

Teil 4: Intelligentes Üben – Von der Automatisierung zur Mustererkennung

Traditionelle Arbeitsblätter fördern oft nur blindes Rechnen. Mit KI lassen sich Übungsreihen generieren, die durch gezielte Variation mathematische Strukturen offenlegen und tiefes Verständnis verankern.

Nachdem wir in den ersten drei Teilen Einstiege, Elementarisierung und Diagnostik beleuchtet haben, erreichen wir die Phase, die im Schulalltag oft die meiste Zeit beansprucht: die Festigung. Herkömmliche Aufgabensammlungen in Schulbüchern tendieren dazu, eine Vielzahl von Aufgaben mit willkürlich wechselnden Zahlen und sofortigen Sprüngen im Schwierigkeitsgrad (oder dem plötzlichen Einweben von Textaufgaben) anzubieten. Das Resultat ist oft ein Übersehen tieferer mathematischer Strukturen. Lernende arbeiten Aufgaben algorithmisch ab, ohne Zusammenhänge zu erkennen.

Durch strukturiertes Prompting können wir KI nutzen, um Übungsmaterial zu erstellen, das dem Prinzip des *Purposeful Practice* (zweckmäßiges Üben) folgt. Zwei Konzepte stehen dabei im Fokus: Die Variationstheorie und systematische Lückenformate.

Die Variationstheorie: Minimale Änderungen, maximaler Fokus

Die Variationstheorie besagt, dass Lernen dann am effektivsten stattfindet, wenn wir nur ein einziges kritisches Merkmal eines Objekts verändern, während alle anderen Merkmale konstant bleiben ("Minimally different examples"). Wenn sich zwischen zwei Aufgaben nur ein Vorzeichen, ein Faktor oder die Position einer Variablen ändert, wird die Aufmerksamkeit der Lernenden zwingend auf die Auswirkung dieser spezifischen Änderung gelenkt.

Für Lehrkräfte ist es kognitiv anstrengend und zeitaufwendig, solche engmaschigen Aufgabenreihen händisch zu konstruieren. Sprachmodelle können diese Systematik jedoch mühelos generieren, wenn sie entsprechende Leitplanken erhalten.

PROMPT-AUSZUG: MINIMAL VARIIERTE AUFGABENREIHEN

Kontext: Du entwirfst Übungsmaterial für eine 8. Klasse.

Thema: Lösen linearer Gleichungen mit zwei Rechenschritten.

Leitplanken:

1. Erstelle eine Sequenz von exakt 6 Aufgaben.
2. Wende das Prinzip der "Minimally different examples" an: Verändere von Aufgabe zu Aufgabe immer nur EIN einziges Detail (z. B. ein Rechenzeichen, einen Koeffizienten, oder die Seite, auf der das x steht).
3. Das Ziel ist es, dass die Schüler erkennen, wie sich diese kleine Änderung strukturell auf den Lösungsweg und das Ergebnis auswirkt.

Ausgabeformat:

Gib den LaTeX-Code für die 6 Aufgaben aus.

Füge danach eine Tabelle an, in der für jede Aufgabe (ab Aufgabe 2) dokumentiert wird: Was genau wurde im Vergleich zur vorherigen Aufgabe geändert und welche mathematische Erkenntnis soll diese Änderung provozieren?

Eine solche Sequenz könnte beispielsweise so aussehen:

1. $2x + 4 = 10$
2. $2x - 4 = 10$ (Fokus: Addition statt Subtraktion im ersten Schritt)
3. $-2x - 4 = 10$ (Fokus: Umgang mit negativem Koeffizienten)
4. $10 = -2x - 4$ (Fokus: Symmetrie der Gleichung / Variable rechts)

Die Lernenden arbeiten nicht mehr nur als "Rechenmaschinen", sondern als Mustererkennungssysteme.

Zweckmäßiges Üben: Lückentabellen (Completion Tables)

Neben der Variation von Reihen ist das Vorwärts- und Rückwärtsarbeiten eine zentrale Kompetenz. Herkömmliche Aufgaben erfordern meist nur die lineare Ausführung eines Algorithmus (Input → Output). *Completion Tables* zwingen Lernende, die Richtung der Transformation ständig zu wechseln.

Eine Tabelle listet dabei verschiedene Eigenschaften eines mathematischen Objekts in Spalten auf. In jeder Zeile ist eine andere Eigenschaft gegeben, und die fehlenden Zellen müssen ergänzt werden. KI eignet sich perfekt, um solche Tabellen so zu berechnen, dass die fehlenden Werte mathematisch sauber (z. B. ganzzahlig) bleiben.

Beispiel: Completion Table für quadratische Funktionen

Spalten: 1. Normalform | 2. Scheitelpunktform | 3. Scheitelpunkt (Koordinaten) | 4. y-Achsenabschnitt.

Die KI kann angewiesen werden, in Zeile 1 die Normalform vorzugeben. In Zeile 2 ist nur der Scheitelpunkt gegeben (Schüler müssen die Funktionen rekonstruieren). In Zeile 3 ist die Scheitelpunktform gegeben. Der stetige Wechsel der Ausgangsinformation verhindert ein rein mechanisches Abarbeiten und überprüft das tiefere Begriffsverständnis.

Automatisierung: Shed Loads of Practice (SLOP)

Auch wenn Strukturverständnis das primäre Ziel ist, gibt es Phasen, in denen reine Automatisierung und Geläufigkeit (Fluency) erforderlich sind – zum Beispiel beim Kürzen von Brüchen oder beim Auflösen von Klammern. Im britischen Raum wird dies pragmatisch als *SLOP (Shed Loads of Practice)* bezeichnet.

Auch hier hilft die KI: Mit einem gezielten Prompt lassen sich 20 reine Routineaufgaben generieren, die den Schwierigkeitsgrad nur minimal anheben, keine ablenkenden Textaufgaben (Kontext) enthalten und direkt mit vollständigen Lösungs- und Rechenwegen im LaTeX-Format ausgespuckt werden. Die Lehrkraft spart hier primär Arbeitszeit, die sie in die Diagnostik oder individuelle Förderung investieren kann.

Fazit: Qualität statt Quantität

Der Mehrwert von KI in der Übungsphase liegt nicht in der unendlichen Generierung zufälliger Arbeitsblätter, sondern in der bewussten Steuerung des Aufgabendesigns. Ob durch minimale Variationen zur Mustererkennung oder durch Tabellen zum multidirektionalen Denken – der Fokus verschiebt sich von der Quantität der gelösten Aufgaben auf die Qualität der kognitiven Auseinandersetzung.

Im fünften und letzten Teil unserer Serie widmen wir uns der Königsdisziplin: Dem Problemlösen. Wir betrachten, wie Aufgabenformate wie "Same Surface, Different Deep" (SSDD) oder Aufgaben ohne Zahlen ("No Number problems") dabei helfen, die Problemlösekompetenz zu schulen und die kognitive Last gezielt zu steuern.