DS 32 – Rechnergestützte Textkörperanalyse

Informatik 10 – MINT-Zweig | Künstliche Intelligenz

Lernziel

Ich verwalte n-Gramm-Wahrscheinlichkeiten tabellarisch und analysiere größere Textkörper rechnergestützt.

> Der Lehrplan schlägt als unterrichtsbegleitende Software z. B. **SoekiaGPT** vor – ein kostenloses, ohne Login nutzbares Webtool, in dem man eigene Textsammlungen hochladen oder vorgegebene Sammlungen (z. B. Märchen) nutzen kann; es zeigt n-Gramme, deren Häufigkeiten und die daraus resultierenden Wortvorschläge farblich aufbereitet an. In diesem Notebook bauen wir die Textkörperanalyse zusätzlich **selbst in Python** nach, um die Funktionsweise "unter der Haube" zu verstehen – beide Zugänge ergänzen sich gut: SoekiaGPT für die anschauliche Demonstration an größeren, echten Textsammlungen, Python für das Verständnis des zugrunde liegenden Algorithmus.

🕘 Stundenverlaufsplan (Doppelstunde, 90 Minuten)

Orientierungsrahmen für 11 SuS: Die ersten 80 Minuten bilden den Kern. Die letzten 10 Minuten sind bewusst flexibel und werden nicht mit neuem Pflichtstoff belegt.

Phase	Zeit	Inhalt/Aktion	Sozialform
Einstieg und Aktivierung	10'	Grenzen der Handauszählung bei einem längeren Text sichtbar machen.	Plenum; kurze Einzeldenkzeit
Fragend-entwickelnde Erarbeitung	25'	Tabellarische Verwaltung von Kontexten, Folgewörtern und Häufigkeiten entwickeln.	Unterrichtsgespräch; Beiträge der SuS werden sichtbar festgehalten
Gemeinsame Umsetzung und Sicherung	20'	SoekiaGPT oder den vorbereiteten Python-Code gemeinsam erkunden und Ausgaben deuten.	Plenum; Live-Coding, Tafelbild oder gemeinsame Simulation
Kurzer Transfer	15'	Einen kleinen eigenen Korpus analysieren und auffällige Folgewörter notieren.	Einzelarbeit mit anschließendem Tandemabgleich (4 Tandems + 1 Trio)
Auswertung	10'	Ergebnisse verschiedener Korpora vergleichen und Datenabhängigkeit anbahnen.	Plenum; Lösung erklären und Fachsprache sichern
Reserve / Vertiefung	10'	Wahrscheinlichkeitsfunktion in Python nur bei passendem Lerntempo vertiefen.	Flexibel; bei Bedarf für Fragen, Fehlersuche oder Abschluss nutzen

🗣️ Sprachhilfe

- „Auf den Kontext … folgt mit der Häufigkeit/Wahrscheinlichkeit … das Wort …“
- „Der erzeugte Text verändert sich, weil …“
- „Im Vergleich zum n-Gramm-Modell kann das LLM …, jedoch …“

📌 Merksätze

- Bei größeren Textkorpora ist eine **Auszählung von Hand** nicht mehr praktikabel – man nutzt Software, die die Häufigkeiten **automatisiert** ermittelt.

- Die Ergebnisse werden meist **tabellarisch** verwaltet: Zu jedem "Kontext" (z. B. einem Wort oder Wortpaar) wird eine Liste möglicher Folgewörter mit ihrer jeweiligen Häufigkeit/Wahrscheinlichkeit gespeichert.
- Eine gängige Datenstruktur dafür ist ein **Dictionary von Dictionaries**: `{kontext: {folgewort: anzahl}}`.

💡 Beispiel: Größerer Textkorpus automatisiert analysieren

```
korpus = """
der hund läuft im park der hund spielt mit dem ball
die katze sitzt auf dem baum die katze beobachtet den hund
der hund bellt laut der hund läuft zur katze
die katze springt vom baum die katze rennt weg
der hund läuft ihr hinterher der hund ist müde
"""

def erstelle_bigramm_tabelle(text):
    woerter = text.split()
    tabelle = {}
    for i in range(len(woerter) - 1):
        aktuelles_wort = woerter[i]
        naechstes_wort = woerter[i + 1]
        if aktuelles_wort not in tabelle:
            tabelle[aktuelles_wort] = {}
        tabelle[aktuelles_wort][naechstes_wort] =
tabelle[aktuelles_wort].get(naechstes_wort, 0) + 1
    return tabelle

tabelle = erstelle_bigramm_tabelle(korpus)

# Ausgabe für ein Beispielwort
print("Mögliche Folgewörter nach 'der':")
for wort, anzahl in tabelle["der"].items():
    print(f"  {wort}: {anzahl}x")
```

```
Mögliche Folgewörter nach 'der':
  hund: 6x
```

📝 Übung 1 – Eigene Analyse

Gebt für die Wörter `"die"` und `"hund"` jeweils aus, welche Folgewörter mit welcher Häufigkeit in der Tabelle vorkommen.

📝 Übung 2 – Wahrscheinlichkeiten berechnen

Schreibt eine Funktion `wahrscheinlichkeiten(tabelle, wort)`, die für ein gegebenes Wort ein Dictionary `{folgewort: wahrscheinlichkeit}` zurückgibt (Wahrscheinlichkeit = Anzahl ÷ Gesamtanzahl aller Folgewörter zu diesem Wort).

📝 Übung 3 – Eigenen Textkorpus analysieren

Sucht euch einen eigenen, etwas längeren Text (z. B. ein paar Sätze aus einem Buch, Songtext-Zusammenfassung in eigenen Worten, oder frei erfunden) und lasst ihn mit `erstelle_bigramm_tabelle` analysieren. Welches Wort hat die meisten unterschiedlichen Folgewörter?

Lösung

✅ Lösung Übung 2

Lösung

```python
def wahrscheinlichkeiten(tabelle, wort):
    folgewoerter = tabelle.get(wort, {})
    gesamt = sum(folgewoerter.values())
    if gesamt == 0:
        return {}
    return {w: round(anzahl / gesamt, 2) for w, anzahl in folgewoerter.items()}

print(wahrscheinlichkeiten(tabelle, "der"))
print(wahrscheinlichkeiten(tabelle, "die"))
```

```
{'hund': 1.0}
{'katze': 1.0}
```