirklich! Man kennt das von Fachgesprächen mit Experten, dass einen immer Mal wieder mit einem Hinweis zu solchen existierenden Lösungen versorgen. Aber von einer KI hatte ich das nicht erwartet. Allein dieser eine Hinweis macht mein Pair-Programming-Experiment mit der KI zu einem Erfolg.

Große Transformer-Sprachmodelle sind eine revolutionäre Technik. So weit würde ich dem Hype zustimmen. Aber wie genau sie einem Menschen Arbeit abnehmen können, ist oft überraschend und ganz anders, als man zunächst gedacht hat. Deswegen kann ich nur dazu raten, die Technik selbst auszuprobieren und auch dafür offen zu sein, was die Interaktion mit der KI mit einem selbst macht. Ich werde der KI jedenfalls nicht das Programmieren überlassen. Aber vielleicht lasse ich mir auch künftig helfen, damit ich besser programmiere. (pmk)

Kommentare lesen (33)

Zur Startseite

MEHR ZUM THEMA

CHATGPT | KÜNSTLICHE INTELLIGENZ | SOFTWAREENTWICKLUNG

Forum bei heise online: Künstliche Intelligenz & Digitale Assistenten

TEILE DIESEN BEITRAG

Kurzlink: https://heise.de/-8971702

Das Beste aus heise+ »