TECHNISCHE UNIVERSITÄT DRESDEN

10.20378/irb-107576

KI-gestütztes Programmierenlernen

Erprobung und Erfahrungen eines Lehr-Lernkonzepts

Yannick Frommherz, Anne Josephine Matz, Simon Meier-Vieracker

Macht generative KI die Vermittlung von Programmierskills in der Hochschullehre überflüssig?

Über das virTUos-Teilprojekt „Experimentierraum Digitale Medienkompetenz“ erstellen wir seit 2021 Materialien zum Programmierenlernen spezifisch für Studierende aus den Geistes- & Sozialwissenschaften. Diese haben wir erfolgreich in Flipped Classroom-Seminaren erprobt.

Code bleibt das Mittel der Wahl, um Forschungsfragen & Anwendungsprobleme bedarfsgerecht & flexibel zu lösen.

Bedarf es aber noch expliziter Lehr-Lehr-Lernressourcen, um diese Vorteile auszunutzen, wenn KI perfekten Code im Handumdrehen ausgeben kann? Und wenn ja,

Wie muss die Vermittlung von Programmierskills an die neuen Rahmenbedingungen angepasst werden, die KI für die Kulturtechnik des Programmierens stiftet?

Abbildung: Ausschnitt aus unseren Lehr-Lernmaterialien, die theoretische Erklärungen mit ausführbarem Code und Übungen verweben.

Operationen bei Zeichenketten / strings

Bei einem langen Text verliert man schnell den Überblick darüber, was beim Ausführen von Code genau geschieht. Deswegen wollen wir erst mit einem simplen string arbeiten, um die Funktionsweise der einzelnen Operationen genau zu verstehen. Das erlernte Wissen übertragen wir später einfach auf die beiden Koalitionstexte. Der Beispielstring lautet:

```
sentence = "Gesagt ist gesagt."
```

Auch wollen wir fürs Erste nur ein spezifisches Wort zählen, nämlich "gesagt". Wir wissen natürlich, dass "gesagt" in `sentence` vorkommt, hätten wir aber einen längeren Text, bei dem wir uns nicht sicher sind, ob ein bestimmtes Wort vorkommt, so könnten wird dies mit dem `in` -Statement überprüfen:

```
print("gesagt" in sentence)
```

```
True
```

Dabei erhalten wir ganz einfach den Boolschen Wert `True` zurück.

✏️ **Übung 3:** Die beiden Koalitionsverträge stammen von unterschiedlichen Parteien (CDU, CSU, SPD bzw. SPD, Grüne, FDP). Find heraus, welche beiden Parteien es mit ihrem Kurznamen in beide Verträge geschafft haben.

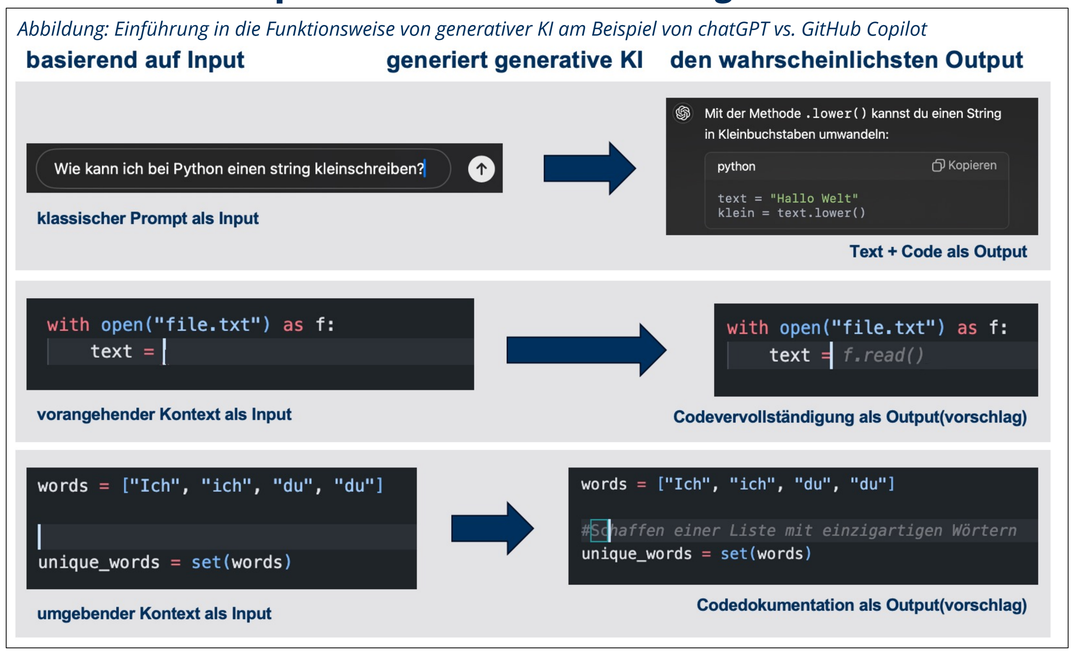
▶ 💡 Tipp

```
#In diese Zelle kannst Du den Code zur Übung schreiben.
```

Ein KI-gestütztes Lehr-Lernkonzept

1. Abfrage von Kenntnissen & Nutzungsverhalten zu generativer KI am Seminaranfang.
   - 💡 **Learning: Studierende nutzen zwar generative KI, aber nicht zum Programmieren. Gründe hierfür sind fehlende Basics für Prompting & Einschätzung von KI-Output sowie Angst vor Fehlern im KI-Code.**
2. Vermittlung der Basics anhand unserer asynchronen Selbstlernmaterialien: Objekte, Variablen, Datentypen, Kontrollstrukturen, Funktionen & Methoden etc.
3. **Einschub zu generativer KI im Anschluss an das Erlernen der Basics:**
   - **theoretischer Input zur Funktionsweise von generativer KI**

Abbildung: Einführung in die Funktionsweise von generativer KI am Beispiel von chatGPT vs. GitHub Copilot



   - **Typisierung verschiedener Anwendungsfälle: Codegeneration „from scratch“, Fehlerbehebung, Codedokumentation, Codeerklärung**
   - **Coden praktischer Anwendungsfälle dieser Typen mithilfe von *chatGPT* und der fürs Programmieren spezialisierten KI *GitHub Copilot***
   - **Gemeinsame Reflexion der Potenziale & Probleme von KI beim Coden**
4. Erlernen fortgeschrittener Inhalte (Input & Output, Datenanalyse mit `pandas` etc.) & abschließender Hackathon. Studierende wählten frei, ob sie KI einsetzen – der Dozent demonstrierte den Einsatz, wo sinnvoll, in gemeinsamen, synchronen Übungen.
5. Extern moderierte Fokusgruppen zur Evaluation der KI-Integration im Seminar.

💡 Learnings

Studierende bewerten KI-Integration ins Seminar als sehr sinnvoll.

Zeitpunkt der KI-Einführung erst nach Erlernen der Basics wird positiv bewertet.

Ohne niedrigschwelliges Programmierseminar hätten die meisten Studierenden nie mit Programmieren begonnen.

Selbstansprüche, Angst vor Fehlern & eigenem Kreativitätsverlust hindert Studierende, KI (mehr) einzusetzen.

Das Gefühl, zu betrügen steht (weiterem) Einsatz von KI im Weg.

Selbstbestimmung, ob KI eingesetzt wird, wurde geschätzt.

Interaktion mit KI wird als sehr leicht empfunden, auch auf Deutsch.

Studierende fühlen sich dank Seminar imstande, sinnvolle Prompts zu formulieren, Antworten einzuschätzen & Code in eigener IDE auszuführen – also KI zielführend einzusetzen.

Fehlende Menschlichkeit hemmt KI-Einsatz („lieber den Dozenten als die KI fragen“).

KI wird primär zur Korrektur von syntaktischen Fehlern & Codedokumentation eingesetzt.

KI-Codes sind häufig zu komplex & schaffen dadurch neue Probleme (z. B. fehlende Module).

Divergierende Einstellungen bzgl. KI-Einsatz führte bei Gruppenarbeiten zu Konflikten.

Studierende nutzen KI auch, um sich Code „herunterbrechen“ zu lassen.

Interaktion mit KI dauert z. T. länger als eigene Lösung.

Code ist eine zentrale Problemlösungsstrategie – auch in den Geistes- und Sozialwissenschaften.

Generative KI als Katalysator funktioniert nur aufbauend auf Grundkenntnissen.

Die Vermittlung von Programmierkenntnissen in der Hochschullehre ist gerade in Zeiten von KI hochrelevant.

Stiftung Innovation in der Hochschullehre
virTUos
DRESDEN concept