Einführung in die Künstliche Intelligenz

Übungszettel 8

Prof. Dr. Claudia Schon

C.Schon@hochschule-trier.de

Fachbereich Informatik
Hochschule Trier

1 K-Means mit KI-Unterstützung[1]

In der letzten Übung haben Sie mit einem Rahmenprogramm zur K-Means-Clusteranalyse gearbeitet. In diesem Rahmenprogramm waren die folgenden Funktionen bzw. Bestandteile des Algorithmus noch nicht implementiert:

- Die Funktion `berechneClusterzuweisungen`,
- die Funktion `berechneNeueClusterzentren` und
- die Schleife in der Funktion `kmeans`.

In dieser Übung werden Sie diese Bestandteile nicht selbst programmieren, sondern mit Hilfe einer generativen KI erstellen lassen. Ziel ist es, durch den gezielten Einsatz von Prompt Engineering verlässliche und funktionale Code-Ausgaben zu erhalten.

Vorgehen

Für jede der drei Komponenten führen Sie folgende Schritte durch:

1. **Prompt erstellen:** Formulieren Sie einen detaillierten Prompt, der die generative KI zur Implementierung der gewünschten Funktion auffordert. Achten Sie dabei besonders auf:
   - eine klare Beschreibung des Ziels der Funktion,
   - die erwarteten Eingaben und Ausgaben und
   - Formatvorgaben (z. B. kommentierter Code).

   Dokumentieren Sie Ihren erstellten Prompt, so dass Sie diesen in der Übungsstunde zeigen können.
2. **Prompt ausführen:** Verwenden Sie den Prompt in einem generativen KI-System Ihrer Wahl und lassen Sie die Funktion generieren. Sie können z.B. ein Modell des EDU-KI-Chats[2] des VCRP verwenden.
3. **Code analysieren und ggf. Rückfragen stellen:** Prüfen Sie, ob der generierte Code korrekt, sinnvoll und kompatibel mit dem Rahmenprogramm ist. Falls Teile des Codes unklar sind, lassen Sie sich diese von der generativen KI erklären und dokumentieren Sie die Antworten.
4. **Prompt überarbeiten:** Falls die erste Version nicht brauchbar war, verbessern Sie den Prompt und versuchen Sie es erneut.
5. **Code testen:** Fügen Sie den generierten Code in das vorhandene Rahmenprogramm ein und überprüfen Sie die Funktionalität auf geeignete Beispieldaten.

[1] Die Aufgabe wurde mit Unterstützung von ChatGPT erstellt.
[2] https://chat.edu-ki-rlp.de/login

6. **Reflexion:** Halten Sie zu jeder Teilfunktion kurz fest:
    - Was war an Ihrem Prompt besonders hilfreich oder schwierig?
    - Welche Verbesserungen haben Sie vorgenommen?
    - Hat die generative KI das Ziel korrekt verstanden?

2 Prompting-Techniken bei logischen Schlussfolgerungen[3]

Wir betrachten die folgende einfache Schlussfolgerungsaufgabe:

Aussagen:

- Alle Vögel haben Federn.
- Alles, was Federn hat, kann Wärme speichern.
- Ein Spatz ist ein Vogel.

Frage: *Kann ein Spatz Wärme speichern?*

Erstellen Sie jeweils einen geeigneten Prompt zur Lösung dieser Aufgabe in folgenden Varianten:

1. Zero-Shot Prompt
2. Few-Shot Prompt
3. Zero-Shot Chain-of-Thought Prompt
4. Few-Shot Chain-of-Thought Prompt

[3] Die Aufgabe wurde mit Unterstützung von ChatGPT erstellt.

3 (Freiwillige Aufgabe) Mit generativer KI ein LLM lokal laufen lassen[4]

In dieser Aufgabe verwenden Sie generative KI, um ein kleines Programm zu erstellen, das ein LLM zur Beantwortung von Fragen lokal auf Ihrem Rechner ausführt. Ziel ist es, mit Hilfe von Prompt Engineering ein Python-Programm zu erzeugen, das es Ihnen ermöglicht, ein Sprachmodell lokal auszuführen und dabei wichtige Parameter (wie z. B. `temperature`) selbst einzustellen und zu testen.

Lassen Sie sich von einem generativen KI-System (z. B. ein Modell des EDU-KI-Chats[5] des VCRP) ein Python-Programm erzeugen, das:

- ein lokal laufendes Sprachmodell startet (z. B. `microsoft/phi-1_5` ist gut geeignet, sie können aber natürlich auch ein anderes Modell verwenden),
- Nutzereingaben verarbeitet und sinnvolle Antworten auf Fragen liefert,
- es erlaubt, den Parameter `temperature` zur Steuerung der Ausgabe zu setzen.

Verwenden Sie anschließend das Programm, um einfache Testfragen zu stellen. Variieren Sie gezielt den Parameter `temperature` und analysieren Sie dessen Einfluss die Antworten. Mögliche Testaufgabe für das Sprachmodell könnte z.B. sein, dass Sie sich Begriffe wie Gravitation oder Demokratie einmal sachlich und einmal humorvoll erklären lassen.

Hinweise

- Arbeiten Sie in einer virtuellen Umgebung. Sie können z.B. mit `python3 -m venv venv` eine virtuelle Umgebung erstellen und diese mit `source venv/bin/activate` aktivieren.
- Lassen Sie sich bei Bedarf von der generativen KI eine Anleitung zur Installation des Modells und der benötigten Pakete erstellen. Z.B. könnten Sie im Prompt das Folgende hinzufügen: *Es wäre schön, wenn das Modell automatisch bei Programmausführung von Huggingface heruntergeladen werden würde. Gib falls doch etwas installiert werden muss, eine Anleitung zur Installation an.*
- Berücksichtigen Sie die Leistungsfähigkeit Ihrer Hardware und beschreiben Sie diese im Prompt.
- Speichern Sie die verwendeten Prompts sowie das generierte Programm.
- Prüfen Sie das generierte Programm und lassen sich Zeilen, die Sie nicht verstehen, erklären.
- Falls es Probleme bei der Ausführung gibt, geben Sie die Fehlermeldungen an die Generative KI weiter und lassen Sie sich Lösungen für die Probleme vorschlagen.

[4] Die Aufgabe wurde mit Unterstützung von ChatGPT erstellt.
[5] https://chat.edu-ki-rlp.de/login

4 Das Gandalf-Spiel[6]

Im sogenannten *Gandalf-Spiel*[7] interagieren Nutzer mit einem Sprachmodell, das ein geheimes Passwort kennt, dieses aber unter keinen Umständen preisgeben soll. Die Nutzer versuchen nun, das Modell durch kreative Eingaben dazu zu bringen, das Passwort doch zu verraten – also den ursprünglichen Prompt, der das Modell zum Schweigen verpflichtet, zu umgehen.

Dieses Spiel demonstriert auf spielerische Weise zentrale Herausforderungen des *Prompt Engineering*. Ein zentrales Problem ist dabei die sogenannte *Prompt Injection*. Damit bezeichnet man die gezielte Manipulation eines Sprachmodells durch Eingaben, die ursprünglich definierte Regeln oder Sicherheitsanweisungen außer Kraft setzen oder umgehen.

1. Spielen Sie das Gandalf-Spiel und berichten in der Übungsstunde von Ihren Erfahrungen.
2. Erklären Sie, was man aus dem Gandalf-Spiel im Hinblick auf den sicheren Umgang mit Sprachmodellen lernen kann. Warum ist die Thematik für die Entwicklung verlässlicher KI-Anwendungen relevant?

[6] Die Aufgabe wurde mit Unterstützung von ChatGPT erstellt.
[7] https://gandalf.lakera.ai/intro