



Hochschule für Technik,
Wirtschaft und Kultur Leipzig

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Studiengang Informatik
der Fakultät Informatik und Medien
der Hochschule für Technik, Wirtschaft und Kultur Leipzig

Einsatz von Large Language Models zur automatisierten Erstellung mathematischer Übungsaufgaben

vorgelegt von
Ines Rohrbach

Leipzig, den 31. August 2024

Erstgutachter:	Prof. Dr. rer. nat. habil. Martin Grüttmüller (HTWK Leipzig)
Zweitgutachter:	M.Sc. Alexander Pögelt (HTWK Leipzig)

OFFENLEGUNGSVERMERK

Diese Arbeit © 2024 von Ines Rohrbach ist unter CC BY-SA 4.0 lizenziert. Eine Kopie dieser Lizenz finden Sie unter <https://creativecommons.org/licenses/by-sa/4.0/>.

This work © 2024 by Ines Rohrbach is licensed under CC BY-SA 4.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by-sa/4.0/>.

DANKSAGUNG

Vielen Dank an Frau Dr. Katja Ihsberner für das Evaluieren der generierten Übungsaufgaben und die Bereitschaft zu helfen. Danke an meine Betreuer Herr Prof. Dr. Martin Grüttmüller und Alexander Pögelt für die Möglichkeit, diese Arbeit zu schreiben.

Besonderen Dank an alle, die mich während dieser Arbeit und auch auf meinem Weg begleitet und unterstützt haben. Und an diejenigen, die diesen Moment nicht miterleben können.

KURZFASSUNG

Large Language Models (LLMs) bieten viele Möglichkeiten, die Lehre zu bereichern. In dieser Arbeit wird betrachtet, inwiefern mathematische Übungsaufgaben von LLMs automatisiert generiert und die Qualität evaluiert werden können. Dafür wurde eine Anwendung implementiert, welche unter Einsatz von GPT-4o Übungsaufgaben zu einem gewählten mathematischen Konzept erstellt und die kognitive Prozess- und Wissensdimensionen „Erinnern“ und „Faktenwissen“ nach Anderson und Krathwohl anspricht. Die Aufgaben wurden anschließend maschinell klassifiziert und nach verschiedenen Qualitätsmerkmalen evaluiert, bevor die Aufgaben für eine leichte Bereitstellung an die Lernenden in das Question & Test Interoperability (QTI) Format konvertiert wurden. Für die Betrachtung der Leistung der Anwendung wurden im Rahmen dieser Arbeit insgesamt 364 Aufgaben generiert, von denen 144 von einer Mathematik-Expertin evaluiert wurden. Die Ergebnisse zeigen, dass 60 % der Aufgaben von hoher Qualität waren. Allerdings eignet sich die maschinelle Evaluation mit einer Accuracy von 0.63 nicht, um die Evaluation vollständig zu automatisieren. Dennoch deutet die Precision von 0.76 daraufhin, dass Potenzial besteht, die automatisierte Aufgabenbewertung weiter zu verbessern.

Schlagwörter: Large Language Models, Aufgabengenerierung, GPT-4o, QTI, Aufgabenbewertung

ABSTRACT

Large Language Models (LLMs) offer numerous opportunities to enhance education. This work explores the potential for automating the generation of mathematical exercises and evaluating their quality using LLMs. To achieve this, an application was developed utilising GPT-4o to generate exercises based on selected mathematical concepts. The generated tasks target the cognitive process and knowledge dimensions of “Remember” and “Factual Knowledge” as defined by Anderson and Krathwohl. After generation, the tasks were automatically classified and evaluated based on various quality criteria. The exercises were then converted into the Question & Test Interoperability (QTI) format to facilitate distribution to students. To assess the performance of the application, a total of 364 tasks were generated for this study, with 144 of them being evaluated by a mathematical expert. The results indicate that 60 % of the tasks were of high quality. However, the machine evaluation’s accuracy of 0.63 suggests it is not yet suitable for fully automating the evaluation process. Despite this, a precision of 0.76 demonstrates potential for further development in automated task assessment.

Keywords: Large Language Models, Task Generation, GPT-4o, QTI, Task Assessment

Inhaltsverzeichnis

Kurzfassung, Abstract	i
Inhaltsverzeichnis	ii
Abkürzungsverzeichnis	iv
1 Einleitung	1
2 Wissenschaftlicher Kontext	3
2.1 tech4compKI	3
2.2 Klassifizierung von Aufgaben	3
2.3 Question & Test Interoperability Standard	5
2.4 Large Language Model	6
2.5 Stand der Forschung	7
3 Anforderungen	9
3.1 Aufgaben	9
3.2 Qualität und Klassifizierung	9
3.3 Format	10
4 Konzept	11
4.1 Large Language Model	11
4.1.1 Generative Pre-Trained Transformer	11
4.1.2 Spezialisierte LLMs	12
4.1.3 Vergleich der LLMs	13
4.2 Question & Test Interoperability Standard	14
4.3 Methodik	16
4.3.1 Maschinelle Evaluation der Qualität	17
4.3.2 Maschinelle Evaluation der Taxonomie	17
4.3.3 Menschliche Evaluation der Aufgaben	17
5 Implementierung	19
5.1 Struktur	19
5.2 Benutzeroberfläche	20
5.3 Aufgaben in JSON	21

5.4	Generierung der Aufgaben	23
5.5	Maschinelle Evaluation	24
5.6	Umwandlung von JSON zu QTI	26
5.7	Docker	27
6	Evaluation	28
6.1	Bewertungsmetriken	28
6.2	Phase 1	29
6.3	Phase 2	31
6.4	Diskussion	34
7	Zusammenfassung	37
	Abbildungsverzeichnis	39
	Tabellenverzeichnis	40
	Quelltextverzeichnis	41
	Literaturverzeichnis	42
A.1	LLM-Tests	49
A.2	Beispiele generierter Übungsaufgaben	59
	Selbstständigkeitserklärung	76

ABKÜRZUNGSVERZEICHNIS

API	Application Programming Interface
CoT	Chain-of-Thought
CSV	Comma-Seperated Values
FAISS	Facebook AI Similarity Search
FN	False Negative
FP	False Positive
HTWK	Hochschule für Technik, Wirtschaft und Kultur
JSON	JavaScript Object Notation
KI	Künstliche Intelligenz
LLM	Large Language Model
LMS	Lernmanagementsystem
MMR	Maximal Marginal Relevance
NLP	Natural Language Processing
OPAL	Online-Plattform für Akademisches Lehren und Lernen
QTI	Question & Test Interoperability
TN	True Negative
TP	True Positive
XML	Extensible Markup Language

1. EINLEITUNG

Large Language Models (LLMs) spielen eine zunehmend wichtigere Rolle in verschiedenen Anwendungsbereichen und bieten ein breites Spektrum an Möglichkeiten. Dazu zählt auch die Lehre, in der sie unter anderem ermöglichen, die Lernerfahrung zu bereichern und Studierende individuell zu unterstützen. Für Lehrende liegt der Vorteil beispielsweise darin, arbeits- und zeitaufwendige Prozesse zu verringern oder vollständig zu automatisieren, wodurch ihnen mehr Raum für persönliche Betreuung und Mentoring gewährt wird.

Eine besonders zeitaufwendige Aufgabe ist die Erstellung von Übungsaufgaben. Um sicherzustellen, dass für jeden Lern- und Wissensstand passende Aufgaben bereitgestellt werden können, ist ein umfangreicher Aufgabenpool notwendig. Die Erstellung eines solchen Pools erfordert jedoch nicht nur viel Zeit, sondern auch erhebliche Ressourcen.

Im Jahr 2023 wurde im Rahmen des *tech4compKI*-Verbundprojekts in einer Publikation von Meissner, Pögelt et al. [39] erforscht, wie LLMs zur automatisierten Generierung von Übungsaufgaben in der Hochschullehre im Bereich der Mathematik eingesetzt werden können. Die Studie erstellte Aufgaben für jede Taxonomiestufe von erlangten Kompetenzen, nach Anderson und Krathwohl [3], und evaluierte deren Qualität und Klassifikation durch Experten. Die Ergebnisse zeigten, dass das verwendete LLM in der Lage war, Aufgaben mit hoher Präzision in den kognitiven Prozess- und Wissensdimensionen *Anwenden* und *Konzeptionelles Wissen* zu erstellen. Allerdings erwies sich die Generierung in den unteren Dimensionen *Erinnern* und *Faktenwissen* als weniger zuverlässig.

Um die Lehrenden weiter zu entlasten und den Einsatz von LLMs zur Aufgabengenerierung in der Bildung zu optimieren, ist es wichtig, die maschinelle Evaluation der Aufgaben zu erforschen. Diese könnte automatisiert die Qualität und Klassifikation der kognitiven Prozess- und Wissensdimensionen der Aufgaben bewerten.

Darüber hinaus sollten die generierten Aufgaben in einem standardisierten Format wie *Question & Test Interoperability* (QTI) bereitgestellt werden. Dieses ist weit verbreitet und ermöglicht es, Aufgaben direkt in verschiedene Lernmanagementsysteme (LMS) zu integrieren, welche von vielen Bildungseinrichtungen genutzt werden. QTI könnte es den Lehrenden erleichtern, die Aufgaben ohne zusätzliche Konvertierungsprozesse direkt in ihre Kurse zu übernehmen und den Studierenden zur Verfügung zu stellen.

Daher ist das Ziel dieser Bachelorarbeit, an der Publikation von Meissner, Pögelt et al. [39] anzuschließen und einen Aufgabengenerator zu entwickeln, der unter Einsatz von LLMs mathematische Übungsaufgaben im QTI-Format erstellt und diese anschließend automatisiert evaluiert. Dazu soll erforscht werden, inwiefern eine maschinelle Bewertung der Aufgabenqualität und deren Einordnung in die Taxonomie mit der menschlichen Wahrnehmung übereinstimmt.

Aufbau der Arbeit Im ersten Teil der Arbeit, in Kapitel 2, wird grundlegend das Projekt tech4compKI, die verschiedenen Aspekte der automatisierten Generierung von Übungsaufgaben im Kontext dieser Arbeit und der aktuelle Forschungsstand vorgestellt. In Kapitel 3 werden für das Erstellen der Aufgaben Anforderungen an die Aufgaben, deren Qualität, Klassifizierung und Format aufgestellt. Darauffolgend wird im Kapitel 4 das Konzept erläutert, indem untersucht wird, welches LLM sich für diese Arbeit eignet. Außerdem wird betrachtet, ob Aufgaben direkt im QTI-Format generiert oder wie diese andernfalls umgewandelt werden können. Anschließend wird die Methodik der verschiedenen Evaluationen dargestellt. Im Weiteren befasst sich Kapitel 5 mit der technischen Umsetzung der Anwendung. Das Kapitel 6 beinhaltet zunächst die Vorstellung der Bewertungsmetriken, bevor die Ergebnisse evaluiert und abschließend diskutiert werden. Schließlich werden in Kapitel 7 die Arbeit und ihre Ergebnisse zusammengefasst und ein Ausblick für zukünftige Arbeiten gegeben.

2. WISSENSCHAFTLICHER KONTEXT

In diesem Kapitel wird ein Überblick über das Konzept der Klassifizierung der Aufgaben und deren Format, insbesondere im Rahmen des Verbundprojekts *tech4compKI*, gegeben. Außerdem werden LLMs grundlegend erläutert und der aktuelle Forschungsstand im wissenschaftlichen Kontext diskutiert.

2.1 tech4compKI

Diese Arbeit wird im Rahmen des tech4compKI-Verbundprojekts (ehemals *tech4comp*) geschrieben. Dies ist ein Zusammenschluss deutscher Universitäten und Hochschulen, welche gemeinsam seit Oktober 2018 an einer mentoriellen Unterstützung für Lehrende und Studierende forschen. Das Projekt wird durch das Bundesministerium für Bildung und Forschung gefördert und fokussiert sich in der zweiten Förderphase, in der diese Arbeit inbegriffen ist, besonders auf die Nutzung von künstlicher Intelligenz (KI) bei der Konzeption individuellen Mentorings, um die Vorteile dessen für viele Studierende verfügbar zu machen [58, 62].

Zentral ist die Entwicklung einer hybriden, technologiegestützten Lernumgebung, welche durch die Unterstützung von KI für große Mengen von Lernenden skalierbar ist. Die Forschung basiert unter anderem auf Daten aus Lernmanagementsystemen (LMS) und Sensoren, welche freiwillig von Studierenden und Lehrenden bereitgestellt werden [62].

In Abb. 1 ist die Struktur der Microservices des Bildungsnetzwerks dargestellt. Dieses basiert auf einem Kubernetes-Cluster. Als Data Proxies werden verschiedene Datenquellen integriert, zum Beispiel Sensoren zur Ermittlung des Stresslevels der Studierenden, aber auch Lernplattformen wie die Online-Plattform für Akademisches Lehren und Lernen (OPAL), welche unter anderem an der Hochschule für Technik, Wirtschaft und Kultur (HTWK) Leipzig verwendet wird. Über OPAL können Lehrmaterialien verteilt und Aufgaben bereitgestellt sowie ausgewertet werden. In Abb. 1 orange gefärbte Services werden von KI unterstützt, welche den in dieser Arbeit behandelten *Task Generator* und den *Task Recommender* beinhalten. Der Recommender empfiehlt den Studierenden passend zu ihrem Lern- und Wissenstand sowie Hilfebedarf Aufgaben, welche unter anderem vom Generator erstellt wurden.

2.2 Klassifizierung von Aufgaben

Im Rahmen des tech4compKI-Projekts wird die Lehre nach der Art des *Constructive Alignment* von Biggs [8] gestaltet. Dieses Konzept beschreibt das Abstimmen von Lehrmaterial, Lernaktivitäten und Prüfungen mit klar formulierten Lernzielen. Dementsprechend erfordert das Bestreben, Verständnis bzw. bestimmte Kompetenzen zu vermitteln, dass

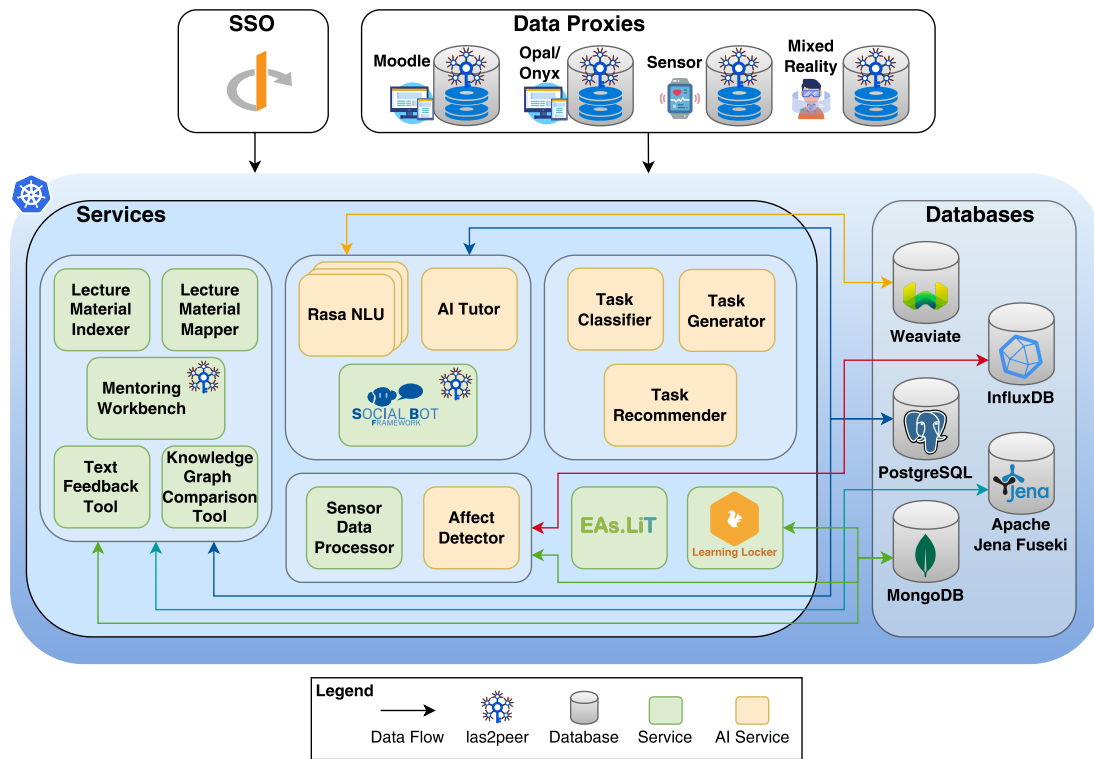


Abb. 1: Übersicht der tech4compKI-Services, Darstellung aus [43]

die Lehre und besonders die Aufgaben eben diese Kompetenzen ansprechen. Bezogen auf Übungsaufgaben ist es daher erforderlich, dass die adressierten Fähigkeiten dieser bestimmt werden und betrachtet wird, inwieweit sie sich für die Evaluation der studentischen Leistung eignen. Mit der Bearbeitung der Aufgaben weisen die Studierenden nach, ob die erlangten Kompetenzen mit den intendierten Lernzielen korrespondieren [8].

Studien weisen darauf hin, dass Verstehen einer hierarchischen Struktur folgt [8], weshalb Taxonomien für die Operationalisierung eingesetzt werden. 1956 veröffentlichte Bloom eine Taxonomie [9], welche die Fertigkeiten und Fähigkeiten zum Erreichen von Lernzielen klassifiziert. Diese beschreibt ursprünglich sechs *kognitive Prozessdimensionen* mit steigender Komplexität: *Wissen*, *Verständnis*, *Anwendung*, *Analyse*, *Synthese* und *Evaluation*. 2001 wurde die Taxonomie, basierend auf Erkenntnissen in den Kognitionswissenschaften, von Anderson und Krathwohl [3] überarbeitet. Diese Revision verändert die Nomenklatur und beschreibt die Synthese als höchste Stufe der Komplexität: *Erinnern*, *Verstehen*, *Anwenden*, *Analysieren*, *Beurteilen* und *Erschaffen*. Es wird ebenfalls eine weitere Dimension eingeführt, welche, zusätzlich zu den kognitiven Fähigkeiten, vier *Wissensdimensionen* unterscheidet: *Faktenwissen*, *Konzeptionelles Wissen*, *Prozedurales Wissen* und *Metakognitives Wissen*. Im tech4compKI-Projekt wird mit der Taxonomie nach Anderson und Krathwohl

gearbeitet, da dieses von den Richtlinien der *Hochschulrektorenkonferenz* befürwortet wird [22, 39].

Hieraus ergibt sich, dass es für den Aufbau einer Lehrveranstaltung nach dem Prinzip des Constructive Alignments bedeutsam ist, die Lernziele des Moduls präzise zu formulieren und an die Taxonomie anzupassen. So können Aufgaben, die ebenfalls nach der Taxonomie ausgerichtet sind, die erlangten Kompetenzen widerspiegeln. An der HTWK Leipzig werden Lernaktivitäten, wie beispielsweise Aufgaben, unter anderem über OPAL bereitgestellt und sollen vom *Recommender* zugeteilt werden. Damit der Recommender für jeden Lern- und Wissensstand Aufgaben vermitteln kann, müssen Aufgaben für alle Dimensionen zur Verfügung stehen. Aufgrund des hohen Zeitaufwands beim Prozess, eine große Anzahl an Aufgaben manuell bereitzustellen, ist ein Hauptanliegen dieser Arbeit, Übungsaufgaben mittels LLMs automatisiert zu generieren. Damit die Aufgaben verwendet werden können, müssen diese ebenfalls korrekt klassifiziert werden.

2.3 Question & Test Interoperability Standard

Die erstellten Aufgaben sollen über das LMS OPAL zugänglich sein. Dieses verwendet die Prüfungsplattform *ONYX*. Darüber können Aufgaben bereitgestellt, von Studierenden gelöst und größtenteils automatisiert kontrolliert werden. Über OPAL steht Lehrenden der *ONYX*-Editor zur Verfügung, mit dem Aufgaben und Tests, das heißt zusammengestellte Aufgabensets, erstellt, angepasst, geteilt und importiert werden können.

Die Aufgaben von *ONYX* sind nach dem *Question & Test Interoperability* (QTI) Standard [51] formatiert. Dieses Format bietet, neben generellem Feedback bei richtig bzw. falsch gelöster Aufgabe, ebenfalls die Möglichkeit, individuelles Feedback je nach Antworten, erreichten Punkten und Wiederholungsversuchen zu geben. Zur Auswahl stehen 18 verschiedene Aufgabentypen, von denen 15 automatisiert bewertet werden können. Davon ausgeschlossen sind *Freitext*- und *Uploadaufgaben* sowie der Aufgabentyp *Zeichnen*.

In dieser Arbeit wird sich auf die Generierung der folgenden Aufgabentypen fokussiert:

- Einfache Auswahlaufgabe (*Single Choice*): Bei der einfachen Auswahlaufgabe sind mehrere Antwortmöglichkeiten gegeben, aus denen Lernende eine richtige Antwort selektieren müssen. Es kann allgemeines, aber auch individuelles Feedback für jede Antwortmöglichkeit gegeben werden.
- Mehrfache Auswahlaufgabe (*Multiple Choice*): Die mehrfache Auswahlaufgabe unterscheidet sich von der einfachen Auswahlaufgabe insofern, dass mehrere Antwortmöglichkeiten richtig sein können. Lernende müssen die richtigen Antworten (bzw. die richtige Antwort) entsprechend auswählen.

- Einfache Zuordnung (*Drag-and-Drop*): Die einfache Zuordnung ermöglicht die Zuordnung von Antwortpaaren zueinander. Dabei wird per Drag and Drop jeweils ein *Quellelement* zu einem *Zielelement* platziert.
- Textboxaufgabe: Die Textboxaufgabe ermöglicht eine Auswahl innerhalb eines Textes in einer Aufklappbox. Individuelles Feedback ist für jede Option aus einer Aufklappbox möglich.

Im Anhang A.2 ist zu jedem Aufgabentyp eine Beispielaufgabe zu finden.

Das QTI-Format ist ein Standard, welcher von *1EdTech* [51] entworfen wurde und auf der Extensible Markup Language (XML) basiert. Es garantiert eine konsistente Darstellung von Aufgaben und Tests und ermöglicht einen einfachen Austausch von Inhalten zwischen verschiedenen Plattformen. Neben dem bereits erwähnten möglichen Feedback bietet es Logik für eine vielseitige Bewertung und Kontext in Form von Metadaten, wie beispielsweise Variablen [51]. Eine Beispielaufgabe im QTI-Format ist im Anhang A.2 gegeben.

2.4 Large Language Model

In dieser Arbeit wird der Einsatz von Large Language Models (LLMs) zur Generierung von Übungsaufgaben und deren Evaluation untersucht, um den Prozess der Aufgabenerstellung zu unterstützen. LLMs sind neuronale Netze, welche mit großen Datenmengen trainiert wurden. Sie sind in der Lage, natürliche Sprache zu verstehen und zu generieren (*Natural Language Processing*, NLP). Mit dem Paper von Vaswani et al. [61], mit dem 2017 die Transformer-Architektur vorgestellt wurde, nahm die Entwicklung von Sprachmodellen stark zu [10, 42]. Eine Übersicht zu LLM bezogenen Konzepten bietet Naveed et al. [42]. Einem Sprachmodell wird ein sogenannter *Prompt* als Eingabesequenz gegeben. Dieser wird in einzelne *Token* zerlegt, welche Zeichen, Teilwörter, Symbole oder Wörter sein können [42]. Darauf basierend werden die Token der Antwort vorhergesagt.

Die Qualität der Antworten ist vom Training und Prompting abhängig. Spezialisierte Modelle werden für eine konkrete Anwendung, zum Beispiel dem Lösen von Mathematikaufgaben, trainiert. Dazu gehört auch der *Fine-Tuning* Ansatz, bei dem die Parameter eines vortrainierten Sprachmodells für eine Anwendung weiter trainiert werden [10]. Dabei existiert die Gefahr des *Overfitting*, bei dem ein Model stark auf eine Datenmenge spezialisiert ist, sodass es auf neue Eingaben nicht generalisiert [33].

Prompt Engineering beschreibt die Herangehensweise, wie Prompts aufgebaut und optimiert werden können [49]. Besonders *Few-Shot* und *Zero-Shot* Prompting liefern gute Ergebnisse, wie Brown et al. in ihrem Paper [10] dargestellt haben. Der Prompt wird bei diesen Techniken mit wenigen Beispielen (Few-Shot) oder nur Anweisungen zum Lösen (Zero-Shot)

ergänzt [31]. Dies ist in Abbildung 2 beispielhaft dargestellt. In dieser ist zu erkennen, dass nach einer Aufgabe gefragt und eine Beispielaufgabe mit Lösung gegeben ist, sowie eine zweite Aufgabe angefragt wird. Mit diesem *1-Shot* Prompt wird dem Sprachmodell das intendierte Ausgabeformat gezeigt, in welchem das LLM die Aufgabe generiert.

Prompt	
Erstelle eine Aufgabe zu der Fibonacci-Folge:	
# Aufgabe	
Nennen Sie die ersten drei Zahlen der Fibonacci-Folge!	
# Lösung	
1, 1, 2	
Erstelle eine Aufgabe zu dem Summationszeichen:	
Output	
# Aufgabe	
Berechnen Sie die Summe der ersten fünf natürlichen Zahlen mithilfe des Summationszeichens!	
	$\sum_{i=1}^5 i$
# Lösung	
	$\sum_{i=1}^5 i = 1 + 2 + 3 + 4 + 5 = 15$

Abb. 2: Beispiel eines Few-Shot Prompting, eigene Darstellung

Ein weiterer Ansatz ist das sogenannte *Chain-of-Thought* (CoT) Prompting, bei dem das Ergebnis durch das Berechnen von Zwischenschritten verbessert werden soll [63]. CoT Prompting funktioniert sowohl mit Few-Shot als auch mit Zero-Shot Prompting, wie Kojima et al. [31] gezeigt haben. Dafür wird das LLM aufgefordert, Schritt für Schritt vorzugehen.

2.5 Stand der Forschung

2023 forschten Meissner, Pögelt et al. [39] bereits im Rahmen des tech4compKI-Projekts an der Erstellung mathematischer Aufgaben durch ein LLM. Dabei war das Ziel zu untersuchen, inwiefern sich die Technologie für die Erstellung eignet und inwieweit sie Lehrende bei diesem Prozess unterstützen kann. Die erstellten Aufgaben wurden in einer Studie von drei Mathematik-Experten bewertet und nach der Taxonomie von Anderson und Krathwohl [3]

klassifiziert. Dabei zeigte sich, dass LLMs Aufgaben einhergehend mit den intendierten Eingaben – mathematisches Konzept, Lernziel und Dimension – bereitstellen können. Besonders aufgefallen ist, dass die erstellten Aufgaben in den Bereichen *Anwenden* und *Konzeptionelles Wissen* die höchste Präzision erreichten [39].

Afzaal, Nouri und Aayesha [1] untersuchten, wie automatisiert konzeptbasierte Übungsaufgaben durch Verarbeitung natürlicher Sprache (NLP) generiert werden können. Dafür wurde ein LLM weitertrainiert, aus einem Paragraphen Aufgaben und deren Antworten zu erstellen. Anschließend wurden mit einem zweiten Modell und demselben Paragraphen Distraktoren erstellt. Die Evaluation der Lehrenden ergab, dass die Aufgaben auf einem zufriedenstellenden Niveau generiert wurden, aber erst von einem Menschen überprüft werden sollten, bevor diese an Studierende ausgeteilt werden [1].

Elkins et al. [20] haben ebenfalls die Aufgabengenerierung zu Textpassagen durch LLMs untersucht. Die Aufgaben wurden nach zwei verschiedenen Strategien erstellt: einem einfacheren Ansatz und einem nach *Bloom's Taxonomie* gerichteten pädagogischen Ansatz. Das Ergebnis zeigte, dass der pädagogische Ansatz von Lehrenden deutlich bevorzugt wurde, aber beide Strategien Aufgaben von vergleichbarer Qualität erzeugt haben [20].

Aus der Übersicht von Kurdi et al. [32] zum automatisierten Erstellen von Aufgaben in den Jahren 2015 bis 2019 geht hervor, dass die Überprüfung der Qualität der Aufgaben von Menschen übernommen wurde. Für maschinelle Bewertungen der Qualität eignet sich neben der Nutzung von LLMs ein regelbasierter Ansatz. Moore et al. [40] untersuchten 2023 das Bewerten von Multiple-Choice-Aufgaben anhand von *Item Writing Flaws*. Die Untersuchung zeigte, dass der verwendete regelbasierte Ansatz näher an der menschlichen Wahrnehmung lag als GPT-4. Es wurde ebenfalls dargestellt, dass beide maschinellen Bewertungen strikter im Vergleich zu den Fachexperten waren.

3. ANFORDERUNGEN

In diesem Kapitel werden die Anforderungen an die Erstellung der Übungsaufgaben beschrieben, welche bei der weiteren Konzeption von Bedeutung sind.

3.1 Aufgaben

Grundlegend sind in dieser Arbeit mathematische Übungsaufgaben unter Einsatz von LLMs zu erstellen. Da die Aufgaben im Rahmen des teh4compKI-Projekts verwendet werden, sind diese ausschließlich in deutscher Sprache zu formulieren. Zu einer Übungsaufgabe gehört in diesem Sinne eine Fragestellung, Antwortmöglichkeiten und Feedback.

Dabei ist von der Fragestellung gefordert, dass diese jegliche zur Bearbeitung der Aufgabe notwendigen Informationen klar und präzise formuliert beinhaltet. Dazu zählt auch, dass Fachtermini richtig verwendet und mathematische Ausdrücke in eine Mathematikumgebung gesetzt werden. Es wird einerseits erwartet, dass alle Antwortmöglichkeiten angemessen, klar voneinander abgrenzbar und eindeutig sind, und andererseits, dass Feedback eine wirkliche Unterstützung darstellt. Darunter ist in diesem Fall zu verstehen, dass das Feedback motivierend ist und optimalerweise die Studierenden zur richtigen Lösung leitet. Die Aufgaben haben außerdem dem angegebenen mathematischen Konzept und Aufgabentyp zu entsprechen. Für die Bearbeitung der Aufgaben über OPAL ist gefordert, dass der Aufgabentyp einem der 15 von ONYX automatisiert überprüfbaren entspricht – jedoch nicht, dass alle 15 Aufgabentypen von der entwickelten Anwendung unterstützt werden müssen.

Beim Erstellen der Aufgaben ist es nicht von Relevanz, wie schnell die Aufgaben erstellt werden, da diese nicht direkt an die Studierenden, sondern in eine Aufgaben-Datenbank eingefügt und bei Bedarf durch den *Recommender* verteilt werden.

3.2 Qualität und Klassifizierung

Für die Verwendung in der Lehre sind nur qualitativ hochwertige Übungsaufgaben zu erstellen. Dazu zählen Aspekte wie eine fehlerfreie Ausführbarkeit sowie, dass Aufgabenstellung, Auswertung, Feedback und mathematischer Kontext korrekt sind.

Zudem sollen die Aufgaben den intendierten kognitiven Prozess- und Wissensdimensionen entsprechen, damit die angestrebten Lernziele korrekt abgedeckt werden. In dieser Arbeit liegt der Fokus darauf, Aufgaben in den Dimensionen *Erinnern* und *Faktenwissen* zu erstellen. Das bedeutet, dass die Aufgaben die Kompetenz erfordern, den Fachterminus und die Konzepte zu kennen und Wissen über Merkmale und Eigenschaften dieser Konzepte zu besitzen (*Faktenwissen*). Außerdem müssen diese wiedererkannt und abgerufen werden können

(*Erinnern*) [3]. Das Ziel dabei ist, Aufgaben besonders zu den Definitionen der mathematischen Konzepte zu generieren. Aufgaben, die solche Lernziele ansprechen, können ebenfalls in den nächstgelegenen Dimensionen klassifiziert sein. Daher sind leichte Abweichungen der intendierten Dimensionen akzeptabel.

Die Qualität und Klassifizierung der Aufgaben sind maschinell zu bewerten. Bei einer geringen Qualität der generierten Übungsaufgabe und/oder bei einer starken Abweichung der bestimmten von den intendierten Dimensionen ist dies zu melden.

3.3 Format

Eine technische Anforderung besteht darin, dass die generierten Übungsaufgaben im QTI-Format vorliegen, damit diese ohne weitere Anpassungen von den Studierenden über LMS wie OPAL bearbeitet werden können.

An die Darstellung der Aufgaben werden zudem die Anforderungen gestellt, dass der Text in einer angemessenen Schriftgröße wiedergegeben wird und mathematische Ausdrücke und Symbole in eine Mathematikumgebung nach dem $\text{\LaTeX}$ -Standard angezeigt werden. Dabei sollen alle von der entwickelten Anwendung unterstützten Aufgabentypen vollständig dargestellt werden und Feedback möglichst individuell gestaltet sein.

Generell ist abzuwägen, ob die Aufgaben vom LLM direkt im QTI-Format erstellt oder erst anschließend umgewandelt werden.

4. KONZEPT

Das folgende Kapitel befasst sich mit dem Konzept sowie der Methodik dieser Arbeit. Da die Übungsaufgaben durch ein LLM zu generieren und evaluieren sind, ist die Wahl des Modells von Relevanz. Um eine qualifizierte Entscheidung treffen zu können, werden zunächst verschiedene Modelle betrachtet. Folgend wird diskutiert, in welchem Format die Aufgaben generiert werden sollen. Abschließend erfolgt eine Beleuchtung der einzelnen Elemente der Methodik.

4.1 Large Language Model

Die Übungsaufgaben und maschinelle Evaluation sind durch ein LLM zu erstellen. Von dem gewählten LLM wird gefordert, dass es die Eingaben verarbeitet und vollständige Ausgaben generiert. Zudem ist es notwendig, dass die Ausgaben korrekt formatiert sind, damit diese weiterverarbeitet werden können. Zusätzlich ist gefordert, dass die erstellten Aufgaben den in Kapitel 3 formulierten Anforderungen entsprechen.

Dafür gilt es zu erörtern, welches Modell sich am besten für die Erfüllung der Anforderungen eignet. Genau betrachtet werden in dieser Arbeit zwei universell einsetzbare Modelle von *OpenAI* [45] sowie spezialisierte Modelle von *EleutherAI* [7] und *InternLM* [65].

4.1.1 Generative Pre-Trained Transformer

Zu den populärsten LLMs zählen die *Generative Pre-Trained Transformer* (GPT) von *OpenAI* [44]. Diese sind für einen allgemeinen Gebrauch bestimmt und können daher eine breite Vielfalt an Aufgaben, auch solche, für die sie nicht trainiert wurden, lösen [42, 50]. Die *Flagship*-Modelle sind *GPT-4o*, *GPT-4 Turbo* und *GPT-3.5 Turbo* (Stand Juni 2024). Dabei ist GPT-4o das fortschrittlichste [45] und erzielt bessere Leistungen in nicht englischen Sprachen und bei Benchmarks, wie beispielsweise *MMLU* [25] und *MATH* [26] [24].

Die Modelle werden über eine kostenpflichtige API zur Verfügung gestellt, wodurch die lokalen Hardware-Anforderungen gering gehalten werden können. Dabei werden die Kosten pro Token vom Input und Output berechnet.

Die *Temperatur* ist ein Parameter, der bei der Verwendung von GPT zu betrachten ist. Diese regelt die Kreativität bzw. Zufälligkeit bei der Generierung der Antwort und kann zwischen 0.0 und 2.0 liegen [45]. Praktisch betrifft dies die Gewichtung der nächsten möglichen Token [49]. Eine höhere Temperatur erhöht jedoch die Gefahr der sogenannten *Halluzinationen*. Bei Halluzinationen handelt es sich beispielsweise um falsche oder nicht passende Antworten eines LLMs, welche jedoch plausibel klingen [42]. Dazu zählt, dass die Antwort vom *Input*

abweicht, der *Kontext* sich selbst widerspricht oder dass *Fakten*, welche nicht der Wahrheit entsprechen, generiert werden [42, 66].

Agarwal et al. [2] vergleicht die Diversität von generierten Aufgaben mit verschiedenen Temperaturen. Dabei wurde festgestellt, dass im Vergleich zu einer niedrigen Temperatur (0.2), höhere Temperaturen (1.0 - 1.2) mehr Diversität erzeugen. Ebenfalls wird diskutiert, dass vor allem in den unteren Dimensionen von Bloom's Taxonomie [9] die Generierung von vielfältigen Aufgaben ein Problem für GPT-4 darzustellen scheint. Jedoch wird in dem Paper das Problem der Halluzination nicht betrachtet.

Während die Diversität der Aufgaben nicht uninteressant ist, wird auf diese und deren Überprüfung im Rahmen dieser Arbeit nicht weiter eingegangen. Grund dafür ist, dass der Fokus auf der grundsätzlichen Generierung liegt und darauf, inwiefern eine maschinelle Evaluation der menschlichen Wahrnehmung entspricht. Erst wenn Aufgaben ohne menschliche Intervention in den Aufgabenpool übernommen werden, wäre die ausgiebige Betrachtung der Diversität erforderlich. In dem Zusammenhang ist bei der Verwendung von GPT die Temperatur auf 0.0 gestellt, damit sich die Aufgaben möglichst nah an der Formulierung, beispielsweise von Definitionen, aus dem gegebenen Wissen orientieren.

4.1.2 Spezialisierte LLMs

Universell einsetzbare Sprachmodelle, wie die GPT-Modelle, werden auf einer Vielfalt von Texten trainiert, um möglichst passende Ergebnisse bei der gesamten Bandbreite an natürlichen Texten zu erzielen [7, 42]. Verschiedene Anwendungsbereiche stellen jedoch unterschiedliche Herausforderungen dar. Im Bereich der Mathematik steht das Lösen von mathematischen Problemen und das Darlegen der Schlussfolgerungen im Vordergrund [42]. Während Defizite im arithmetischem Rechnen von LLMs durch *Prompt Engineering* reduziert werden können [28], sind spezialisierte LLMs in der Lage, bessere Leistungen für geringere Rechenkosten zu erzeugen [7]. Daher werden im Folgenden verschiedene spezialisierte LLMs betrachtet.

Llemma Eins der betrachteten spezialisierten LLMs ist *Llemma*, ein Open-Source-Sprachmodell von *EleutherAI* [7]. Dieses wurde 2024 veröffentlicht und basiert auf *Code Llama* [54], ein Open-Source-Modell, speziell trainiert für die Programmierung. Dadurch besteht eine Basis aus allgemeinem Text- und Code-Verständnis, sowie Berücksichtigung des gesamten Umgebungskontexts und der Verarbeitung von längeren Kontexten [54]. Dies ist für die Generierung von Aufgaben und Evaluationen von Vorteil, da Eingaben durch Few-Shot Prompting und *Knowledge Injection*, dem Einfügen von notwendigen Informationen für das Generieren der Antwort, welche das LLM wahrscheinlich nicht besitzt [37], län-

ger werden. Llemma wurde mit dem Open-Source-Datenset *proof-pile-2* trainiert, welches hauptsächlich aus *algebraic-stack*, *OpenWebMath* [46] und *ArXiv* [14] zusammengesetzt ist [7]. Das Fine-Tuning spezialisiert das LLM für das Lösen mathematischer Probleme. Dies sollte beispielsweise dafür sorgen, dass die korrekte mathematische Formatierung eingehalten wird. Llemma wird in einem 7B- und einem 34B-Modell bereitgestellt [7], wobei sich die Zahl auf die Parameteranzahl des LLM bezieht. Eine kleinere Anzahl bedeutet, dass das Modell schneller trainiert ist und weniger Speicherbedarf hat, jedoch bei einer großen Parameteranzahl Potenzial für bessere Ergebnisse besteht [30].

InternLM-Math Ein zweites in dieser Arbeit betrachtetes spezialisiertes LLM ist *InternLM-Math* von *InternLM* [65]. Dieses ist ebenfalls quelloffen und wurde 2024 veröffentlicht, basiert aber auf *InternLM2-Base* [11], einem Sprachmodell für den allgemeinen Gebrauch. InternLM2-Base kann größere Kontexte verarbeiten, wobei Inferenzkosten durch die Nutzung von *Group Query Attention* reduziert werden [11]. Trainiert wurde InternLM-Math hauptsächlich mit den Datensets *Knowledge Pile* [21], *algebraic-stack* [7] und *OpenWebMath* [46]. Diese decken sich teils mit denen von Llemma, weshalb InternLM-Math ebenfalls mathematische Formatierung berücksichtigen sollte. Insgesamt ist InternLM-Math performanter bei Benchmarks wie *MATH* [26] und *GSM8K* [13] als Llemma [65]. InternLM-Math wird in einer 7B- und einer 20B-Variante bereitgestellt [65].

4.1.3 Vergleich der LLMs

Für die Entscheidung, welches LLM in dieser Arbeit verwendet wird, wurden die vorgestellten Modelle mit jeweils denselben Prompts getestet und verglichen. Die Tests wurden auf dem Server des tech4compKI-Projekts mit einem Intel Core i9-7920X Prozessor und vier NVIDIA GeForce RTX 2080 TIs ausgeführt. Die Prompts und jeweiligen Ausgaben sind im Anhang A.1 beigefügt. Bei der Entscheidung wurde neben der Qualität besonders auf das Format der Antwort geachtet, da dies für die Weiterverarbeitung bedeutsam ist. Die Ergebnisse der Tests sind in Tabelle 1 dargestellt. Die Zeit zur Generierung ist dort ebenfalls notiert. Sie dient vor allem zur allgemeinen Information, da diese, wie in Kapitel 3 festgehalten, nicht von Relevanz ist.

Tabelle 1 zeigt, dass die Modelle von *OpenAI* Ausgaben meistens in dem gewünschten Format und in hochwertiger Qualität generierten, während die Modelle von *InternLM* mangelhafte Ausgaben erstellten. Besonders *InternLM-Math 20B* hatte bei jedem Test Speicherprobleme, welche ebenfalls bei *Llemma 34B* auftraten. Mit dem gekürzten Prompt und bei der Evaluation konnte *Llemma 34B* Ausgaben erzeugen, welche aber nicht von hoher Qualität waren und das Format nur teils dem intendierten entsprach. *Llemma 7B* hat von

	Zeit (Ø)	Aufgaben Generierung		Aufgaben Generierung *		Evaluation	
		Format	Qualität	Format	Qualität	Format	Qualität
GPT-3.5 Turbo	6s	++	++	++	+	–	++
GPT-4o	7s	++	++	+	++	++	++
Llemma 7B	1min	– +	+	– +	– +	– +	+
Llemma 34B	30min	X	X	– +	–	– +	+
InternLM-Math 7B	1min	– –	–	– –	–	– –	+
InternLM-Math 20B	X	X	X	X	X	X	X

Tabelle 1: Testergebnisse der betrachteten LLMs. Die Bewertung des Formats und der Qualität reicht dabei von – – (sehr schlecht) bis ++ (sehr gut), wobei X einen fehlenden Wert darstellt. * kennzeichnet, dass der Prompt weiter gekürzt wurde.

den spezialisierten Modellen am besten abgeschnitten; das Format wurde allerdings nur zum Teil eingehalten und die Qualität war ausbaufähig.

Insgesamt erzielte GPT-4o die besten Resultate, wobei das Format stets eingehalten wurde und die Qualität ebenfalls zufriedenstellend war. Daher ist dieses Modell von den betrachteten am besten für diese Arbeit geeignet und wird folglich für die Implementierung verwendet.

Für die Tests wurde mit *LangChain* [34] gearbeitet, welches es ermöglicht, Modell leicht auszutauschen. Dieses Framework wird auch in der weiteren Implementierung verwendet, damit zukünftige LLMs leicht in die Anwendung eingebunden werden können.

4.2 Question & Test Interoperability Standard

Wie in den Anforderungen in Kapitel 3 festgehalten, ist es nicht relevant, ob Aufgaben direkt im QTI-Format erstellt oder erst später umgewandelt werden. Durch Few-Shot Prompting ist es möglich, durch das Hinzunehmen von Beispielaufgaben im QTI-Format, ebenfalls eine QTI-formatierte Aufgabe zu generieren. Dies wurde mit einigen Testprompts ausprobiert, bei denen mit einem kurzen Skriptausschnitt und zwei Beispielen korrekt formatierte QTI-Aufgaben erstellt wurden.

Bei diesem Ansatz ist problematisch, dass die Anzahl der *Prompt* und *Completion Token* sehr hoch ist, da eine Beispielaufgabe circa 1500 bis 2000 Token umfasst. Bei limitierten Token kann das dazu führen, dass die Aufgabe ggf. nicht vollständig generiert wird. Zudem bedeuten mehr Prompt-Token auch höhere Inferenz und somit Kosten [36]. Es wurden keine konkreten Anforderungen bzgl. der Kosten gestellt. Allgemein gilt jedoch das Bestreben, diese gering zu halten, ohne dabei die Qualität deutlich zu vermindern.

Daher wurde der Ansatz gewählt, Aufgaben in einem anderen Format erstellen zu lassen und anschließend in das QTI-Format umzuwandeln.

Für die Konvertierung gibt es verschiedene, bereits existierende Projekte. Zu nennen ist *text2qti* [48]. Dies ermöglicht, QTI-Aufgaben aus Markdown-Dateien zu erstellen. Codeausschnitt 1 zeigt beispielhaft das Format einer *Single-Choice*-Aufgabe, welche von der Anwendung umgewandelt werden kann. Gezeigt wird ebenfalls, dass das Feedback für die einzelnen Antwortmöglichkeiten unterstützt wird. Das ermöglicht individuelles hilfreiches Feedback zu geben, wie es in Kapitel 3 gefordert wird.

```
1 Quiz title: Addition
2 Quiz description: Checking addition.
3
4 Title: An addition question
5 Points: 2
6 1. What is 2+3?
7 ... General question feedback.
8 + Feedback for correct answer.
9 - Feedback for incorrect answer.
10 a) 6
11 ... Feedback for this particular answer.
12 b) 1
13 ... Feedback for this particular answer.
14 *c) 5
15 ... Feedback for this particular answer.
```

Codeausschnitt 1: Format einer Single-Choice-Aufgabe die von text2qti umgewandelt werden kann [48]

Das Package *text2qti* ist für diese Arbeit nicht verwendbar, da es QTI-Aufgaben der Version *v1.2* erstellt, welche inkompatibel mit der bei ONYX verwendeten Version *v2.1* ist.

CSV2QTI [55] wurde entwickelt, um Aufgaben für die *TAO Testplattform* aus CSV-Dateien umzuwandeln. Mit diesem Programm können *Single-Choice*-Aufgaben erstellt werden. Die Comma-Separated Values (CSV)-Datei muss dafür den Aufgabentext, den Index der richtigen Antwort und die Antwortmöglichkeiten beinhalten. Für diese Arbeit kommt *CSV2QTI* nicht in Frage, da kein Feedback gegeben werden kann, dies aber in den Anforderungen in Kapitel 3 verlangt wird.

Da kein für diese Arbeit geeignetes Projekt gefunden wurde, wird ein eigener Konvertierer implementiert. Dieser wandelt Aufgaben von *JavaScript Object Notation* (JSON) in QTI-Aufgaben um. Der Aufbau ist ähnlich zu dem von *text2qti*. JSON wurde gewählt, da dies

ein weit verbreitetes Text-basiertes Datenaustauschformat ist, welches leicht von Menschen gelesen werden kann [17, 29].

4.3 Methodik

In dieser Arbeit werden in zwei Phasen Übungsaufgaben iterativ erstellt und evaluiert. Abbildung 3 zeigt das Vorgehen, bei der zunächst mittels Few-Shot Prompting Aufgaben generiert sowie anschließend maschinell und von einer Expertin bewertet werden. Die Bewertung der Expertin wird verwendet, um die Generierung der Aufgaben zu verbessern sowie die maschinelle Evaluation mit Aufgaben-Evaluationspaaren (Few-Shot Prompting) zu bereichern. In der zweiten Phase werden anschließend weitere Aufgaben erstellt und erneut bewertet.

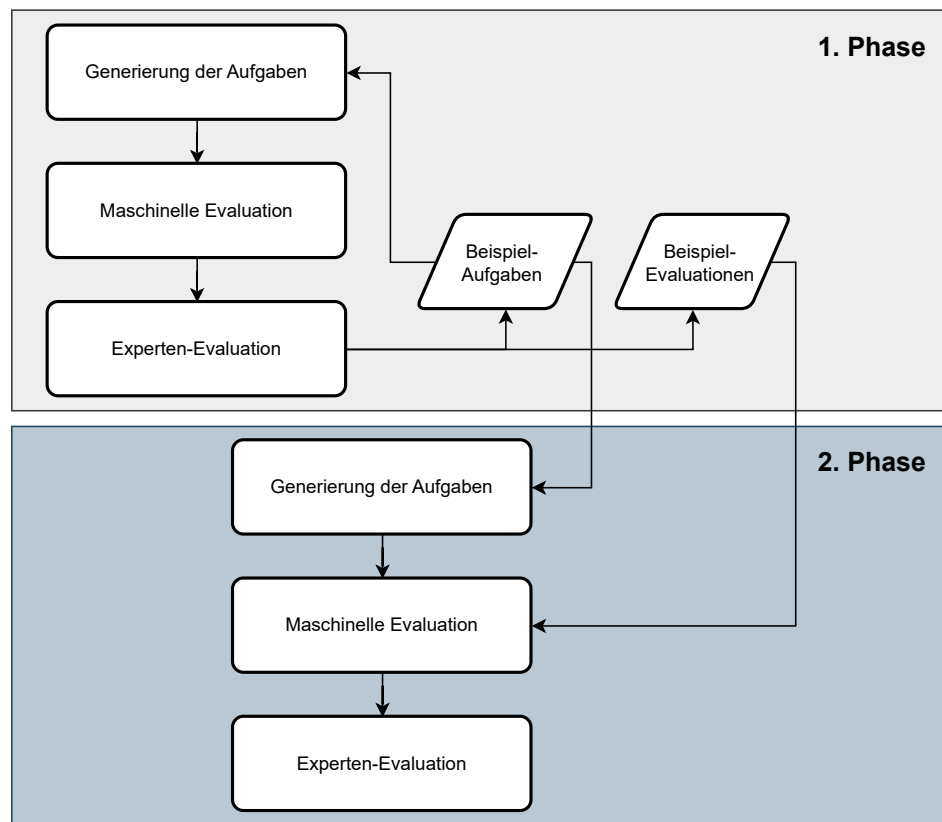


Abb. 3: Herangehensweise der Aufgabengenerierung und Evaluierung von dieser Arbeit, eigene Darstellung

4.3.1 Maschinelle Evaluation der Qualität

Die Qualität der Aufgaben ist automatisiert zu überprüfen, um Aufgaben von niedriger Qualität zu erkennen und ggf. herauszufiltern. Das ist von Bedeutung, damit die Arbeit für die Lehrenden weiter reduziert bzw. die Lehre der Studierenden nicht beeinträchtigt wird. Dabei wird NLP-basiert vorgegangen, das heißt, dass einzelne Merkmale einer Aufgabe, z.B. Lösbarkeit, durch Anfragen an ein Sprachmodell überprüft werden. Der Ansatz orientiert sich an der Vorgehensweise von Moore et al. [40].

Das LLM wird aufgefordert, neben Angaben zum Format der Ausgabe, die Entscheidung über das Merkmal zu begründen (Zero-Shot-CoT). In der zweiten Phase wird der Prompt um Beispiellevaluationen ergänzt (Few-Shot-CoT).

Damit die Einheitlichkeit der Evaluation gewährleistet werden kann, wird pro Qualitätsmerkmal nur eine Anfrage an das LLM gestellt. Aufgrund dessen unterscheidet sich die Semantik der kontextuelle Korrektheit. Maschinell wird überprüft, ob das intendierte mathematische Konzept im Fokus der Aufgabe steht, während die menschliche Evaluation bereits die Korrektheit der mathematischen Sprache in dem Kontext beurteilt.

4.3.2 Maschinelle Evaluation der Taxonomie

Es ist relevant, dass die generierten Aufgaben nach Anderson und Krathwohl klassifiziert werden. Dies ist ebenfalls notwendig, damit der *Recommender* durch die klassifizierte Kompetenz, welche bei der Aufgabe angesprochen wird, das Lernen unterstützen kann. Daher wird wie die Qualität auch die Klassifizierung der Taxonomie automatisiert überprüft. Hier wird ebenfalls NLP-basiert vorgegangen. Dem LLM wird eine Beschreibung der kognitiven Prozess- bzw. Wissensdimensionen gegeben und die Aufforderung, in diese zu klassifizieren sowie die Entscheidung der Wahl zu begründen (Zero-Shot-CoT). Ebenfalls wird in der zweiten Phase mit Beispielklassifikationen gearbeitet (Few-Shot-CoT).

Entsprechend den festgehaltenen Anforderungen sind die intendierten Dimensionen *Erinnern* und *Faktenwissen*. Die nächst komplexeren Dimensionen (*Verstehen* und *Konzeptuelles Wissen*) werden ebenfalls akzeptiert, da sich die Dimensionen überschneiden.

4.3.3 Menschliche Evaluation der Aufgaben

Für die Darstellung der menschlichen Wahrnehmung wird eine Fachexpertin die Qualität und Klassifizierung der erstellten Übungsaufgaben evaluieren. Sie ist unter anderem wissenschaftliche Mitarbeiterin im tech4compKI-Verbundprojekt und hat viele Jahre Lehrerfahrung im Bereich der Mathematik, ist geschult in der Kompetenzeinschätzung und routiniert in der Erstellung mathematischer Aufgaben.

Konzept	AT	KP	W	AF	AS	AW	FZ	KK	Bewertung
Addition	SC	1	1	ja	ja	ja	ja	ja	ja
Anstiegsdreieck	TB	1	1	ja	ja	ja	ja	ja	ja
Anstiegsdreieck	DD	1	1	ja	ja	ja	ja	ja	ja
Approximation	TB	1	1	ja	–	ja	ja	–	–
Approximation	DD	1	1	ja	ja	ja	ja	ja	ja
Approximation	MC	1	1	ja	–	ja	ja	–	–
Aussagenverknüpfung	DD	1	1	ja	ja	ja	ja	ja	ja
Aussagenverknüpfung	MC	1	1	ja	ja	ja	ja	ja	ja
Bereichsintegral	SC	1	1	ja	ja	ja	ja	–	–
Bereichsintegral	TB	1	1	ja	ja	ja	ja	–	–

Tabelle 2: Ausschnitt aus der Ergebnistabelle der Expertin

Bedeutung der Abkürzungen: AT – Aufgabentyp, KP – Kognitive Prozessdimension, W – Wissensdimension, AF – Ausführbarkeit, AS – Aufgabenstellung, AW – Auswertung, FZ – Feedbackzweige, KK – Kontextuelle Korrektheit

Die Expertin wurde gebeten, die über ONYX bereitgestellten generierten Aufgaben nach denselben Merkmalen zu beurteilen, welche auch automatisiert bewertet wurden. Die erste Phase besteht aus der Evaluation von 64 Aufgaben (für 16 ausgesuchte Konzepte jeweils eine für jeden Aufgabentyp). In der zweiten Phase wird eine Teilmenge von 80 zufällig gewählten Aufgaben (20 von jedem Aufgabentyp) von den 300 generierten Aufgaben von der Expertin bewertet.

Die Ergebnisse der Evaluation wurden, wie die der maschinellen Evaluation, in einer Tabelle festgehalten. Ein Ausschnitt ist in Tabelle 2 dargestellt. Diese beinhaltet die binäre Klassifikation, ob das Merkmal akzeptabel ist oder nicht. Grundsätzlich wurde eine Übungsaufgabe durch das mathematische Konzept und dem Aufgabentyp identifiziert. Beispielaufgaben sowie deren menschlichen und maschinellen Evaluation sind im Anhang A.2 gegeben.

5. IMPLEMENTIERUNG

Dieses Kapitel beschäftigt sich mit der Umsetzung der Anforderungen und Konzepte. Zunächst wird die Struktur beleuchtet, bevor auf die Implementierung einzelner Komponenten eingegangen wird. Der gesamte Code ist im zugehörigen GitLab-Projekt [53] einzusehen.

5.1 Struktur

Auf Basis der in dem Kapitel 3 formulierten Anforderungen und der in Kapitel 4 festgehaltenen Konzepte wurde eine Anwendung entwickelt, welche den in Abbildung 4 dargestellten Prozess realisiert. Diese zeigt den Ablauf von oben nach unten, bei dem die Generierung einer Aufgabe nach Erhalt von Eingabeparametern gestartet wird. Diese Aufgabe liegt zunächst in JSON vor und wird an die maschinelle Evaluation weitergegeben. Unabhängig von dem Ergebnis der Evaluation wird die Aufgabe an den JSON-QTI-Konvertierer weitergereicht. Anschließend ist die erstellte Aufgabe fertig und kann bei ONYX hochgeladen werden.

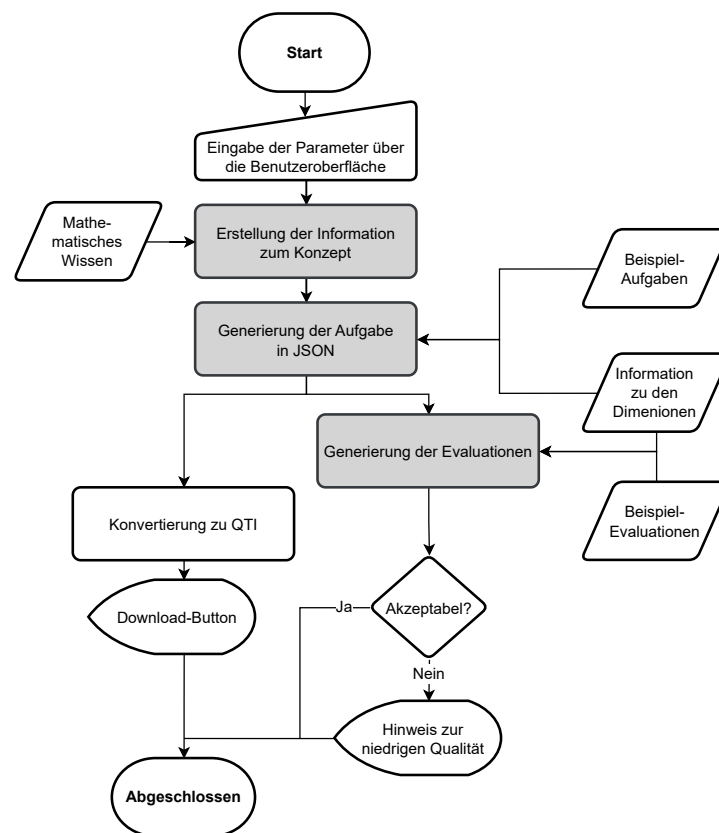


Abb. 4: Ablauf der Aufgabengenerierung und maschinellen Evaluation, eigene Darstellung. Bei dunkel hinterlegten Prozessen wird ein LLM verwendet.

Grundlegend ist die Anwendung in Python programmiert, da diese Programmiersprache sich besonders durch die Verfügbarkeit von Tools und Libraries als präferierte Programmiersprache für Data Science und maschinelles Lernen etabliert hat [52]. Wie in Kapitel 4 festgehalten wird die Generierung von Aufgaben und Evaluationen mit Hilfe von GPT-4o umgesetzt. Dafür wird eine Instanz von `ChatOpenAI` mit einem API-Key, Modellnamen und Temperatur erstellt.

5.2 Benutzeroberfläche

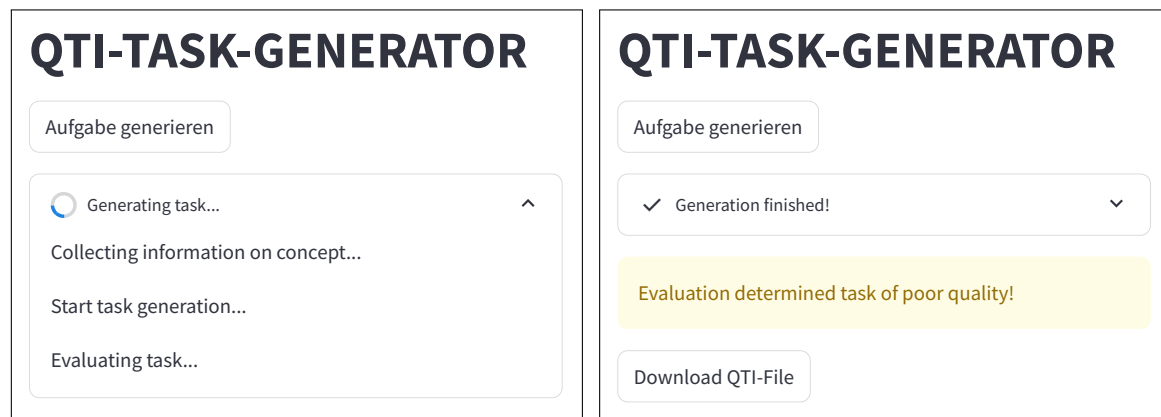
Die Anwendung ist über eine grafische Benutzeroberfläche zu verwenden, welche sich an Meissner und Pögelt [38] orientiert. Diese ermöglicht, eine Übungsaufgabe zu erstellen und bei Bedarf im QTI-Format herunterzuladen. Für die Implementierung der interaktiven Webanwendung wurde das Framework *Streamlit* [56] verwendet. Das Open-Source-Framework bietet die Möglichkeit, Daten über eine API zu visualisieren, zu mutieren und zu teilen [4].

The screenshot displays the 'QTI-TASK-GENERATOR' web application. On the left, a sidebar contains several configuration options: 'Welches mathematische Konzept soll erzeugt werden?' with a dropdown set to '\$p\$-Norm [Norm]'; 'Welcher Aufgabentyp soll erzeugt werden?' with a dropdown set to 'Single-Choice'; 'Einstellungen' section including 'Welches Modell soll verwendet werden?' (GPT-4o), 'Bitte geben Sie den OpenAI API-Key ein:' (with a text input showing 'Beispiel: sk-pr...' and a toggle icon), 'Welche Chain soll verwendet werden?' (map_reduce), 'Welcher Algorithmus soll verwendet werden?' (mmr), and 'Anzahl Dokumente' (a slider set to 4 between 1 and 10). On the right, the title 'QTI-TASK-GENERATOR' is prominently displayed above a button labeled 'Aufgabe generieren'.

Abb. 5: Benutzeroberfläche der Anwendung, eigener Screenshot

Durch den Aufruf von `streamlit run ./src/generator.py` wird die Anwendung gestartet und die Webanwendung unter dem Port `8501` bereitgestellt. Abbildung 5 zeigt die Benutzeroberfläche, bei der auf der linken Seite eine *Sidebar* mit Konfigurationsoptionen sichtbar ist. In dieser kann die Auswahl der Parameter zur Erstellung einer Übungsaufgabe eingegeben werden. Dazu zählt die Wahl des gewünschten mathematischen Konzepts und des Aufgabentyps. Hinzu kommen unter der Überschrift „Einstellungen“ weitere Einstellungsmöglichkeiten für die Generierung. Dazu zählen das zu verwendende Sprachmodell und ggf. den notwendigen API-Key, die in Abschnitt 5.4 genannten *Chains*, Algorithmen zur Ähnlichkeitssuche sowie die Anzahl der Dokumente, die verwendet werden soll. Dabei ist zu beachten, dass das ausgewählte Modell für die Generierung der Aufgaben sowie die Evaluation verwendet wird.

Nach dem Drücken des Buttons **Aufgabe generieren** wird der in Abbildung 4 im Abschnitt 5.1 beschriebene Prozess gestartet. Die einzelnen Etappen sind über den erscheinenden *Status Container* einsehbar. Dieser ist in Abbildung 6a dargestellt.



(a) Anzeige der Generierung einer Aufgabe

(b) Warnung über niedrige Qualität

Abb. 6: Anzeige während und nach der Generierung einer Aufgabe, eigene Screenshots

In Abbildung 6b wird der Button **Download QTI-File** gezeigt. Über diesen kann die erstellte Aufgabe im QTI-Format heruntergeladen werden. Weiterhin ist zu sehen, dass im Falle von festgestellter niedriger Qualität der Aufgabe eine entsprechende Mitteilung angezeigt wird, wie es in Kapitel 3 gefordert ist.

5.3 Aufgaben in JSON

Die generierten Aufgaben liegen, bis sie im letzten Schritt ins QTI-Format konvertiert werden, in JSON vor. Dafür wurde eine Struktur für die unterstützten Aufgabentypen erstellt. In Codebeispiel 2 ist beispielhaft eine *Single-Choice*-Aufgabe dargestellt. Die gezeigte grund-

legende Struktur ist bei jedem Aufgabentyp gleich und besteht aus einem *Name/String*-Paar `question_type` und einem *Name/Object*-Paar `question`. Damit kann der Aufgabentyp direkt bestimmt und die Aufgabe aus dem `question`-Objekt extrahiert werden. Alle Aufgaben teilen sich zudem die *Name/String*-Paare `title` und `question_text`.

Für die Auswahlaufgaben liegen die Antwortmöglichkeiten in dem Array `options`. Eine Antwort besteht aus einem Text, der Markierung, ob die Antwort richtig ist, und dem Feedback. Da jede Antwortmöglichkeit ein eigenes Feedback besitzt, bietet es die Möglichkeit unterstützendes und hilfreiches Feedback individuell anzugeben. Dies ist ebenfalls in dem Codeabschnitt gezeigt.

```
1 {
2   "question_type": "single-choice",
3   "question": {
4     "title": "Allquantor",
5     "question_text": "Wählen Sie den Allquantor aus.",
6     "options": [
7       {"text": "$$\forall\text{forall}$$", "is_correct": true, "feedback":
8         ↪ "Sieht gut aus!"},
9       {"text": "$$\exists\text{exists}$$", "is_correct": false, "feedback":
10        ↪ "Das ist der falsche Quantor. Damit die Vertauschung
11        ↪ sich nicht festsetzt, schauen Sie sich noch einmal
12        ↪ die Quantoren an."},
13       {"text": "$$A$$", "is_correct": false, "feedback": "Das ist
14        ↪ ein normales A in einer Mathematik-Umgebung. Schauen
15        ↪ Sie sich die Quantoren noch einmal an."}
16     ]
17   }
18 }
```

Codeausschnitt 2: Darstellung einer Single-Choice-Aufgabe in JSON

Drag-and-Drop-Aufgaben besitzen anstelle von `options` drei weitere *Name/Value*-Paare: `pairs`, `distractors` und `feedback`. In `pairs` finden sich die zueinander gehörenden Paare als Objekte unter `left` und `right`. Da von QTI bei diesem Aufgabentyp festgelegt ist, dass *linke* Quellelemente *rechten* Zielelementen zugeteilt werden, können bei den `distractors` weitere Antwortmöglichkeiten als `left` gegeben sein. Es ist kein individuelles Feedback für Antwort-Kombinationen möglich. Daher sind beim `feedback`-Objekt Texte für eine vollständig richtige Antwort und für teilweise oder vollständig falsche Antworten notiert. Der Aufgabentyp *Textbox* besitzt die *Name/Value*-Paare `gap` und `feedback`. Wichtig ist, dass in dem Aufgabentext Platzhalter der Form `[gap1]` vorhanden sind. Die Antwortmöglichkeiten für die einzelnen Lücken sind in einem eigenen Array in `gap` enthalten. Feedback

für das vollständig richtige Lösen der Aufgabe und pro Lücke, bei falschen Antworten, werden im `feedback`-Objekt gegeben.

Zu jedem Aufgabentyp ist im Anhang A.2 ein Beispiel gegeben.

5.4 Generierung der Aufgaben

Das Erstellen der Aufgaben orientiert sich an der Implementierung von Meissner und Pögelt [38]. Der Prompt wird aus verschiedenen Teilen zusammengesetzt. Zunächst wird ein *System*-Prompt übergeben, welcher grundlegende Informationen wie Regeln zur Erstellung festlegt. Dazu zählt beispielsweise die Aufforderung, die Aufgaben ausschließlich in deutscher Sprache zu generieren, mathematische Terme nach dem $\text{\LaTeX}$ -Standard und die Ausgabe in JSON zu formatieren, wie es in Kapitel 3 und nach dem Konzept in Kapitel 4 vorgegeben ist.

Anschließend werden dem LLM Informationen zum gewünschten mathematischen Konzept und der kognitiven Prozessdimension und Wissensdimension überreicht. Die verwendeten Wissensquellen für diese Arbeit sind zwei Mathematikbücher [5, 6] und das Vorlesungsskript des Moduls *Mathematik für Informatiker I* [23]. Dies dient der Sicherstellung, dass das LLM Wissen zu diesem Konzept hat und der Text möglichst in den Studierenden bekannten Formulierungen vorliegt. Dem LLM wird dazu eine Zusammenfassung von Abschnitten aus den Wissensquellen gegeben. Diese Abschnitte werden durch die *Facebook AI Similarity Search* (FAISS) [19] bereitgestellt. FAISS ist eine *Library*, welche ermöglicht effizient *Vector Stores*, in der die Wissensquellen eingebettet vorliegen, zu durchsuchen [15, 19]. Bei der Suche nach einer gewählten Anzahl an Abschnitten, wird der ausgewählte Suchalgorithmus verwendet; zur Verfügung stehen *Maximal Marginal Relevance* (MMR) und *Similyrity Score Threshold*. Die Zusammenfassung erstellt der *Summerizer* [38] mit dem Sprachmodell und der gewählten *Chain*-Art; implementiert sind *Map Reduce*, *Stuff*, *Map Rerank*, *Refine*. Die Standardauswahl, welche bei der Generierung der Aufgaben im Rahmen dieser Arbeit verwendet wurde, liegt bei MMR und Map Reduce. MMR eignet sich, da durch die reduzierte Dopplung von Informationen eine breitere Auswahl zustande kommen kann, was zu einer größeren Kontextualisierung des Konzepts führen kann [12]. Map Reduce ist gewählt, damit die Zusammenfassung dieser Informationen zunächst getrennt geschieht und dadurch die einzelnen Teile möglichst differenziert betrachtet werden können [16].

Informationen über die Dimensionen orientieren sich an der Beschreibung von Anderson und Krathwohl [3]. Entsprechend der in Kapitel 3 formulierten Anforderungen sind Aufgaben im Bereich *Faktenwissen* und *Erinnern* zu erstellen. Da keine anderen Dimensionen bei der Generierung von Aufgaben in dieser Arbeit gewählt werden, existieren keine Eingabeparameter zur Auswahl der Dimensionen.

Abschließend werden mit einem *5-Shot* Prompting Beispiele für Aufgaben des gewählten Aufgabentyps eingelesen und dem LLM übergeben. Diese sind bereits in JSON formatiert. Das Übergeben des Prompts an das Sprachmodell und das Parsen des Outputs in JSON wird mit Hilfe der LangChain Expression Language [35] in Form einer weiteren *Chain* realisiert. Diese Chain wird dann mit den gewünschten Parametern (mathematisches Konzept, Aufgabentyp und Sprachmodell) aufgerufen. In Codeausschnitt 3 wird in Zeile 1 die Chain definiert und in Zeile 3 aufgerufen. Der `try-except`-Block dient dazu, dass der Output im Fall eines „Invalid json output“-Errors möglichst korrigiert und somit die generierte Aufgabe weiter verarbeitet werden kann. In der Regel kommt es zu dieser Meldung, wenn Backslashes für mathematische Ausdrücke nur *einfach* vorhanden sind und somit als *Escape-Sequenz* gewertet werden.

```
1 chain = prompts() | get_llm(model) | get_parser()
2     try:
3         result = chain.invoke({
4             "topic_information": topic_information,
5             "cpd_information": cpd_information,
6             "kd_information": kd_information,
7             "topic": topic,
8             "item_type": item_type,
9             "examples": examples,
10        }, config={"callbacks": [PromptLogger()]})
11    except Exception as e:
12        result = correct_backslash_in_output(e)
13    log_prompt_output_pairs(is_prompt=False, text=result)
```

Codeausschnitt 3: Aufruf zur Aufgabenerstellung mit Output-Korrektur und Logging

Der `PromptLogger` in Zeile 10 und der Aufruf in Zeile 13 dienen der Speicherung des Input-Output-Paares. Diese Paare können für das *Fine Tuning* eines Sprachmodells verwendet werden. Dafür liegen diese in JSON mit der Struktur `{"prompt": prompt, "output": result}` vor.

5.5 Maschinelle Evaluation

Die maschinelle Evaluation erstellt Bewertungen zu den Qualitätsmerkmalen, welche aus den Anforderungen in Kapitel 3 abgeleitet sind: festgestellte *kognitive Prozessdimension*, festgestellte *Wissensdimension*, korrekte *Ausführbarkeit*, Korrektheit der *Aufgabenstellung*, Korrektheit der *Auswertung*, Korrektheit der *Feedbackzweige* sowie die *kontextuelle Korrektheit*. Diese Bewertungen, abgesehen von den Klassifizierungen, entsprechen der Form

„okay“, wenn das Merkmal hochwertig ist und „–“, wenn das Merkmal mangelhaft ist. Fehler bei der Generierung der Wertung werden mit „Error“ gekennzeichnet und ebenfalls als mangelhaft gewertet.

Die Evaluation wird in der Klasse *Evaluation* realisiert. Eine Instanz dieser Klasse wird mit der Übungsaufgabe, dem intendierten mathematischen Konzept und dem zu verwendenden Sprachmodell initialisiert. Die einzelnen Merkmale werden in der Datenklasse **Attribute** gespeichert. Jede Instanz eines Merkmals setzt sich aus dem Namen des Merkmals, der Entscheidung, ob das Merkmal hochwertig ist, und der Begründung über die Entscheidung zusammen.

Mit Ausnahme der korrekten Ausführbarkeit werden die Merkmale durch ein Sprachmodell überprüft. Dafür wird der jeweilige Prompt aus den folgenden Teilen zusammengesetzt:

- Informationen über die Dimensionen (Kognitive Prozessdimension, Wissensdimension) bzw. über die mathematischen Konzepte (Kontextuelle Korrektheit)
- Die Aufforderung, die Ausgabe in JSON zu formatieren
- Der Anweisung, das Merkmal zu bestimmen und diese Entscheidung zu begründen
- Phase 1: Die Musterausgabe, um das Format der Ausgabe genauer zu beschreiben
Phase 2: Beispiele in Form von einer Aufgabe und anschließender Bewertung im gewünschten Ausgabeformat (*5-Shot* Prompting)
- Der Aufgabe, die zu bewerten ist

Bei den Merkmalen „korrekte Auswertung“ und „kontextuelle Korrektheit“ entspricht die Form der Bewertung des Sprachmodells einer Liste an richtigen Antworten, respektive einem ermittelten Konzept. Diese Bewertungen werden anschließend mit den in der Aufgabe markierten richtigen Antworten bzw. dem intendierten Konzept abgeglichen und die Wertung festgelegt.

Für die Bewertung der „korrekten Ausführbarkeit“ wird die Aufgabe hinsichtlich der einzelnen Komponenten überprüft. Fehlt ein *Name/Value*-Paar oder ist es unzureichend, wird die Aufgabe als nicht korrekt ausführbar gewertet, da die Umwandlung in das QTI-Format sowie eine korrekte Darstellung und Lösung der Aufgabe nicht mehr gegeben ist. Beispiele dafür wären, dass der Aufgabentext fehlt oder keine Lücke bei einer Textboxaufgabe vorhanden ist.

Sobald bei einer Aufgabe ein Mangel bestimmt wird, wird diese insgesamt als mangelhaft gewertet [40, 57]. Es ist zu beachten, dass Mängel ebenfalls durch falsche Formatierung der Entscheidung entstehen können. Dies liegt daran, dass ausschließlich eine positive Antwort als solche gewertet wird.

Zudem wurde in den Kapiteln 3 und 4 festgehalten, dass die nächst gelegenen Dimensionen nicht als Mangel anzumerken sind. Das Abweichen beider Dimensionen zur nächst gelegenen Stufe wird daher toleriert, sofern kein anderer Mangel festgestellt wurde.

5.6 Umwandlung von JSON zu QTI

Für die Umwandlung von JSON zu QTI wurde, gemäß den Untersuchungen in Abschnitt 4.2, ein eigener Konvertierer implementiert. Dieser wurde mit dem *Template-Method*-Pattern [41] umgesetzt. Wie im Klassendiagramm in Abbildung 7 dargestellt, wurde dafür die Klasse **Question** implementiert. Diese definiert die Struktur der Konvertierung, aber erlaubt abgeleiteten Klassen Teilbäume zu erstellen, ohne dass die Struktur verändert wird. In dieser Arbeit wurden für die vier Aufgabentypen **Single-Choice**, **Multiple-Choice**, **Drag-and-Drop** und **Textbox** abgeleitete Klassen erstellt.

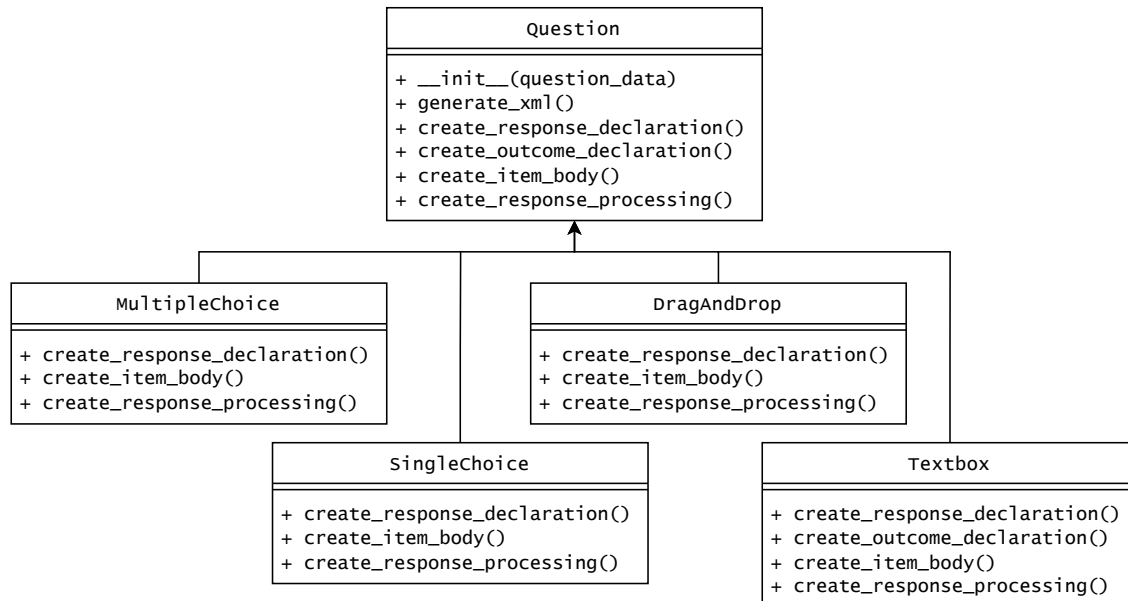


Abb. 7: Klassendiagramm nach dem *Template-Method-Pattern*, eigene Darstellung

Wie Abbildung 7 zu entnehmen ist, lässt sich der XML-Baum einer QTI-Aufgabe in verschiedene Unterbäume strukturieren. Während des Initialisierens einer **Question**-Instanz wird der Header erstellt, dem jeder weitere Baum untergeordnet wird. In diesem wird beispielsweise der Titel der Aufgabe und die QTI-Version festgehalten. Anschließend wird eine `responseDeclaration` erstellt, welche die richtige Lösung angibt. In der `outcomeDeclaration` wird angegeben, wie viele Punkte erreichbar sind. `itemBody` beinhaltet die Aufgabenstellung und die Antwortmöglichkeiten. In diesem wird ebenfalls mit dem

Attribut `shuffle` festgelegt, dass die Antwortmöglichkeiten in zufälliger Reihenfolge angezeigt werden sollen. Darauffolgend kommt das `responseProcessing`. Hier wird die Logik der Bewertung festgehalten, wie zum Beispiel ob die erreichten Punkte in dem festgelegten Bereich liegen und ob die maximale Punktzahl erreicht wurde. Abschließend wird das Feedback zugeordnet.

Beim Umwandeln einer Aufgabe wird nach dem angegebenen Aufgabentyp eine Instanz der speziellen Klasse erstellt und anschließend der XML-Baum gespeichert. Die Erstellung des Baums wird mit dem Modul *ElementTree* [64] aus der Python Standard Library umgesetzt.

5.7 Docker

Für die Bereitstellung der Anwendung wird *Docker* [18] genutzt. Die Umsetzung dessen orientiert sich erneut an der Implementierung von Meissner und Pögelt [38]. Mithilfe der *compose.yml* und der *Dockerfile* kann die Anwendung samt Abhängigkeiten leicht in einem Container aufgebaut werden. Dafür wird mit dem Befehl `docker compose up` die Dockerfile gebaut und der Service `task-generator`, definiert in der *compose.yml*, gestartet. Dabei werden *Volumes* verwendet, um die Ausgaben und Sprachmodelle zu persistieren. Mit dem Attribut `ports` wird der Port 8501, welcher von Streamlit verwendet wird, zwischen der Hostmaschine und dem Container zugeordnet. Das ist notwendig, um den Zugriff auf die Anwendung zu ermöglichen, welche dadurch unter `localhost:8501` auffindbar ist. Mit dem Befehl `docker compose down` wird der Service gestoppt und entfernt.

6. EVALUATION

Das folgende Kapitel befasst sich mit der Darstellung und Gegenüberstellung der menschlichen und maschinellen Evaluation. Zunächst werden die verwendeten Bewertungsmetriken vorgestellt, bevor die Ergebnisse der Phasen betrachtet und evaluiert werden. Abschließend wird die Generierung von Aufgaben, die Übereinstimmung der maschinellen Evaluation mit der menschlichen Wahrnehmung und die Limitationen dieser Arbeit diskutiert.

6.1 Bewertungsmetriken

Um die Ergebnisse dieser Arbeit zu bewerten, wird auf Metriken für Klassifikationsmodelle zurückgegriffen. Verwendet werden die etablierten Metriken *Accuracy*, *Precision*, *Recall* und *F1-Score* [59]. Bei diesen wird jeweils mit bestimmten Ereignissen gearbeitet:

- *True Positive* (TP) – ein Qualitätsmerkmal wurde korrekt erkannt
- *False Positive* (FP) – ein mangelhaftes Merkmal wurde als korrekt erkannt
- *True Negative* (TN) – ein mangelhaftes Merkmal wurde korrekt erkannt
- *False Negative* (FN) – ein wahres Merkmal wurde falsch klassifiziert

Für diese Ereignisse wird die Evaluation der Expertin als *Ground Truth* verwendet.

Accuracy gibt an, wie viele richtige (TP und TN) Klassifikationen über alle Aussagen getroffen wurden [59]. Diese Metrik gibt einen allgemeinen Überblick zu der Leistung des Modells.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Precision beschreibt, wie viele der getroffenen richtigen Aussagen wirklich der Realität entsprechen [59]. Relevant ist diese Metrik besonders, wenn FP mit hohen Kosten verbunden ist.

$$Precision = \frac{TP}{TP + FP}$$

Recall zeigt, wie viele der Aufgaben ohne mangelhaftes Qualitätsmerkmal auch als dieses klassifiziert wurden [59]. Wenn das Finden aller Vorkommnisse von Bedeutung ist, wird besonders auf dieses Merkmal geachtet.

$$Recall = \frac{TP}{TP + FN}$$

F1-Score besteht aus dem harmonischen Mittel von Precision und Recall [59]. Diese Metrik bietet ebenfalls einen allgemeinen Überblick, besonders wenn der Datensatz unausgewogen ist.

$$F = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Im Kontext dieser Arbeit gibt die Precision ein Anzeichen, ob die maschinelle Evaluation als alleinstehende Filterinstanz geeignet ist. Ein niedriger Wert bedeutet, dass mehr Aufgaben von niedriger Qualität im tech4compKI-Projekt weitergereicht werden. Somit sind aufwendigere Korrekturen bei der weiteren Beurteilung der Aufgaben notwendig. Der Recall hingegen ist, bezogen auf diese Arbeit, relevant bei einer Filterung der Aufgaben nach der Beurteilung der maschinellen Evaluation, bei der die Generierungskosten und die hochqualitativen FNs verschwendet würden.

Die Berechnung der Werte und die Darstellung dieser in Konfusionsmatrizen wurde für diese Arbeit mithilfe von *scikit-learn* [47] und *Matplotlib* [27] umgesetzt. Die dafür erstellten Jupyter-Notebooks sind ebenfalls unter in dem GitLab-Projekt [53] einzusehen.

6.2 Phase 1

In der ersten Phase wurden für 16 mathematische Konzepte, gesammelt aus dem Skript [23], Übungsaufgaben in jedem unterstützten Aufgabentyp generiert. Die 64 Aufgaben wurden anschließend von der Expertin und maschinell evaluiert.

Taxonomie Bei der Wissensdimension ergibt die Evaluation der Expertin, dass alle 64 Aufgaben in der Dimension *Faktenwissen* einzuordnen sind. Dies entspricht der intendierten Wissensdimension. Eine Aufgabe wurde von der maschinellen Evaluation als *Konzeptionelles Wissen* klassifiziert. Die anderen Aufgaben wurden ebenfalls als *Faktenwissen* eingestuft. Demnach ergibt sich für die maschinelle Evaluation ein F1-Score von 0.99.

Die Expertin ordnet ebenfalls alle Aufgaben in die intendierte kognitive Prozessdimension *Erinnern* ein. Tabelle 3 zeigt die Einordnung der Aufgaben durch die maschinelle Evaluation.

	Erinnern	Verstehen	Anwenden	Ana- lysieren	Beurteilen	Erstellen
Anzahl	39	14	–	11	–	–

Tabelle 3: Maschinelle Evaluation der kognitiven Prozessdimensionen in der ersten Phase

Dabei ist dargestellt, dass 38 Aufgaben ebenfalls als *Erinnern* klassifiziert wurden; dies ent-

spricht einem F1-Score von 0.76. Auch die nächst abstraktere Dimension *Verstehen* wurde von der maschinellen Evaluation für 14 Aufgaben angegeben. Auffallend ist dabei die Klassifizierung in die Dimension *Analysieren*, welche deutlich komplexer als *Erinnern* ist.

Diese Einordnung betrifft vor allem den Aufgabentyp *Drag-and-Drop*. Bei der Betrachtung der gegebenen Begründung, gibt das LLM an, dass die Zuordnung der Optionen zueinander erfordert, die einzelnen Bestandteile des Konzepts aufzugliedern und die Beziehung dieser zueinander zu bestimmen. Insbesondere der Prozess des Zuordnens und das Unterscheiden sowie Organisieren von Eigenschaften, wird als ausschlaggebend für diese Klassifizierung angegeben.

Qualität Grundsätzlich waren alle 64 erstellten Übungsaufgaben problemlos ausführbar. Bei der *Korrektheit der Aufgabenstellung* (0.77 Precision, 0.90 Recall) war die maschinelle Evaluation weniger streng. Insgesamt wurden 13 Aufgaben von der Expertin als mangelhaft bewertet, welche bei der maschinellen Evaluation nicht angemerkt wurden. Hingegen sind bei dem Merkmal der *korrekten Auswertung* (0.84 Precision, 0.85 Recall) beide Werte nah beieinander; allgemein wurden nur wenige Mängel festgestellt. Die *Feedbackzweige* (0.85 Precision, 0.88 Recall) wurden von der maschinellen Evaluation mit einem etwas höheren Recall als Precision ebenfalls weniger streng bewertet.

Auffällig niedrig ist die *kontextuelle Korrektheit* (0.65 Precision, 0.32 Recall), besonders im Recall. Weitaus mehr Aufgaben wurden in der maschinellen Evaluation als mangelhaft klassifiziert. Dies könnte mit der leicht anderen Bedeutung dieses Qualitätsmerkmals von der maschinellen und Experten-Evaluation, wie sie in Abschnitt 4.3.1 beschrieben ist, zusammenhängen. Bei der maschinellen Evaluation wurde demnach ein anderer kontextueller Fokus der Aufgabe festgestellt, während die Expertin häufig bemängelt hat, dass der Ausdruck nicht der Fachsprache entspricht. Insgesamt war ein Mangel in der kontextuellen Korrektheit jedoch in beiden Evaluationen am häufigsten festgestellt worden.

Konklusion Die Evaluation der Expertin ergab, dass 39 von den 64 generierten Aufgaben der ersten Phase von hoher Qualität waren. Die Abbildung 8 zeigt die Konfusionsmatrix mit dem Ergebnis der maschinellen Evaluation. Diese zeigt, dass die Precision und der Recall mit 0.74 identisch sind, was auf eine balancierte Leistung beim Identifizieren von hochqualitativen Aufgaben hinweist. Die FPs und FNs werden gleichermaßen minimiert; insgesamt kann die Leistung jedoch noch gesteigert werden.

Für die zweite Phase wurden die Informationstexte der Dimensionen angepasst, da die Ergebnisse der kognitiven Prozessdimensionen darauf hingewiesen haben, dass diese den Fokus auf die Wortwahl der Aufgaben legten, anstatt auf die geforderten Kompetenzen. Zusätzlich war bei der maschinellen Evaluation eine mangelhafte *kontextuelle Korrektheit*,

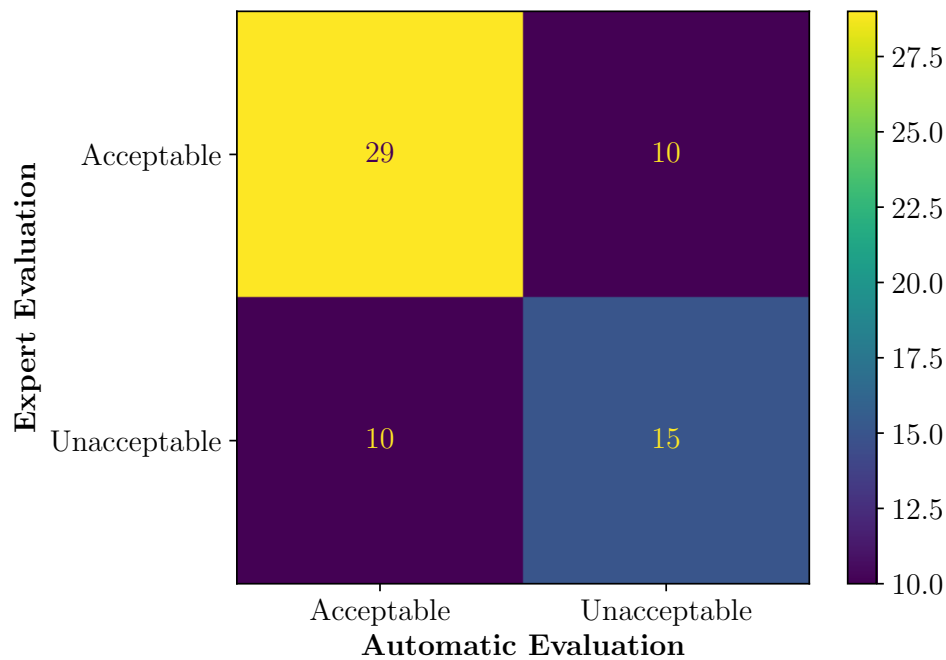


Abb. 8: Maschinelle Entscheidung über die Qualität der Übungsaufgaben in der ersten Phase im Vergleich zu der menschlichen, eigene Darstellung

wegen der leicht anderen Bedeutung, nicht ausschlaggebend bei der Entscheidung über die Qualität der Übungsaufgaben insgesamt. Damit eine Aufgabe, welche nicht den intendierten Kontext vorrangig behandelt, nicht unberücksichtigt bleibt, wird in der zweiten Phase ein alleiniger Mangel in der kontextuellen Korrektheit ebenfalls als mangelhaft bewertet. Zudem wurden Experten-Klassifikationen aus der ersten Phase als Beispielklassifikationen (Few-Shot Prompting) verwendet.

6.3 Phase 2

Für die zweite Phase wurde ein *Sample* von 75 mathematischen Konzepten zufällig ausgewählt und zu jedem Aufgabentyp eine Aufgabe erstellt. Somit umfasste diese Phase 300 Aufgaben, welche maschinell evaluiert wurden. Von diesen wurde für die menschliche Evaluation pro Aufgabentyp ein Sample von 20 Aufgaben bewertet, womit das *Experten-Sample* insgesamt 80 Aufgaben umfasste.

Taxonomie Die Expertin klassifizierte erneut alle 80 generierten Übungsaufgaben als *Faktenwissen*. Dahingegen ordnete die maschinelle Evaluation, welche in Tabelle 4 dargestellt ist, Aufgaben ebenfalls in die Dimensionen *Konzeptionelles Wissen* und *Prozedurales Wissen* ein.

	Faktenwissen	Konzeptionelles Wissen	Prozedurales Wissen	Metakognitives Wissen
Anzahl beim Experten-Sample	74	2	4	–
Anzahl bei allen 300 Aufgaben	270	20	10	–

Tabelle 4: Maschinelle Evaluation der Wissensdimensionen in der zweiten Phase

Bei dem Experten-Sample hat die maschinelle Evaluation einen F1-Score von 0.96 und ist damit niedriger als in der ersten Phase. Auffallend ist hier die abweichende Klassifizierung in die deutlich komplexere Wissensdimension *Prozedurales Wissen*. Als Grund wird vom LLM angegeben, dass das Verständnis für fachspezifische Methoden benötigt wird, was in diese Dimension fällt.

Bei der kognitiven Prozessdimension der erstellten Aufgaben ergab die Evaluation der Expertin, dass 79 der 80 Aufgaben der Dimension *Erinnern* entsprechen. Abweichend wurde eine Aufgabe der Dimension *Verstehen* zugeordnet. Die maschinelle Evaluation klassifiziert ebenfalls den Großteil der Aufgaben in die Dimension *Erinnern* (0.99 Precision, 0.96 Recall), wie in Tabelle 5 dargestellt ist.

	Erinnern	Verstehen	Anwenden	Ana- lysieren	Beurteilen	Erstellen
Anzahl beim Experten-Sample	76	2	1	–	–	–
Anzahl bei allen 300 Aufgaben	294	4	2	–	–	–

Tabelle 5: Maschinelle Evaluation der kognitiven Prozessdimensionen in der zweiten Phase

Die maschinelle Evaluation der kognitiven Prozessdimension hat sich in der zweiten Phase deutlich mit einem F1-Score von 0.96 verbessert. Dies zeigt, dass die Anpassung der Informationstexte sowie das Hinzunehmen der Beispielklassifikationen (Few-Shot Prompting) den gewünschten Effekt erbracht hat.

Qualität In der zweiten Phase wurde erneut von beiden Evaluationen angegeben, dass alle generierten Aufgaben korrekt ausführbar waren. Die maschinelle Evaluation der *Korrektheit der Aufgabenstellung* (0.80 Precision, 0.88 Recall) blieb weniger streng im Vergleich zur Beurteilung der Expertin, verbesserte sich jedoch im Vergleich zur ersten Phase. Die *korrekte Auswertung* (0.88 Precision, 0.97 Recall) steigert sich ebenfalls in der maschinellen

Evaluation. Dahingegen hat sich in der zweiten Phase die *Korrektheit der Feedbackzweige* (0.82 Precision, 0.89 Recall) leicht verschlechtert. Allerdings verbesserte sich die Evaluation der *kontextuellen Korrektheit* (0.62 Precision, 0.64 Recall). Die Accuracy stieg von 0.45 in der ersten Phase auf 0.53 in der zweiten. Trotz der Anpassung des Prompts hat besonders die Evaluation der kontextuellen Korrektheit Probleme, die Entscheidung richtig zu formulieren. Grundsätzlich ist die maschinelle Evaluation der kontextuellen Korrektheit weiterhin schlechter als die anderen Qualitätsmerkmale.

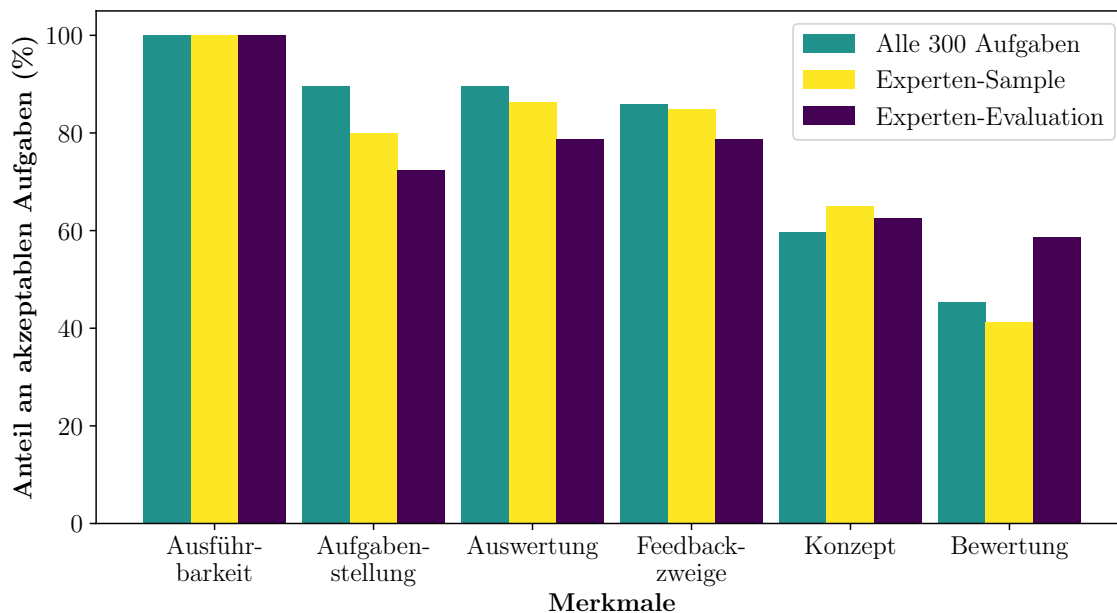


Abb. 9: Anteil an akzeptablen Aufgaben bei den 300 Aufgaben im Vergleich zu den Evaluationen des Experten-Samples, eigene Darstellung

Das Balkendiagramm in Abbildung 9 stellt dar, wie der Anteil an akzeptablen Aufgaben pro Qualitätsmerkmal von der maschinellen Evaluation (über die 300 Aufgaben und über das Experten-Sample) im Vergleich zur Experten-Evaluation ausfällt. Dabei ist zu sehen, dass über die 300 Aufgaben ein größerer Anteil als akzeptabel bewertet wurde, als bei den Evaluationen beim Experten-Sample. Insgesamt ist die Bewertung der Expertin ähnlich zur maschinellen Evaluation. Auffallend ist die starke Abweichung bei der Bewertung.

Konklusion Von den Übungsaufgaben der zweiten Phase hat die Expertin 47 der 80 als hochwertig befunden. Die maschinelle Evaluation hat diese Aufgaben strenger bewertet, und hat 33 als hochwertig beurteilt. Die Konfusionsmatrix in Abbildung 10 stellt die Evaluationen dar.

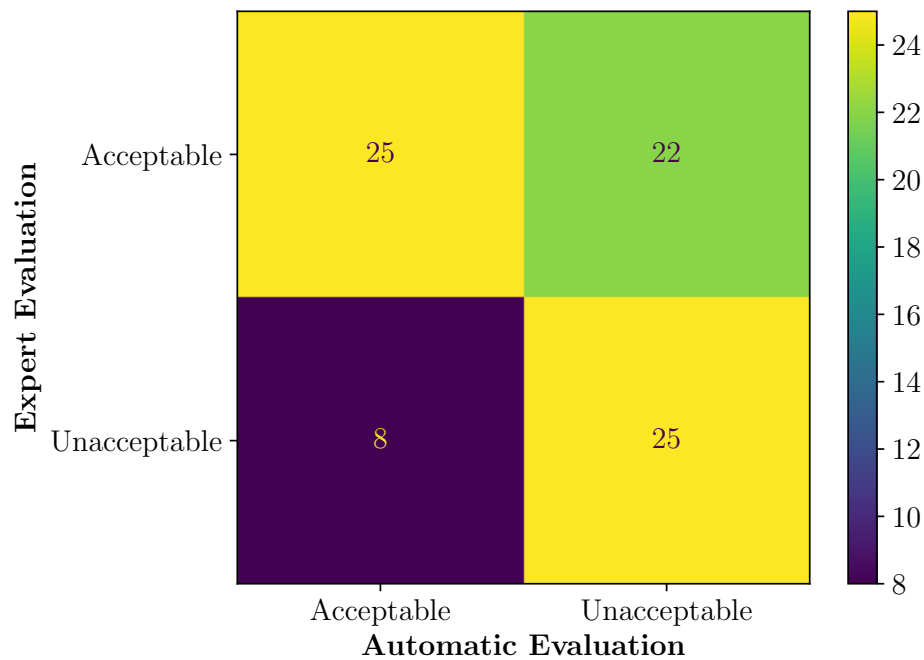


Abb. 10: Maschinelle Entscheidung über die Qualität der 80 Übungsaufgaben in der zweiten Phase insgesamt im Vergleich zu der menschlichen, eigene Darstellung

Insgesamt sank die Accuracy auf 0.63. Die Anzahl an FNs ist sichtbar gestiegen und führt zu einem niedrigeren Recall von 0.53 im Vergleich zur ersten Phase (0.74). Dafür stieg in der zweiten Phase die Precision von 0.74 auf 0.76. Das deutet daraufhin, dass das Potenzial besteht, durch weitere Anpassungen die Precision dahingehend steigern zu können, dass damit trotz den Abweichungen zur menschlichen Wahrnehmung, eine grundlegende Filterinstanz entstehen kann.

6.4 Diskussion

In dieser Arbeit konnten Übungsaufgaben für ein intendiertes mathematisches Konzept mit der Wissensdimension *Faktenwissen* und der kognitiven Prozessdimension *Erinnern* durch das verwendete LLM *GPT-4o* generiert werden. Diese wurden maschinell evaluiert und von JSON in ausführbare QTI-Aufgaben konvertiert.

Aufgabengenerierung Die in Kapitel 3 formulierten Anforderungen an eine Aufgabe wurden entsprechend der Evaluation von 144 Aufgaben durch die Expertin zu 60 % eingehalten. Bis auf eine Aufgabe, bei der als kognitive Prozessdimension *Verstehen* festgestellt wurde, wurden alle Aufgaben in den intendierten Dimensionen erstellt. Auch der Ausreißer war, nach den Anforderungen des Kapitels 3, akzeptabel. Insgesamt zeigte die Generierung

demnach eine starke Leistung bei der Erstellung von Aufgaben in den unteren Taxonomiestufen.

Besonders wurde auch die Fähigkeit dargestellt, dass das LLM die Aufgaben im gewünschten Format erstellen kann, was somit zu einer korrekten Aufführbarkeit führte. Problematisch war vor allem die korrekte Verwendung von Fachtermini. Dies führte auch dazu, dass beim Qualitätsmerkmal *kontextuelle Korrektheit* die meisten Mängel festgestellt wurden.

Bei den umgesetzten Aufgabentypen wurde kein auffälliger Unterschied in der Qualität festgestellt. Die *Textboxaufgaben* hatten mit 24 von 36 Aufgaben die meisten hochqualitativen Aufgaben, während *Multiple-Choice*-Aufgaben, besonders durch die Mängel in der Auswertung, mit 19 die wenigsten hatten.

Die Expertin appellierte, dass die Aufgaben zunächst von einer fachkundigen Person betrachtet werden sollten, bevor sie in den Aufgabenpool gegeben werden. Sie meldete ebenfalls zurück, dass diese Automatisierung der Aufgabenerstellung unterstützend sei, da sie inspirierend für neue Aufgaben sind und auch durch leichtes Anpassen der Aufgaben dafür sorgen, dass die Lehrenden entlastet werden. Von den 144 generierten Aufgaben dieser Arbeit gab die Expertin an, 70 Aufgaben bereits in den Aufgabenpool integriert zu haben, davon 21 % nach stärkerer Abänderung. Insgesamt kam sie zu dem Schluss, dass der Prozess der Aufgabenerstellung durch die Ergebnisse dieser Arbeit erleichtert wurde.

Maschinelle Evaluation In dieser Arbeit wurde der Frage nachgegangen, inwiefern eine maschinelle Evaluation der generierten Aufgaben mit der menschlichen Wahrnehmung übereinstimmt. Hierfür wurden die Klassifizierung in die Taxonomie von Anderson und Krathwohl [3], die Beurteilung der Qualität sowie die Entscheidung über die Aufgabe insgesamt betrachtet. Repräsentativ für die menschliche Wahrnehmung wurden die erstellten Aufgaben von einer Expertin evaluiert.

Das Ergebnis war, dass die Bewertung der generierten Aufgaben dieser Arbeit zu 63 % übereinstimmte. In der zweiten Phase war die maschinelle Evaluation strenger und hat weniger Aufgaben als hochqualitativ beurteilt als die Expertin. Dies ist positiv für eine allein stehende Filterinstanz, da so die Lehre der Studierenden nicht durch mangelhafte Übungsaufgaben gestört wird. Mit der Precision von 0.76 ist aber davon abzuraten, ausschließlich die maschinelle Evaluation zur Beurteilung der Qualität zu verwenden.

In der zweiten Phase war die Klassifizierung der kognitiven Prozess- und Wissensdimensionen mit einem F1-Score von jeweils 0.96 besonders leistungstark. Die niedrigste Accuracy hatte das Bestimmen der kontextuellen Korrektheit mit 0.60 in der zweiten Phase. Dies kann auf anderen Fokus des Qualitätsmerkmals in der maschinellen Evaluation zurückzuführen sein. In dieser wird betrachtet, welches mathematische Konzept in der Aufgabe behandelt wird. Demnach bedarf dieses Merkmal eindeutig weiterer Anpassung, um der menschlichen

Wahrnehmung zu entsprechen.

Am meisten stimmte die Evaluation der Textboxaufgaben mit der Evaluation der Expertin überein; die Accuracy betrug 0.70 in der zweiten Phase. Darauf folgen die Aufgabentypen *Drag-and-Drop* und *Multiple-Choice* mit einer Accuracy von 0.65. Dabei wurde bei den *Drag-and-Drop*- und *Textboxaufgaben* keine mangelhafte Aufgabe fälschlich als hochwertig klassifiziert. Dies zeigt, dass die Leistung je nach Aufgabentyp variiert und bereits eine hohe Precision erreicht werden konnte. Somit besteht Potenzial, die Leistung der maschinellen Evaluation insgesamt bzw. als Filterinstanz weiter zu verbessern.

Einschränkungen Die Generierung und Evaluation der Übungsaufgaben wurden unter Einsatz von LLMs umgesetzt. Die Antworten von LLMs sind trotz der verwendeten niedrigen Temperatur nicht deterministisch, was besonders auf die Evaluation bezogen bedeutet, dass beim erneuten Evaluieren einer Aufgabe die Entscheidung über das Merkmal anders sein kann. Zudem kann aus verschiedenen Gründen, wie beispielsweise einem Netzwerkfehler, die Antwort in mangelhafter Qualität oder überhaupt nicht erstellt werden.

Für das Generieren der Aufgaben ist domänenspezifisches Wissen erforderlich, wobei dieses nicht bei LLMs vorausgesetzt werden kann. Die Qualität der generierten Aufgaben hängt stark davon ab, wie präzise und detailliert das intendierte mathematische Konzept in den Prompts reflektiert wird. Wird das Konzept nicht gründlich behandelt oder ungenau formuliert, kann dies die Qualität der Aufgaben erheblich beeinträchtigen. In dieser Arbeit wurde mit dem in Abschnitt 5.4 genannten Material gearbeitet; die Qualität der Aufgaben ist demnach von diesem Material und dessen Zusammenfassung für den Prompt abhängig. Zukünftige Arbeiten könnten sich darauf konzentrieren, das domänenspezifische Wissen in den Prompts weiter zu verfeinern und zu optimieren, um die Qualität der generierten Aufgaben weiter zu verbessern.

Eine weitere Anmerkung gilt der menschlichen Evaluation. Während diese theoretisch keine Einschränkungen besitzt, sofern Zeit und Geld vorhanden sind, ist die Anzahl der Experten entscheidend für die Widerspiegelung der menschlichen Wahrnehmung. In dieser Arbeit setzt sich die menschliche Evaluation alleine aus einer einzigen Expertin zusammen. Die Qualität der Übungsaufgaben und deren Klassifikation kann deswegen von individuellen Präferenzen, Erfahrungen und Einschätzungen beeinflusst sein. Dadurch besteht eine wesentliche Einschränkung hinsichtlich der Allgemeingültigkeit der Ergebnisse. Daher sollten zukünftige Studien idealerweise von mehreren Experten evaluiert werden, um eine ausgewogene Einschätzung zu ermöglichen. Folgend könnte ebenfalls eine Anwendung zur Qualitätssicherung von *E-Assessments*, wie *EAs.LiT* [60], für eine kollaborative Durchführung der Evaluationen verwendet werden.

7. ZUSAMMENFASSUNG

Ziel dieser Arbeit war es mathematische Übungsaufgaben unter der Verwendung von LLMs zu erstellen. Es war gefordert, dass diese im QTI-Format vorliegen, damit das Bereitstellen der Aufgaben für die Studierenden vereinfacht wird. Dabei wurde untersucht, inwiefern eine maschinelle Evaluation der Aufgaben mit der menschlichen Wahrnehmung übereinstimmt. Dafür wurden zunächst, wie in Kapitel 4 beschrieben, verschiedene LLMs verglichen. Dabei erzielte *GPT-4o* von *OpenAI* die beste Leistung und wurde für diese Arbeit ausgewählt. Daraufhin wurde eine Anwendung entwickelt, welche nach den, in Kapitel 3 formulierten Anforderungen und Erkenntnissen aus Kapitel 4, mathematische Übungsaufgaben in der Wissensdimension *Faktenwissen* und der kognitiven Prozessdimension *Erinnern* nach der Taxonomie von Anderson und Krathwohl [3] erstellt. Diese wurden anschließend NLP-basiert maschinell evaluiert und mittels des implementierten Konverters in das QTI-Format umgewandelt. Das Programm wurde über Docker bereitgestellt und die Bedienung mit einer mittels *Streamlit* erstellten Benutzeroberfläche erleichtert.

Für die Betrachtung der Qualität der generierten Aufgaben und der Tauglichkeit der maschinellen Evaluation wurden insgesamt 144 Aufgaben von einer Expertin beurteilt. Die menschliche Evaluation ergab, dass 60 % der generierten Übungsaufgaben eine hohe Qualität aufwiesen. Die Aufgaben entsprachen mit sehr hoher Präzision der intendierten kognitiven Prozess- und Wissensdimension und waren stets korrekt ausführbar.

Die Aufgaben wurden in zwei Phasen generiert, wobei nach der ersten Phase die maschinelle Evaluation angepasst wurde.

Die Überprüfung, ob die maschinellen Evaluation mit der menschlichen Wahrnehmung übereinstimmt, hat ergeben, dass dies zu 63 % der Fall war. Besonders die Klassifikation der kognitiven Prozess- und Wissensdimensionen waren mit hoher Präzision repräsentiert; der F1-Score lag bei jeweils 0.96 in der zweiten Phase. Das Betrachten der Leistung bei den einzelnen Aufgabentypen zeigte, dass in der zweiten Phase die Aufgabentypen *Drag-and-Drop* und *Textbox* eine Precision von 1.0 erzielten und somit Potenzial für weitere Verbesserungen besteht.

Insgesamt erfüllt der entwickelte Aufgabengenerator dieser Arbeit die Anforderungen. Auch wenn die maschinelle Evaluation nicht ausreicht, um die Qualität der generierten Aufgaben in Übereinstimmung mit der menschlichen Wahrnehmung zu beurteilen, ermöglicht die Anwendung durch die automatisierte Erstellung der Übungsaufgaben, den Aufwand der Lehrenden zu reduzieren. Damit zeigt diese Arbeit ein Beispiel für das Potenzial von LLMs, wie diese die Lehre unterstützen und Raum für möglichst individuelles Mentoring schaffen können.

Ausblick Der Bereich rund um LLMs entwickelt sich besonders rasant weiter. Dementsprechend kann an dieser Stelle in zukünftigen Arbeiten angesetzt werden. Durch die Verwendung von *LangChain* lassen sich andere LLMs leicht in die Anwendung integrieren. Zudem ist ein vielversprechender Ansatz, ein kleines Open-Source-Sprachmodell mit den Prompt-Output-Paaren, welche bei Anfragen dieser Arbeit gespeichert werden, zu trainieren.

Die Optimierung der Prompts und Parameter des LLMs ist ein weiterer Ansatz für weitere Forschung. Da das Ergebnis der LLMs empfindlich gegenüber der gegebenen Aufforderung und deren Wortwahl ist, kann weiter an der Leistungssteigerung mittels Prompt Engineering geforscht werden. Dies bezieht sich ebenfalls auf die, in den Limitationen bereits genannte, Optimierung des mitgegebenen Fachwissens. Zusätzlich können weitere Anfragen an das LLM in der maschinellen Evaluation gestellt werden, welche spezifischere Qualitätsmerkmale abfragen, um so der menschlichen Wahrnehmung besser zu entsprechen.

Die Funktionalität der Anwendung kann ebenfalls weiter ausgebaut werden. So ist es denkbar, weitere Aufgabentypen zu implementieren, wie zum Beispiel Lückentextaufgaben, und zu ermöglichen, Aufgaben mit Kombinationen von mathematischen Konzepten zu generieren. Zudem kann die Bewertung der Qualitätsmerkmale von einer binären Klassifikation auf eine Regression verändert werden, um zu untersuchen inwiefern dies eine Verbesserung der Übereinstimmung im Vergleich zur menschlichen Wahrnehmung bringt.

Innerhalb des tech4compKI-Projekts könnte der Frage nachgegangen werden, wie effizient Dopplungen von Aufgaben in dem Aufgabenpool festgestellt bzw. verhindert werden können. Bei mehrfacher Generierung von Übungsaufgaben mit demselben Aufgabentyp und mathematischen Konzept gilt es Folgendes zu beachten: Trotz der gewählten niedrigen Temperatur in dieser Arbeit ist es möglich, dass unterschiedliche Aufgaben erstellt werden, welche sich sehr ähnlich können. Daher ist die Entscheidung, ab wann eine Aufgabe bereits im Aufgabenpool vorhanden ist oder nicht, keineswegs trivial.

Abbildungsverzeichnis

1	Übersicht der tech4compKI-Services, Darstellung aus [43]	4
2	Beispiel eines Few-Shot Prompting, eigene Darstellung	7
3	Herangehensweise von dieser Arbeit, eigene Darstellung	16
4	Ablauf der Anwendung, eigene Darstellung	19
5	Benutzeroberfläche der Anwendung, eigener Screenshot	20
6	Anzeige während und nach der Generierung einer Aufgabe, eigene Screenshots	21
a	Anzeige der Generierung einer Aufgabe	21
b	Warnung über niedrige Qualität	21
7	Klassendiagramm nach dem Template-Method-Pattern, eigene Darstellung .	26
8	Konfusionsmatrix der Qualität der Aufgaben (erste Phase), eigene Darstellung	31
9	Anteil akzeptabler Aufgaben der Evaluationen (zweite Phase), eigene Darstellung	33
10	Konfusionsmatrix der Qualität der Aufgaben (zweite Phase), eigene Darstellung	34
11	Screenshot der Single-Choice-Aufgabe in ONYX	59
12	Screenshot der Multiple-Choice-Aufgabe in ONYX	66
13	Screenshot der Drag-and-Drop-Aufgabe in ONYX	69
14	Screenshot der Textboxaufgabe in ONYX	73

Tabellenverzeichnis

1	Testergebnisse der betrachteten LLMs	14
2	Ausschnitt aus der Ergebnistabelle der Expertin	18
3	Maschinelle Evaluation der kognitiven Prozessdimensionen (erste Phase) . .	29
4	Maschinelle Evaluation der Wissensdimensionen (zweite Phase)	32
5	Maschinelle Evaluation der kognitiven Prozessdimensionen (zweite Phase) .	32
6	Evaluationen der Single-Choice-Aufgabe	64
7	Evaluationen der Multiple-Choice-Aufgabe	67
8	Evaluationen der Drag-and-Drop-Aufgabe	71
9	Evaluationen der Textboxaufgabe	74

Quellcodeverzeichnis

1	Format einer <i>Single-Choice</i> -Aufgabe die von <i>text2qti</i> umgewandelt werden kann [48]	15
2	Darstellung einer Single-Choice-Aufgabe in JSON	22
3	Aufruf zur Aufgabenerstellung mit Output-Korrektur und Logging	24
4	Prompt für die Aufgabengenerierung beim LLM-Test	49
5	Prompt für die Evaluation beim LLM-Test	51
6	Ausgabe der Aufgabengenerierung von GPT-3.5 Turbo	52
7	Ausgabe der Aufgabengenerierung (gekürzt) von GPT-3.5 Turbo	52
8	Ausgabe der Evaluation von GPT-3.5 Turbo	53
9	Ausgabe der Aufgabengenerierung von GPT-4o	53
10	Ausgabe der Aufgabengenerierung (gekürzt) von GPT-4o	54
11	Ausgabe der Evaluation von GPT-4o	54
12	Ausgabe der Aufgabengenerierung von Llemma 7B	55
13	Ausgabe der Aufgabengenerierung (gekürzt) von Llemma 7B	55
14	Ausgabe der Evaluation von Llemma 7B	56
15	Ausgabe der Aufgabengenerierung (gekürzt) von Llemma 34B	56
16	Ausgabe der Evaluation von Llemma 34B	56
17	Ausgabe der Aufgabengenerierung von InternLM-Math 7B	57
18	Ausgabe der Aufgabengenerierung (gekürzt) von InternLM-Math 7B	58
19	Ausgabe der Evaluation von InternLM-Math 7B	58
20	Darstellung der Single-Choice-Aufgabe in JSON	59
21	Single-Choice-Aufgabe im QTI-Format in verkürzter Form	60
22	Darstellung der Multiple-Choice-Aufgabe in JSON	66
23	Darstellung der Drag-and-Drop-Aufgabe in JSON	70
24	Darstellung der Textboxaufgabe in JSON	73

Literaturverzeichnis

- [1] M. Afzaal, J. Nouri und A. Aayesha. „A Transformer-Based Approach for the Automatic Generation of Concept-Wise Exercises to Provide Personalized Learning Support to Students“. In: *Responsive and Sustainable Educational Futures*. Hrsg. von O. Viberg et al. Cham: Springer Nature Switzerland, 2023, S. 16–31. ISBN: 978-3-031-42682-7. DOI: [10.1007/978-3-031-42682-7_2](https://doi.org/10.1007/978-3-031-42682-7_2).
- [2] A. Agarwal et al. *Understanding the Role of Temperature in Diverse Question Generation by GPT-4*. 14. Apr. 2024. DOI: [10.1145/3626253.3635608](https://doi.org/10.1145/3626253.3635608). arXiv: [2404.09366](https://arxiv.org/abs/2404.09366) [cs]. URL: <http://arxiv.org/abs/2404.09366> (besucht am 04.06.2024). Vorveröffentlichung.
- [3] L. W. Anderson und D. Krathwohl. *A Taxonomy for Learning, Teaching, and Assessing : A Revision of Bloom's Taxonomy of Educational Objectives*. New York : Longman, 2001. 396 S. ISBN: 978-0-321-08405-7 978-0-8013-1903-7. URL: <http://archive.org/details/taxonomyforlearn0000unse> (besucht am 26.06.2024).
- [4] *API Reference*. Streamlit Docs. URL: <https://docs.streamlit.io/develop/api-reference> (besucht am 03.07.2024).
- [5] T. Arens et al. *Grundwissen Mathematikstudium*. 2. Aufl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. ISBN: 978-3-8274-2308-5 978-3-8274-2309-2. DOI: [10.1007/978-3-8274-2309-2](https://doi.org/10.1007/978-3-8274-2309-2). URL: <http://link.springer.com/10.1007/978-3-8274-2309-2> (besucht am 05.07.2024).
- [6] T. Arens et al. *Mathematik*. 5. Aufl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2022. ISBN: 978-3-662-64388-4 978-3-662-64389-1. DOI: [10.1007/978-3-662-64389-1](https://doi.org/10.1007/978-3-662-64389-1). URL: <https://link.springer.com/10.1007/978-3-662-64389-1> (besucht am 05.07.2024).
- [7] Z. Azerbayev et al. *Llemma: An Open Language Model For Mathematics*. 15. März 2024. DOI: [10.48550/arXiv.2310.10631](https://doi.org/10.48550/arXiv.2310.10631). arXiv: [2310.10631](https://arxiv.org/abs/2310.10631) [cs]. URL: <http://arxiv.org/abs/2310.10631> (besucht am 21.04.2024). Vorveröffentlichung.

- [8] J. Biggs. „Enhancing Teaching through Constructive Alignment“. In: *Higher Education* 32.3 (1. Okt. 1996), S. 347–364. ISSN: 1573-174X. DOI: [10.1007/BF00138871](https://doi.org/10.1007/BF00138871). URL: <https://doi.org/10.1007/BF00138871> (besucht am 17.05.2024).
- [9] B. S. Bloom. *Taxonomy of Educational Objectives: The Classification of Educational Goals*. New York, Longmans, Green, 1956. 214 S. ISBN: 978-0-679-30209-4 978-0-582-32386-5 978-0-582-32385-8. URL: http://archive.org/details/taxonomyofeducat0000bloo_o9o7 (besucht am 26.06.2024).
- [10] T. B. Brown et al. „Language Models Are Few-Shot Learners“. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS ’20. Red Hook, NY, USA: Curran Associates Inc., 6. Dez. 2020, S. 1877–1901. ISBN: 978-1-71382-954-6.
- [11] Z. Cai et al. *InternLM2 Technical Report*. 25. März 2024. arXiv: [2403.17297](https://arxiv.org/abs/2403.17297) [cs]. URL: <http://arxiv.org/abs/2403.17297> (besucht am 20.07.2024). Vorveröffentlichung.
- [12] J. Carbonell und J. Goldstein. „The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries“. In: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’98. New York, NY, USA: Association for Computing Machinery, 1. Aug. 1998, S. 335–336. ISBN: 978-1-58113-015-7. DOI: [10.1145/290941.291025](https://dl.acm.org/doi/10.1145/290941.291025). URL: <https://dl.acm.org/doi/10.1145/290941.291025> (besucht am 25.08.2024).
- [13] K. Cobbe et al. *Training Verifiers to Solve Math Word Problems*. 17. Nov. 2021. DOI: [10.48550/arXiv.2110.14168](https://arxiv.org/abs/2110.14168). arXiv: [2110.14168](https://arxiv.org/abs/2110.14168) [cs]. URL: <http://arxiv.org/abs/2110.14168> (besucht am 24.06.2024). Vorveröffentlichung.
- [14] T. Computer. *RedPajama: An Open Dataset for Training Large Language Models*. Okt. 2023. URL: <https://github.com/togethercomputer/RedPajama-Data> (besucht am 14.07.2024).
- [15] *Conceptual Guide - Vector Stores*. LangChain. URL: <https://python.langchain.com/v0.2/docs/concepts/#vector-stores> (besucht am 29.08.2024).
- [16] J. Dean und S. Ghemawat. „MapReduce: Simplified Data Processing on Large Clusters“. In: *Commun. ACM* 51.1 (1. Jan. 2008), S. 107–113. ISSN: 0001-0782. DOI: [10.1145/1327452.1327492](https://dl.acm.org/doi/10.1145/1327452.1327492). URL: <https://dl.acm.org/doi/10.1145/1327452.1327492> (besucht am 25.08.2024).

- [17] H. K. Dhalla. „A Performance Analysis of Native JSON Parsers in Java, Python, MS.NET Core, JavaScript, and PHP“. In: *2020 16th International Conference on Network and Service Management (CNSM)*. 2020 16th International Conference on Network and Service Management (CNSM). Nov. 2020, S. 1–5. DOI: [10.23919/CNSM50824.2020.9269101](https://doi.org/10.23919/CNSM50824.2020.9269101). URL: <https://ieeexplore.ieee.org/abstract/document/9269101> (besucht am 03.07.2024).
- [18] *Docker: Accelerated Container Application Development*. 10. Mai 2022. URL: <https://www.docker.com/> (besucht am 08.08.2024).
- [19] M. Douze et al. *The Faiss Library*. 2024. DOI: [10.48550/arXiv.2401.08281](https://doi.org/10.48550/arXiv.2401.08281). arXiv: [2401.08281 \[cs\]](https://arxiv.org/abs/2401.08281). URL: <http://arxiv.org/abs/2401.08281> (besucht am 29.08.2024). Vorveröffentlichung.
- [20] S. Elkins et al. *How Teachers Can Use Large Language Models and Bloom’s Taxonomy to Create Educational Quizzes*. 11. Jan. 2024. arXiv: [2401.05914 \[cs\]](https://arxiv.org/abs/2401.05914). URL: <http://arxiv.org/abs/2401.05914> (besucht am 31.05.2024). Vorveröffentlichung.
- [21] Z. Fei et al. *Query of CC: Unearthing Large Scale Domain-Specific Knowledge from Public Corpora*. 4. März 2024. DOI: [10.48550/arXiv.2401.14624](https://doi.org/10.48550/arXiv.2401.14624). arXiv: [2401.14624 \[cs\]](https://arxiv.org/abs/2401.14624). URL: <http://arxiv.org/abs/2401.14624> (besucht am 14.07.2024). Vorveröffentlichung.
- [22] F. Gröbblinghoff. „Lernergebnisse praktisch formulieren“. In: *nexus impulse für die Praxis* Nr. 2 (2. Auflage Juni 2015). ISSN: 2195-3619. URL: https://www.hrk-nexus.de/fileadmin/redaktion/hrk-nexus/07-Downloads/07-02-Publikationen/Lernergebnisse_praktisch_formulieren_01.pdf (besucht am 17.05.2024).
- [23] M. Grüttmüller. *Mathematik für Informatiker I: Grundlagen, Lineare Algebra Analysis – Differentialrechnung*. Vorlesungsskript, 17. Okt. 2023.
- [24] *Hello GPT-4o*. OpenAI. 13. Mai 2024. URL: <https://openai.com/index/hello-gpt-4o/> (besucht am 08.07.2024).
- [25] D. Hendrycks et al. *Measuring Massive Multitask Language Understanding*. 12. Jan. 2021. DOI: [10.48550/arXiv.2009.03300](https://doi.org/10.48550/arXiv.2009.03300). arXiv: [2009.03300 \[cs\]](https://arxiv.org/abs/2009.03300). URL: <http://arxiv.org/abs/2009.03300> (besucht am 09.07.2024). Vorveröffentlichung.
- [26] D. Hendrycks et al. *Measuring Mathematical Problem Solving With the MATH Dataset*. 8. Nov. 2021. DOI: [10.48550/arXiv.2103.03874](https://doi.org/10.48550/arXiv.2103.03874). arXiv: [2103.03874 \[cs\]](https://arxiv.org/abs/2103.03874). URL: <http://arxiv.org/abs/2103.03874> (besucht am 09.07.2024). Vorveröffentlichung.
- [27] J. D. Hunter. „Matplotlib: A 2D Graphics Environment“. In: *Computing in Science & Engineering* 9.3 (Mai 2007), S. 90–95. ISSN: 1558-366X. DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55). URL: <https://ieeexplore.ieee.org/document/4160265> (besucht am 25.08.2024).

- [28] S. Imani, L. Du und H. Shrivastava. *MathPrompter: Mathematical Reasoning Using Large Language Models*. 3. März 2023. DOI: [10.48550/arXiv.2303.05398](https://doi.org/10.48550/arXiv.2303.05398). arXiv: [2303.05398 \[cs\]](https://arxiv.org/abs/2303.05398). URL: <http://arxiv.org/abs/2303.05398> (besucht am 12.07.2024). Vorveröffentlichung.
- [29] *Json — JSON Encoder and Decoder*. Python documentation. URL: <https://docs.python.org/3/library/json.html> (besucht am 03.07.2024).
- [30] J. Kaplan et al. *Scaling Laws for Neural Language Models*. 22. Jan. 2020. DOI: [10.48550/arXiv.2001.08361](https://doi.org/10.48550/arXiv.2001.08361). arXiv: [2001.08361 \[cs, stat\]](https://arxiv.org/abs/2001.08361). URL: <http://arxiv.org/abs/2001.08361> (besucht am 25.08.2024). Vorveröffentlichung.
- [31] T. Kojima et al. „Large Language Models Are Zero-Shot Reasoners“. In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS ’22. Red Hook, NY, USA: Curran Associates Inc., 3. Apr. 2024, S. 22199–22213. ISBN: 978-1-71387-108-8.
- [32] G. Kurdi et al. „A Systematic Review of Automatic Question Generation for Educational Purposes“. In: *International Journal of Artificial Intelligence in Education* 30.1 (1. März 2020), S. 121–204. ISSN: 1560-4306. DOI: [10.1007/s40593-019-00186-y](https://doi.org/10.1007/s40593-019-00186-y). URL: <https://doi.org/10.1007/s40593-019-00186-y> (besucht am 18.04.2024).
- [33] U. Lämmel und J. Cleve. *Künstliche Intelligenz*. 4., aktualisierte Auflage. München: Hanser, 2012. ISBN: 978-3-446-42758-7.
- [34] *LangChain*. URL: <https://www.langchain.com/> (besucht am 29.07.2024).
- [35] *LangChain Expression Language (LCEL)*. LangChain. URL: https://python.langchain.com/v0.1/docs/expression_language/ (besucht am 17.07.2024).
- [36] H. Liu et al. „Few-Shot Parameter-Efficient Fine-Tuning Is Better and Cheaper than in-Context Learning“. In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS ’22. Red Hook, NY, USA: Curran Associates Inc., 3. Apr. 2024, S. 1950–1965. ISBN: 978-1-71387-108-8.
- [37] A. Martino, M. Iannelli und C. Truong. „Knowledge Injection to Counter Large Language Model (LLM) Hallucination“. In: *The Semantic Web: ESWC 2023 Satellite Events*. Hrsg. von C. Pesquita et al. Cham: Springer Nature Switzerland, 2023, S. 182–185. ISBN: 978-3-031-43458-7. DOI: [10.1007/978-3-031-43458-7_34](https://doi.org/10.1007/978-3-031-43458-7_34).
- [38] R. Meissner und A. Pögelt. *Llm-Task-Generator*. 15. Mai 2024. URL: https://gitlab.com/Tech4Comp/llm-task-generator/-/tree/deployment/ul-ap_new (besucht am 22.08.2024).

- [39] R. Meissner et al. *LLM-Generated Competence-Based E-Assessment Items for Higher Education Mathematics: Methodology and Evaluation*. 2023. Vorveröffentlichung.
- [40] S. Moore et al. „Assessing the Quality of Multiple-Choice Questions Using GPT-4 and Rule-Based Methods“. In: *Responsive and Sustainable Educational Futures*. Hrsg. von O. Viberg et al. Cham: Springer Nature Switzerland, 2023, S. 229–245. ISBN: 978-3-031-42682-7. DOI: [10.1007/978-3-031-42682-7_16](https://doi.org/10.1007/978-3-031-42682-7_16).
- [41] O. Musch. „Template Method“. In: *Design Patterns mit Java: Eine Einführung*. Hrsg. von O. Musch. Wiesbaden: Springer Fachmedien, 2021, S. 33–39. ISBN: 978-3-658-35492-3. DOI: [10.1007/978-3-658-35492-3_4](https://doi.org/10.1007/978-3-658-35492-3_4). URL: https://doi.org/10.1007/978-3-658-35492-3_4 (besucht am 24.08.2024).
- [42] H. Naveed et al. *A Comprehensive Overview of Large Language Models*. 9. Apr. 2024. arXiv: [2307.06435 \[cs\]](https://arxiv.org/abs/2307.06435). URL: <http://arxiv.org/abs/2307.06435> (besucht am 24.06.2024). Vorveröffentlichung.
- [43] A. T. Neuman et al. *Building an Intelligent Infrastructure for Scalable Mentoring in Higher Education*. 2024. Vorveröffentlichung.
- [44] H. A. Nguyen et al. „Evaluating ChatGPT’s Decimal Skills and Feedback Generation in a Digital Learning Game“. In: *Responsive and Sustainable Educational Futures*. Hrsg. von O. Viberg et al. Cham: Springer Nature Switzerland, 2023, S. 278–293. ISBN: 978-3-031-42682-7. DOI: [10.1007/978-3-031-42682-7_19](https://doi.org/10.1007/978-3-031-42682-7_19).
- [45] *OpenAI Developer Platform*. OpenAI Platform. URL: <https://platform.openai.com/docs/overview> (besucht am 08.07.2024).
- [46] K. Paster et al. *OpenWebMath: An Open Dataset of High-Quality Mathematical Web Text*. 10. Okt. 2023. DOI: [10.48550/arXiv.2310.06786](https://doi.org/10.48550/arXiv.2310.06786). arXiv: [2310.06786 \[cs\]](https://arxiv.org/abs/2310.06786). URL: <http://arxiv.org/abs/2310.06786> (besucht am 14.07.2024). Vorveröffentlichung.
- [47] F. Pedregosa et al. *Scikit-Learn: Machine Learning in Python*. 5. Juni 2018. DOI: [10.48550/arXiv.1201.0490](https://doi.org/10.48550/arXiv.1201.0490). arXiv: [1201.0490 \[cs\]](https://arxiv.org/abs/1201.0490). URL: <http://arxiv.org/abs/1201.0490> (besucht am 25.08.2024). Vorveröffentlichung.
- [48] G. Poore. *Text2qti*. 15. Juni 2024. URL: <https://github.com/gpoore/text2qti> (besucht am 03.07.2024).
- [49] *Prompt Engineering Guide*. URL: <https://www.promptingguide.ai/de> (besucht am 25.06.2024).

- [50] C. Qin et al. „Is ChatGPT a General-Purpose Natural Language Processing Task Solver?“ In: The 2023 Conference on Empirical Methods in Natural Language Processing. 1. Dez. 2023. URL: <https://openreview.net/forum?id=u03xn1C0s0> (besucht am 09.07.2024).
- [51] *Question & Test Interoperability®*. 1EdTech. URL: <https://www.1edtech.org/standards/qti> (besucht am 30.05.2024).
- [52] S. Raschka, J. Patterson und C. Nolet. „Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence“. In: *Information* 11.4 (4 Apr. 2020), S. 193. ISSN: 2078-2489. DOI: [10.3390/info11040193](https://www.mdpi.com/2078-2489/11/4/193). URL: <https://www.mdpi.com/2078-2489/11/4/193> (besucht am 24.08.2024).
- [53] I. Rohrbach. *Qti-Task-Generator*. URL: <https://gitlab.com/Tech4Comp/qti-task-generator> (besucht am 04.07.2024).
- [54] B. Rozière et al. „Code Llama: Open Foundation Models for Code“. In: (24. Aug. 2023).
- [55] N. Stojanovikj. *CSV2QTI*. 15. Sep. 2023. URL: <https://github.com/nenad/CSV2QTI> (besucht am 03.07.2024).
- [56] *Streamlit*. URL: <https://streamlit.io/> (besucht am 03.07.2024).
- [57] M. Tarrant et al. „The Frequency of Item Writing Flaws in Multiple-Choice Questions Used in High Stakes Nursing Assessments“. In: *Nurse Education Today*. Proceedings from the 1st Nurse Education International Conference 26.8 (1. Dez. 2006), S. 662–671. ISSN: 0260-6917. DOI: [10.1016/j.nedt.2006.07.006](https://www.sciencedirect.com/science/article/pii/S0260691706001067). URL: <https://www.sciencedirect.com/science/article/pii/S0260691706001067> (besucht am 30.06.2024).
- [58] *tech4compKI*. Wi-Ho - Wissenschafts- und Hochschulforschung. URL: <https://www.wihoforschung.de/wihoforschung/de/bmbf-projektfoerderung/foerderlinien/forschung-zur-digitalen-hochschulbildung/zweite-foerderlinie-zur-digitalen-hochschulbildung/zweite-foerderphase/tech4compki/tech4compki.html> (besucht am 20.05.2024).
- [59] A. Tharwat. „Classification Assessment Methods“. In: *Applied Computing and Informatics* 17.1 (1. Jan. 2020), S. 168–192. ISSN: 2210-8327. DOI: [10.1016/j.aci.2018.08.003](https://doi.org/10.1016/j.aci.2018.08.003). URL: <https://doi.org/10.1016/j.aci.2018.08.003> (besucht am 28.08.2024).

- [60] A. Thor, N. Pengel und H. Wollersheim. „Digitalisierte Hochschuldidaktik: Qualitätssicherung von Prüfungen Mit Dem E-Assessment-Literacy-Tool EAs.LiT“. In: *Bildungsräume 2017*. Bonn: Gesellschaft für Informatik, 2017. ISBN: 978-3-88579-667-1.
- [61] A. Vaswani et al. „Attention Is All You Need“. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., 4. Dez. 2017, S. 6000–6010. ISBN: 978-1-5108-6096-4.
- [62] *Verbundvorhaben*. tech4comp. URL: [https : / / tech4comp . de/](https://tech4comp.de/) (besucht am 20.05.2024).
- [63] J. Wei et al. „Chain-of-Thought Prompting Elicits Reasoning in Large Language Models“. In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS ’22. Red Hook, NY, USA: Curran Associates Inc., 3. Apr. 2024, S. 24824–24837. ISBN: 978-1-71387-108-8.
- [64] *Xml.Etree.ElementTree — The ElementTree XML API*. Python documentation. URL: <https://docs.python.org/3/library/xml.etree.elementtree.html> (besucht am 31.08.2024).
- [65] H. Ying et al. *InternLM-Math: Open Math Large Language Models Toward Verifiable Reasoning*. 9. Feb. 2024. arXiv: [2402.06332](https://arxiv.org/abs/2402.06332) [cs]. URL: <http://arxiv.org/abs/2402.06332> (besucht am 28.04.2024). Vorveröffentlichung.
- [66] Y. Zhang et al. *Siren’s Song in the AI Ocean: A Survey on Hallucination in Large Language Models*. 24. Sep. 2023. DOI: [10.48550/arXiv.2309.01219](https://doi.org/10.48550/arXiv.2309.01219). arXiv: [2309.01219](https://arxiv.org/abs/2309.01219) [cs]. URL: <http://arxiv.org/abs/2309.01219> (besucht am 09.07.2024). Vorveröffentlichung.

A.1. LLM-TESTS

In diesem Anhang sind die verwendeten Prompts vom LLM-Test sowie die Ausgaben angegeben.

Prompts

In diesem Abschnitt sind die verwendeten Prompts für den Test der LLMs gegeben. Die blau gefärbten Textabschnitte wurden für den kürzeren Prompt entfernt.

Codeausschnitt 4: Prompt für die Aufgabengenerierung beim LLM-Test

```
1 You're an AI whose purpose is to create assessment items (tasks and corresponding solutions).
2
3 ### Rules ###
4 * Think step by step about how to create the requested item correctly.
5 * Always respond with in german.
6 * If mathematical annotations and formulae are included, format them according to the
   ↪ LaTeX standard.
7
8 ### Item Construction Guidelines ###
9 * The task has to be based on the given topic information but not reproduce these verbatim.
10 * The item shall focus on one selected and specific aspect in both the task and the answer options.
11 * The task must be answerable even without knowledge of the answer options.
12 * Make sure that answer options do not repeat words from the task 1:1.
13 * Formulate only relevant and precise information in the items. Avoid misleading information and
   formulations. Also avoid ambiguous task statements and negative formulations for the construct being
   measured.
14 * If you use abbreviations, use only abbreviations that are generally known in order to minimize reading
   time.
15 * Avoid linguistic or formal solution hints (also known as clues, cues, or supportive texts), in the task.
   Use simple, clear, and unambiguous formulations.
16 * Ensure that not only the correct answers are well-formulated, but also focus on crafting plausible
   distractors. Formulate the distractors, or false answer options, in a way that they are not too similar
   to the correct answer, yet still share some thematic relevance.
17 * Make sure that the feedback does not repeat words from the answer.
18
19 ### Answer Format ###
20 * The task and solution should always be in JSON!
21 * The feedback for wrong answers should include clues how to reach the correct solution
   ↪ based on the mistake. If you can't prompt them to read more about the topic.
22
23 ### Available Knowledge ###
24 Use the following information on the topic to create the item:
25 Ein Grenzwert beschreibt das Verhalten einer Funktion  $f(x)$ , wenn sich  $x$  einem
   ↪ bestimmten Wert  $a$  nähert.
26 Formal gesagt, der Grenzwert von  $f(x)$  für  $x$  gegen  $a$  ist  $L$ , wenn für jede
   ↪ noch so kleine positive Zahl  $\epsilon$  eine positive Zahl  $\delta$ 
```

↪ existiert, so dass für alle x innerhalb des Intervalls $(a - \delta, a + \delta)$ (außer möglicherweise bei a selbst) die Ungleichung $|f(x) - L| < \epsilon$ gilt.

27 Ein Beispiel zur Veranschaulichung: Betrachten wir die Funktion $f(x) = 2x$.

28 Wir wollen den Grenzwert dieser Funktion für x gegen 3 bestimmen.

29 Intuitiv sehen wir, dass wenn x sich 3 nähert, $f(x)$ sich 6 nähert.

30 Formal bedeutet dies, dass für jede positive Zahl ϵ eine positive Zahl δ existiert, so dass $|2x - 6| < \epsilon$, wenn $|x - 3| < \delta$.

31 Zusätzlich zur Definition des Grenzwerts ist es wichtig, die Stetigkeit einer Funktion zu verstehen.

32 Eine Funktion ist stetig an einem Punkt a , wenn der Grenzwert der Funktion an diesem Punkt gleich dem Funktionswert an diesem Punkt ist, also $\lim_{x \rightarrow a} f(x) = f(a)$.

33

34 Use the following information of the cognitive process and knowledge dimension to create the item:

35 By addressing items of the cognitive process dimension 'remember', assessment attendees are expected to demonstrate their ability to access relevant knowledge stored in long-term memory, which entails recognizing or recalling previously learned information.

36 You should choose one of the following two categories:

37 1. Recognition

38 Example: Recognizing data of historical events or examples of concepts or definitions.

39 2. Recall

40 Example: Recalling data of historical events or examples of concepts or definitions.

41 ---

42 By addressing items of the knowledge dimension 'factual', assessment attendees are expected to demonstrate that they possess foundational knowledge to be familiar with the topics covered in the mentioned course or to solve related problems.

43 You should choose one of the following two categories:

44 1. Understanding of concepts and technical terminology, as well as knowledge of the components and specific details of these concepts.

45 or

46 2. Knowledge of the components and specific details of these terms and concepts.

47

48 ### Task ###

49 Create an easy single-choice assessment item for the definition of 'Quantoren' that represents the cognitive process dimension 'remember', the knowledge dimension 'factual' in JSON:

50 {

51 "question_type": "single-choice",

52 "question": {

53 "title": "Allquantor (SC)",

54 "question_text": "Wählen Sie den Allquantor aus.",

55 "options": [

56 {"text": " $\forall$ ", "is_correct": true, "feedback": "Sieht gut aus!"},

```
57 {"text": "$$\exists$", "is_correct": false, "feedback": "Das ist der falsche
    ↳ Quantor. Damit die Vertauschung sich nicht festsetzt, gucken Sie sich noch
    ↳ einmal die Quantoren an."},
58 {"text": "$$A$$", "is_correct": false, "feedback": "Das ist ein normales A in einer
    ↳ Mathematik-Umgebung. Gucken Sie sich die Quantoren noch einmal an."}
59 ]
60 }
61 }
62
63 Create an easy single-choice assessment item for the definition of 'Grenzwert [Reelle
    ↳ Funktion]' that represents the cognitive process dimension 'remember', the
    ↳ knowledge dimension 'factual' in JSON:
```

Codeausschnitt 5: Prompt für die Evaluation beim LLM-Test

```
1 The output should be formatted as a JSON object!
2
3 Determine all correct answers of the task and explain why.
4
5 Task:
6 Welche Aussage beschreibt eine streng monoton steigende Funktion korrekt? Für zwei
    ↳ verschiedene Werte x und y im Definitionsbereich gilt:
7 A) Wenn x kleiner ist als y, dann ist der Funktionswert für x gleich dem Funktionswert
    ↳ für y.
8 B) Wenn x kleiner ist als y, dann ist der Funktionswert für x kleiner als der
    ↳ Funktionswert für y.
9 C) Wenn x größer ist als y, dann ist der Funktionswert für x kleiner als der
    ↳ Funktionswert für y.
10 Answer:
11 { "reason": "A strictly monotonically increasing function is defined such that for any
    ↳ two different values x and y in the domain, if x is lower than y, then the function
    ↳ value at x is lower than the function value at y. This is correctly described by
    ↳ statement B. Statement A describes a constant function and statement C describes a
    ↳ strictly monotonically decreasing function. Therefore the answer is ['B']",
    ↳ "decision": ["B"] }
12 Task:
13 Wählen Sie den Allquantor aus.
14 1) $$\forall$$$
15 2) $$\exists$$$
16 3) $$A$$$
17 Answer:
```

Ausgaben

In diesem Abschnitt sind alle Ausgaben der Sprachmodelle zu den Test-Prompts angegeben.

GPT-3.5 Turbo

Codeausschnitt 6: Ausgabe der Aufgabengenerierung von GPT-3.5 Turbo

```
1 {
2   "question_type": "single-choice",
3   "question": {
4     "title": "Grenzwert (SC)",
5     "question_text": "Welcher Begriff beschreibt das Verhalten einer Funktion, wenn sich
      ↪ x einem bestimmten Wert a nähert?",
6     "options": [
7       {"text": "Integral", "is_correct": false, "feedback": "Das ist nicht korrekt. Der
      ↪ Begriff 'Integral' bezieht sich auf die Fläche unter einer Funktion."},
8       {"text": "Ableitung", "is_correct": false, "feedback": "Leider falsch. Die
      ↪ Ableitung einer Funktion beschreibt deren Steigung an einem bestimmten
      ↪ Punkt."},
9       {"text": "Grenzwert", "is_correct": true, "feedback": "Richtig! Der Grenzwert
      ↪ beschreibt das Verhalten einer Funktion, wenn sich x einem bestimmten Wert a
      ↪ nähert."},
10      {"text": "Differential", "is_correct": false, "feedback": "Das ist nicht die
      ↪ richtige Antwort. 'Differential' bezieht sich auf infinitesimale Änderungen
      ↪ in einer Funktion."}
11    ]
12  }
13 }
```

Codeausschnitt 7: Ausgabe der Aufgabengenerierung (gekürzt) von GPT-3.5 Turbo

```
1 {
2   "question_type": "single-choice",
3   "question": {
4     "title": "Grenzwert (SC)",
5     "question_text": "Welcher Begriff beschreibt das Verhalten einer Funktion, wenn sich
      ↪ x einem bestimmten Wert a nähert?",
6     "options": [
7       {"text": "Integral", "is_correct": false, "feedback": "Das ist nicht korrekt.
      ↪ Denken Sie an das Verhalten einer Funktion bei Annäherung an einen bestimmten
      ↪ Wert."},
8       {"text": "Ableitung", "is_correct": false, "feedback": "Das ist nicht korrekt.
      ↪ Denken Sie an das Verhalten einer Funktion bei Annäherung an einen bestimmten
      ↪ Wert."},
```

```
9      {"text": "Grenzwert", "is_correct": true, "feedback": "Richtig! Der Grenzwert
      ↳ beschreibt das Verhalten einer Funktion, wenn sich x einem bestimmten Wert a
      ↳ nähert."}
10    ]
11  }
12 }
```

Codeausschnitt 8: Ausgabe der Evaluation von GPT-3.5 Turbo

```
1 {
2   "reason": "The Allquantor, denoted as  $\forall$ , is used to express that a statement
      ↳ is true for all elements in a set or domain. The Existenzquantor, denoted as
      ↳  $\exists$ , is used to express that a statement is true for at least one element
      ↳ in a set or domain. Option 1)  $\forall$  is the correct choice for the
      ↳ Allquantor.",
3   "decision": [ $\forall$ ]
4 }
```

GPT-4o

Codeausschnitt 9: Ausgabe der Aufgabengenerierung von GPT-4o

```
1 {
2   "question_type": "single-choice",
3   "question": {
4     "title": "Grenzwert (SC)",
5     "question_text": "Wählen Sie die korrekte Definition eines Grenzwerts einer reellen
      ↳ Funktion aus.",
6     "options": [
7       {"text": "Der Grenzwert einer Funktion  $f(x)$  für  $x$  gegen  $a$  ist  $L$ ,
          ↳ wenn für jede positive Zahl  $\epsilon$  eine positive Zahl  $\delta$ 
          ↳ existiert, so dass für alle  $x$  innerhalb des Intervalls  $(a - \delta, a + \delta)$ 
          ↳ die Ungleichung  $|f(x) - L| < \epsilon$  gilt.", "is_correct":
          ↳ true, "feedback": "Richtig! Das ist die korrekte Definition eines
          ↳ Grenzwerts."},
8       {"text": "Der Grenzwert einer Funktion  $f(x)$  für  $x$  gegen  $a$  ist  $L$ ,
          ↳ wenn  $f(x)$  für alle  $x$  innerhalb des Intervalls  $(a - \delta, a + \delta)$ 
          ↳ gleich  $L$  ist.", "is_correct": false, "feedback": "Das ist
          ↳ nicht korrekt. Überprüfen Sie die Definition des Grenzwerts und achten Sie
          ↳ auf die Bedingung mit  $\epsilon$  und  $\delta$ ."},
9       {"text": "Der Grenzwert einer Funktion  $f(x)$  für  $x$  gegen  $a$  ist  $L$ ,
          ↳ wenn  $f(x)$  für alle  $x$  gleich  $L$  ist.", "is_correct": false,
```

```
    ↪ "feedback": "Das ist nicht korrekt. Der Grenzwert bezieht sich auf das
    ↪ Verhalten der Funktion, wenn  $x$  sich  $a$  nähert, nicht auf den
    ↪ Funktionswert für alle  $x$ ."}
10   ]
11   }
12 }
```

Codeausschnitt 10: Ausgabe der Aufgabengenerierung (gekürzt) von GPT-4o

```
1 {
2   "question_type": "single-choice",
3   "question": {
4     "title": "Grenzwert (SC)",
5     "question_text": "Wählen Sie die korrekte Definition eines Grenzwerts einer reellen
    ↪ Funktion aus.",
6     "options": [
7       {"text": "Der Grenzwert einer Funktion  $f(x)$  für  $x$  gegen  $a$  ist  $L$ ,
    ↪ wenn für jede noch so kleine positive Zahl  $\epsilon$  eine positive Zahl
    ↪  $\delta$  existiert, so dass für alle  $x$  innerhalb des Intervalls  $(a - \delta, a + \delta)$  (außer  $a$  selbst) die
    ↪ Ungleichung  $|f(x) - L| < \epsilon$  gilt.", "is_correct": true,
    ↪ "feedback": "Richtig! Das ist die korrekte Definition eines Grenzwerts."},
8       {"text": "Der Grenzwert einer Funktion  $f(x)$  für  $x$  gegen  $a$  ist  $L$ ,
    ↪ wenn  $f(x)$  für alle  $x$  innerhalb des Intervalls  $(a - \delta, a + \delta)$  gleich  $L$  ist.", "is_correct": false, "feedback": "Das ist
    ↪ nicht korrekt. Überprüfen Sie die Definition des Grenzwerts noch einmal."},
9       {"text": "Der Grenzwert einer Funktion  $f(x)$  für  $x$  gegen  $a$  ist  $L$ ,
    ↪ wenn  $f(x)$  für alle  $x$  innerhalb des Intervalls  $(a - \epsilon, a + \epsilon)$  gleich  $L$  ist.", "is_correct": false, "feedback": "Das ist
    ↪ nicht korrekt. Beachten Sie, dass  $\epsilon$  und  $\delta$ 
    ↪ unterschiedliche Rollen in der Definition spielen."}
10    ]
11  }
12 }
```

Codeausschnitt 11: Ausgabe der Evaluation von GPT-4o

```
1 {
2   "reason": "The universal quantifier, also known as the 'for all' quantifier, is
    ↪ represented by the symbol  $\forall$ . This symbol is used in mathematical logic to
    ↪ indicate that a statement applies to all elements of a given set. The symbol  $\exists$ 
    ↪ represents the existential quantifier, which indicates that there exists at least
    ↪ one element in the set for which the statement is true. The symbol  $\neg$  is not a
    ↪ quantifier. Therefore, the correct answer is ['1'].",
3   "decision": ["1"]
4 }
```

Llemma 7B

Mit der Implementierung, mit der die Tests durchgeführt wurden, war von Llemma 7B der Prompt in der Antwort angegeben und terminierte nicht nach den hier gezeigten Antworten.

Codeausschnitt 12: Ausgabe der Aufgabengenerierung von Llemma 7B

```
1 {
2   "question_type": "single-choice",
3   "question": {
4     "title": "Grenzwert (SC)",
5     "question_text": "Wählen Sie den Grenzwert aus.",
6     "options": [
7       {"text": "$$\lim_{x \to a} f(x) = L$$", "is_correct": true, "feedback": "Sieht
          ↳ gut aus!"},
8       {"text": "$$\lim_{x \to a} f(x) = a$$", "is_correct": false, "feedback": "Das ist
          ↳ der falsche Grenzwert. Gucken Sie sich die Definition noch einmal an."},
9       {"text": "$$\lim_{x \to a} f(x) = f(a)$$", "is_correct": false, "feedback": "Das
          ↳ ist der falsche Grenzwert. Gucken Sie sich die Definition noch einmal an."}
10    ]
11  }
12 }
```

Codeausschnitt 13: Ausgabe der Aufgabengenerierung (gekürzt) von Llemma 7B

```
1 {
2   "question_type": "single-choice",
3   "question": {
4     "title": "Grenzwert (SC)",
5     "question_text": "Wählen Sie den Grenzwert aus.",
6     "options": [
7       {"text": "$$\lim_{x \to a} f(x)$$", "is_correct": true, "feedback": "Sieht gut
          ↳ aus!"},
8       {"text": "$$\lim_{x \to a} f(x) = f(a)$$", "is_correct": false, "feedback": "Das
          ↳ ist der falsche Grenzwert. Gucken Sie sich die Definition noch einmal an."},
9       {"text": "$$\lim_{x \to a} f(x) = L$$", "is_correct": false, "feedback": "Das ist
          ↳ der falsche Grenzwert. Gucken Sie sich die Definition noch einmal an."}
10    ]
11  }
12 }
```

Codeausschnitt 14: Ausgabe der Evaluation von Llemma 7B

```
1 {
2   "reason": "The universal quantifier is used to express that a statement is true for all
   ↪ elements in a set. Therefore the answer is ['1']",
3   "decision": ["1"]
4 }
```

Llemma 34B

Mit der Implementierung, mit der die Tests durchgeführt wurden, war von Llemma 34B der Prompt in der Antwort angegeben und terminierte nicht nach den hier gezeigten Antworten.

Codeausschnitt 15: Ausgabe der Aufgabengenerierung (gekürzt) von Llemma 34B

```
1 {
2   "question_type": "single-choice",
3   "question": {
4     "title": "Grenzwert (SC)",
5     "question_text": "Wählen Sie die richtige Definition des Grenzwerts aus.",
6     "options": [
7       {"text": "Der Grenzwert von  $f(x)$  für  $x$  gegen  $a$  ist  $L$ , wenn für jede
        ↪ noch so kleine positive Zahl  $\epsilon$  eine positive Zahl  $\delta$ 
        ↪ existiert, so dass für alle  $x$  innerhalb des Intervalls  $(a - \delta, a +$ 
        ↪  $\delta)$  (außer möglicherweise bei  $a$  selbst) die Ungleichung  $|f(x) -$ 
        ↪  $L| < \epsilon$  gilt.", "is_correct": true, "feedback": "Sieht gut aus!"},
8       {"text": "Eine Funktion ist stetig an einem Punkt  $a$ , wenn der Grenzwert der
        ↪ Funktion an diesem Punkt gleich dem Funktionswert an diesem Punkt ist, also
        ↪  $\lim_{x \rightarrow a} f(x) = f(a)$ .", "is_correct": false, "feedback": "Das ist
        ↪ die Definition der Stetigkeit. Gucken Sie sich die Definition des Grenzwerts
        ↪ noch einmal an."},
9       {"text": "Ein Grenzwert beschreibt das Verhalten einer Funktion  $f(x)$ , wenn sich
        ↪  $x$  einem bestimmten Wert  $a$  nähert.", "is_correct": false, "feedback":
        ↪ "Das ist die Definition des Grenzwerts. Gucken Sie sich die Definition der
        ↪ Stetigkeit noch einmal an."}
10    ]
11  }
12 }
```

Codeausschnitt 16: Ausgabe der Evaluation von Llemma 34B

```
1 {
2   "reason": "The Allquantor is the symbol for the Allquantor. It is used to express that
   ↪ a statement is true for all elements of a set. The answer is ['1']",
3   "decision": ["1"]
4 }
```

InternLM-Math 7B

Codeausschnitt 17: Ausgabe der Aufgabengenerierung von InternLM-Math 7B

```
1 {
2   "question_type": "single-choice",
3   "question": {
4     "title": "Reelle Funktion",
5     "question_text": "Wählen Sie die Definition der Grenzwert einer Reellen Funktion
        ↪ aus.",
6     "options": [
7       {"text": "Eine Grenzwert beschreibt das Verhalten einer Funktion  $f(x)$ , wenn
            ↪ sich  $x$  einem bestimmten Wert  $a$  nähert. Formal gesagt, der Grenzwert
            ↪ von  $f(x)$  für  $x$  gegen  $a$  ist  $L$ , wenn für jede noch so kleine
            ↪ positive Zahl  $\epsilon$  eine positive Zahl  $\delta$  existiert, so dass
            ↪ für alle  $x$  innerhalb des Intervalls  $(a - \delta, a + \delta)$  (außer
            ↪ möglicherweise bei  $a$  selbst) die Ungleichung  $|f(x) - L| < \epsilon$ 
            ↪ gilt.", "is_correct": true, "feedback": "Gute Arbeit! Diese Antwort hat sich
            ↪ sehr gut auf die Frage zu repräsentieren. Es wurde die wichtigste
            ↪ Informationen über die Grenzwert-Definition und die wichtigsten Begriffe wie
            ↪  $\epsilon$  und  $\delta$  verpackt. Selten man sieht so genaue und
            ↪ verständliche Antworten wie diese."},
8       {"text": "Eine Reelle Funktion ist stetig an einem Punkt  $a$ , wenn der Grenzwert
            ↪ der Funktion an diesem Punkt gleich dem Funktionswert an diesem Punkt ist,
            ↪ also  $\lim_{x \rightarrow a} f(x) = f(a)$ .", "is_correct": false, "feedback":
            ↪ "Das sieht nicht sehr passend aus. Es fehlen wichtige Informationen und die
            ↪ Antwort versetzt sich nicht sehr deutlich gegen die anderen Antworten. Es
            ↪ wäre wichtig weitere Informationen wie die Formalisierung der
            ↪ Grenzwert-Definition und Begriffe wie  $\epsilon$  und  $\delta$  in die
            ↪ Antwort zu bekommen."},
9       {"text": "Eine Reelle Funktion  $f(x)$  an einem Punkt  $a$  hat einen Grenzwert
            ↪  $L$ , wenn für jede noch so kleine positive Zahl  $\epsilon$  eine positive
            ↪ Zahl  $\delta$  existiert, so dass für alle  $x$  innerhalb des Intervalls
            ↪  $(a - \delta, a + \delta)$  (außer möglicherweise bei  $a$  selbst) die
            ↪ Ungleichung  $|f(x) - L| < \epsilon$  gilt.", "is_correct": false,
            ↪ "feedback": "Das sieht nicht sehr passend aus. Es fehlen wichtige
            ↪ Informationen und die Antwort versetzt sich nicht sehr deutlich gegen die
            ↪ anderen Antworten. Es wäre wichtig weitere Informationen wie die
            ↪ Formalisierung der Grenzwert-Definition und Begriffe wie  $\epsilon$  und
            ↪  $\delta$  in die Antwort zu bekommen."},
10      {"text": "Wir wollen den Grenzwert dieser Funktion für  $x$  gegen 3 bestimmen.
            ↪ Intuitiv sehen wir, dass wenn  $x$  sich 3 nähert,  $f(x)$  sich 6 nähert.
            ↪ Formal bedeutet dies, dass für jede positive Zahl  $\epsilon$  eine positive
            ↪ Zahl  $\delta$  existiert, so dass  $|2x - 6| < \epsilon$ , wenn  $|x - 3|$ 
            ↪  $< \delta$ .", "is_correct": false, "feedback": "Das sieht nicht sehr passend
            ↪ aus. Es fehlen wichtige Informationen und die Antwort versetzt sich nicht
```

```
    ↪ sehr deutlich gegen die anderen Antworten. Es wäre wichtig viele
    ↪ Informationen wie die Formalisierung der Grenzwert-Definition und Begriffe
    ↪ wie  $\epsilon$  und  $\delta$  in die Antwort zu bekommen."}
11 ]
12 }
```

Codeausschnitt 18: Ausgabe der Aufgabengenerierung (gekürzt) von InternLM-Math 7B

```
1 {
2   "question_type": "single-choice",
3   "question": {
4     "title": "Grenzwert der Reellen Funktion (SC)",
5     "question_text": "Wählen Sie die Definition des Grenzwerts einer Reellen Funktion für
        ↪  $x$  gegen  $a$  aus.",
6     "options": [
7       {"text": "Eine Funktion  $f(x)$  hat Grenzwert  $L$  für  $x$  gegen  $a$ , wenn
            ↪ für jeden nochmal kleinen positive Abstand  $\epsilon$  eine positive Abstand
            ↪  $\delta$  existiert, so dass für alle  $x$  innerhalb des Intervalls  $(a -$ 
            ↪  $\delta, a + \delta)$  (außer möglicherweise bei  $a$  selbst) die Ungleichung
            ↪  $|f(x) - L| < \epsilon$  gilt.", "is_correct": true, "feedback": "Sind Sie
            ↪ sich sicher?"}},
8       {"text": "Eine Funktion  $f(x)$  hat Grenzwert  $L$  für  $x$  gegen  $a$ , wenn
            ↪ für jeden nochmal kleinen positive Abstand  $\epsilon$  eine positive Abstand
            ↪  $\delta$  existiert, so dass für alle  $x$  innerhalb des Intervalls  $(a -$ 
            ↪  $\delta, a + \delta)$  (außerhalb von  $a$  selbst) die Ungleichung  $|f(x) -$ 
            ↪  $L| < \epsilon$  erfüllt ist.", "is_correct": false, "feedback": "Die
            ↪ Ungleichung gilt nicht nur für die Eingrenzung, sondern auch für das Punkt
            ↪  $a$ ."}]
9     ]
10   }
11 }
```

Codeausschnitt 19: Ausgabe der Evaluation von InternLM-Math 7B

```
1 {
2   "answer": "1",
3   "reason": "The Allquantor, symbolized by forall, states that a certain property is true
        ↪ for all elements under consideration. In this case, it would be \"for all x, y in
        ↪ the domain, if x is less than y, then the function value at x is less than the
        ↪ function value at y.\" This statement describes a strictly monotonically
        ↪ increasing function, which is what we are looking for. Therefore, the correct
        ↪ answer is ['1'].\"
4 }
```

A.2. BEISPIELE GENERierter ÜBUNGSAUFGABEN

Single-Choice

In diesem Abschnitt ist die erstellte *Single-Choice*-Aufgabe zum mathematischen Konzept *Maximum [Reelle Funktion]* sowie deren Evaluationen dargestellt.

ONYX

Abb. 11: Screenshot der Single-Choice-Aufgabe in ONYX

Maximum einer reellen Funktion (SC) Punkte: 1 Keine Antwort

Was ist ein Maximum einer reellen Funktion?

- ☐ Ein Punkt, an dem die Funktion nicht definiert ist.
- ☐ Ein Punkt, an dem die Ableitung der Funktion null ist.
- ☐ Der kleinste Funktionswert, den die Funktion erreicht.
- ☐ Der größte Funktionswert, den die Funktion erreicht.

JSON

Codeausschnitt 20: Darstellung der Single-Choice-Aufgabe in JSON

```
1 {  
2   "question_type": "single-choice",  
3   "question": {  
4     "title": "Maximum einer reellen Funktion (SC)",  
5     "question_text": "Was ist ein Maximum einer reellen Funktion?",  
6     "options": [  
7       {"text": "Der größte Funktionswert, den die Funktion erreicht.", "is_correct":  
         ↳ true, "feedback": "Richtig! Ein Maximum ist der größte Funktionswert, den die  
         ↳ Funktion erreicht."},
```

```
8      {"text": "Der kleinste Funktionswert, den die Funktion erreicht.", "is_correct":  
      ↪ false, "feedback": "Das ist ein Minimum. Überlegen Sie noch einmal, was ein  
      ↪ Maximum ist."},  
9      {"text": "Ein Punkt, an dem die Funktion nicht definiert ist.", "is_correct":  
      ↪ false, "feedback": "Das ist eine Definitionslücke. Überlegen Sie noch einmal,  
      ↪ was ein Maximum ist."},  
10     {"text": "Ein Punkt, an dem die Ableitung der Funktion null ist.", "is_correct":  
      ↪ false, "feedback": "Das beschreibt eine kritische Stelle, aber nicht  
      ↪ unbedingt ein Maximum. Überlegen Sie noch einmal, was ein Maximum ist."}  
11 ]  
12 }  
13 }
```

QTI-Format

Codeausschnitt 21: Single-Choice-Aufgabe im QTI-Format in verkürzter Form

```
1 <assessmentItem xmlns="http://www.imsglobal.org/xsd/imsqti_v2p1" xmlns:xsi=...  
  ↪ xsi:schemaLocation=... identifier="8c6ec218-dae9-4b80-86af-e60915c37c95"  
  ↪ title="Maximum einer reellen Funktion (SC)" adaptive="false" timeDependent="false">  
2   <responseDeclaration identifier="CORRECT_RESPONSE" cardinality="single"  
    ↪ baseType="identifier">  
3     <correctResponse>  
4       <value>ID_1</value>  
5     </correctResponse>  
6   </responseDeclaration>  
7   <outcomeDeclaration identifier="SCORE" cardinality="single" baseType="float">  
8     ...  
9   </outcomeDeclaration>  
10  <outcomeDeclaration identifier="MAXSCORE" cardinality="single" baseType="float">  
11    ...  
12  </outcomeDeclaration>  
13  <outcomeDeclaration identifier="MINSORE" cardinality="single" baseType="float">  
14    ...  
15  </outcomeDeclaration>  
16  <outcomeDeclaration identifier="FEEDBACKBASIC" cardinality="single"  
    ↪ baseType="identifier">  
17    ...  
18  </outcomeDeclaration>  
19  <outcomeDeclaration identifier="FEEDBACKMODAL" cardinality="multiple"  
    ↪ baseType="identifier">  
20    ...  
21  </outcomeDeclaration>  
22  <itemBody>  
23    <p>
```

```
24     <span style="font-size: 12pt">Was ist ein Maximum einer reellen Funktion?</span>
25 </p>
26 <choiceInteraction responseIdentifier="CORRECT_RESPONSE" shuffle="true"
    ↪ maxChoices="1">
27     <simpleChoice identifier="ID_1">
28         <p>
29             <span style="font-size: 12pt">Der groesste Funktionswert, den die Funktion
    ↪ erreicht.</span>
30         </p>
31     </simpleChoice>
32     <simpleChoice identifier="ID_2">
33         <p>
34             <span style="font-size: 12pt">Der kleinste Funktionswert, den die Funktion
    ↪ erreicht.</span>
35         </p>
36     </simpleChoice>
37     <simpleChoice identifier="ID_3">
38         <p>
39             <span style="font-size: 12pt">Ein Punkt, an dem die Funktion nicht definiert
    ↪ ist.</span>
40         </p>
41     </simpleChoice>
42     <simpleChoice identifier="ID_4">
43         <p>
44             <span style="font-size: 12pt">Ein Punkt, an dem die Ableitung der Funktion null
    ↪ ist.</span>
45         </p>
46     </simpleChoice>
47 </choiceInteraction>
48 </itemBody>
49 <responseProcessing>
50     <responseCondition>
51         <responseIf>
52             <isNull>
53                 <variable identifier="CORRECT_RESPONSE"/>
54             </isNull>
55         </responseIf>
56         <responseElseIf>
57             <match>
58                 <variable identifier="CORRECT_RESPONSE"/>
59                 <correct identifier="CORRECT_RESPONSE"/>
60             </match>
61             <setOutcomeValue identifier="SCORE">
62                 <sum>
63                     <variable identifier="SCORE"/>
```

```
64         <variable identifier="MAXSCORE"/>
65     </sum>
66 </setOutcomeValue>
67 </responseElseIf>
68 </responseCondition>
69 <responseCondition>
70     <responseIf>
71         <gt>
72             <variable identifier="SCORE"/>
73             <variable identifier="MAXSCORE"/>
74         </gt>
75         <setOutcomeValue identifier="SCORE">
76             <variable identifier="MAXSCORE"/>
77         </setOutcomeValue>
78     </responseIf>
79 </responseCondition>
80 <responseCondition>
81     <responseIf>
82         <lt>
83             <variable identifier="SCORE"/>
84             <variable identifier="MINSORE"/>
85         </lt>
86         <setOutcomeValue identifier="SCORE">
87             <variable identifier="MINSORE"/>
88         </setOutcomeValue>
89     </responseIf>
90 </responseCondition>
91 <responseCondition>
92     <responseIf>
93         <and>
94             <isNull>
95                 <variable identifier="CORRECT_RESPONSE"/>
96             </isNull>
97         </and>
98         <setOutcomeValue identifier="FEEDBACKBASIC">
99             <baseValue baseType="identifier">empty</baseValue>
100         </setOutcomeValue>
101     </responseIf>
102 <responseElseIf>
103     <lt>
104         <variable identifier="SCORE"/>
105         <variable identifier="MAXSCORE"/>
106     </lt>
107     <setOutcomeValue identifier="FEEDBACKBASIC">
108         <baseValue baseType="identifier">incorrect</baseValue>
```

```
109         </setOutcomeValue>
110     </responseElseIf>
111     <responseElse>
112         <setOutcomeValue identifier="FEEDBACKBASIC">
113             <baseValue baseType="identifier">correct</baseValue>
114         </setOutcomeValue>
115     </responseElse>
116 </responseCondition>
117 <responseCondition>
118     ...
119 </responseCondition>
120 <responseCondition>
121     ...
122 </responseCondition>
123 <responseCondition>
124     ...
125 </responseCondition>
126 <responseCondition>
127     ...
128 </responseCondition>
129 </responseProcessing>
130 <modalFeedback identifier="Feedback161459712191804372097634539569822335127"
    ↪ outcomeIdentifier="FEEDBACKMODAL" showHide="show" title="Richtig!">
131 <p>
132     <span style="font-size: 12pt">Richtig! Ein Maximum ist der groesste Funktionswert,
    ↪ den die Funktion erreicht.</span>
133 </p>
134 </modalFeedback>
135 <modalFeedback identifier="Feedback182858044728282060759685983480232837935"
    ↪ outcomeIdentifier="FEEDBACKMODAL" showHide="show" title="Noch nicht ganz">
136 <p>
137     <span style="font-size: 12pt">Das ist ein Minimum. Ueberlegen Sie noch einmal, was
    ↪ ein Maximum ist.</span>
138 </p>
139 </modalFeedback>
140 <modalFeedback identifier="Feedback194131578556864406246019036734359127296"
    ↪ outcomeIdentifier="FEEDBACKMODAL" showHide="show" title="Noch nicht ganz">
141 <p>
142     <span style="font-size: 12pt">Das ist eine Definitionsluecke. Ueberlegen Sie noch
    ↪ einmal, was ein Maximum ist.</span>
143 </p>
144 </modalFeedback>
145 <modalFeedback identifier="Feedback238502922375576141787300424911320084118"
    ↪ outcomeIdentifier="FEEDBACKMODAL" showHide="show" title="Noch nicht ganz">
146 <p>
```

```

147     <span style="font-size: 12pt">Das beschreibt eine kritische Stelle, aber nicht
      ↪ unbedingt ein Maximum. Ueberlegen Sie noch einmal, was ein Maximum ist.</span>
148   </p>
149 </modalFeedback>
150 </assessmentItem>

```

Evaluationen

Tabelle 6: Evaluationen der Single-Choice-Aufgabe

	Akzeptabel?	
	Expertin	Maschinell
Kognitive Prozessdimension	ja	ja
Wissensdimension	ja	ja
Aufführbarkeit	ja	ja
Aufgabenstellung	ja	ja
Auswertung	ja	ja
Feedbackzweige	ja	ja
Kontext	ja	ja
Bewertung	ja	ja

Expertin Alles perfekt! Schöne Feedback-Zweige!

Maschinell

Kognitive Prozessdimension Erinnern

The task requires the student to recall the definition of a maximum of a real function. This involves retrieving previously learned information from memory. Therefore, the task falls under the 'remember' cognitive process dimension, specifically the 'recall' category.

Wissensdimension Faktenwissen

The task requires understanding the specific details and technical terminology related to the concept of a maximum in real-valued functions. This falls under the factual knowledge dimension.

Ausführbarkeit Akzeptabel

Question type is an implemented question type

Title is okay

Question text is okay

Option(s)/Feedback is/are okay

Aufgabenstellung Akzeptabel

The task is asking about the definition of a maximum of a real function. The answers provided are distinct and cover different possible interpretations. However, there is a repetition of the answer choices, which makes the task less clear. The correct answer is 'A) Der größte Funktionswert, den die Funktion erreicht.' The task is solvable with the given information, but the repetition should be corrected. Therefore, the decision is 'okay' with a note to remove the duplicate answers.

Auswertung ['A']

The maximum of a real function is defined as the greatest function value that the function attains. This is correctly described by statement A. Statement B describes the minimum of a function, statement C describes a point where the function is not defined, and statement D describes a critical point which could be a maximum, minimum, or saddle point.

Feedbackzweige Akzeptabel

The feedback for option A correctly confirms the correct answer and provides a clear definition of a maximum. For options B, C, and D, the feedback accurately explains why the answers are incorrect and provides additional information to guide the learner towards the correct understanding. This approach is constructive and motivating, helping the learner to understand the concept of a maximum in a real function.

Kontext Akzeptabel

The task is asking for the definition of the maximum of a real function. According to the list of mathematical concepts provided, 'Maximum [Reelle Funktion]' is the appropriate concept.

Bewertung Flaws: 0

Multiple-Choice

In diesem Abschnitt ist die erstellte *Multiple-Choice*-Aufgabe zum mathematischen Konzept *Untervektorraum* [*Vektorraum*] sowie deren Evaluationen dargestellt.

ONYX

Abb. 12: Screenshot der Multiple-Choice-Aufgabe in ONYX

Untervektorraum (MC) Punkte: 1 Keine Antwort

Welche der folgenden Aussagen beschreiben die Eigenschaften eines Untervektorraums?

- ☐ Ein Untervektorraum ist unter Addition und skalare Multiplikation abgeschlossen.
- ☐ Ein Untervektorraum ist eine nichtleere Teilmenge eines Vektorraums.
- ☐ Ein Untervektorraum ist immer ein Vektorraum.
- ☐ Ein Untervektorraum ist immer ein Untervektorraum eines anderen Untervektorraums.
- ☐ Ein Untervektorraum muss mindestens zwei Elemente enthalten.

Antwort abgeben

JSON

Codeausschnitt 22: Darstellung der Multiple-Choice-Aufgabe in JSON

```
1 {
2   "question_type": "multiple-choice",
3   "question": {
4     "title": "Untervektorraum (MC)",
5     "question_text": "Welche der folgenden Aussagen beschreiben die Eigenschaften eines
      ↳ Untervektorraums?",
6     "options": [
7       {"text": "Ein Untervektorraum ist unter Addition und skalare Multiplikation
      ↳ abgeschlossen.", "is_correct": true, "feedback": "Sieht gut aus :)"},
8       {"text": "Ein Untervektorraum ist eine nichtleere Teilmenge eines Vektorraums.",
      ↳ "is_correct": true, "feedback": "Sieht gut aus :)"},
9       {"text": "Ein Untervektorraum muss mindestens zwei Elemente enthalten.",
      ↳ "is_correct": false, "feedback": "Ein Untervektorraum kann auch die
      ↳ einelementige Menge {0} sein. Schauen Sie noch einmal im Skript nach."},
```

```

10 {"text": "Ein Untervektorraum ist immer ein Untervektorraum eines anderen
    ↳ Untervektorraums.", "is_correct": false, "feedback": "Ein Untervektorraum ist
    ↳ eine Teilmenge eines Vektorraums, nicht notwendigerweise eines anderen
    ↳ Untervektorraums. Schauen Sie noch einmal im Skript nach."},
11 {"text": "Ein Untervektorraum ist immer ein Vektorraum.", "is_correct": true,
    ↳ "feedback": "Sieht gut aus :)"}
12 ]
13 }
14 }

```

Evaluationen

Tabelle 7: Evaluationen der Multiple-Choice-Aufgabe

	Akzeptabel?	
	Expertin	Maschinell
Kognitive Prozessdimension	ja	ja
Wissensdimension	ja	ja
Aufführbarkeit	ja	ja
Aufgabenstellung	ja	nein
Auswertung	nein	ja
Feedbackzweige	ja	nein
Kontext	ja	ja
Bewertung	ja	nein

Expertin Option 4 ist im Prinzip korrekt, wird aber falsch bewertet.

Maschinell

Kognitive Prozessdimension Erinnern

The task requires the student to identify the properties of a subvector space from a list of statements. This involves recognizing the correct properties based on previously learned definitions and characteristics of subvector spaces. Therefore, the task falls under the cognitive process dimension 'remember', specifically the 'recognition' category.

Wissensdimension Faktenwissen

The task requires understanding the defining properties and characteristics of a subspace in vector spaces. This involves knowledge of specific details and technical terminology related to subspaces. Therefore, it falls under the factual knowledge dimension.

Ausführbarkeit Akzeptabel

Question type is an implemented question type

Title is okay

Question text is okay

Option(s)/Feedback is/are okay

Aufgabenstellung Unakzeptabel

The task is asking about the properties of a subvector space and is solvable with the given information. However, the task contains duplicate answer options, which makes it confusing. The technical language is correctly used, but the repetition of answer options needs to be corrected. Therefore, the decision is '-'.

Auswertung ['A', 'B', 'E']

A subvector space (or subspace) of a vector space is a subset that is itself a vector space under the same operations of addition and scalar multiplication. This means it must be closed under these operations and contain the zero vector, making it a non-empty subset. Therefore, the correct answers are A, B, and E. Option C is incorrect because a subvector space can contain only the zero vector, and option D is incorrect because a subvector space is not necessarily a subvector space of another subvector space.

Feedbackzweige Unakzeptabel

The feedback for option A is correct and confirms the correct understanding of a subvector space. However, the feedback for option B is misleading because it suggests that the statement is entirely correct, while it only partially describes a subvector space. The feedback for option C is accurate and helps the learner understand that a subvector space can include the zero vector alone. The feedback for option D is correct and clarifies the misconception about subvector spaces. The feedback for option E is correct and confirms that a subvector space is indeed a vector space. Because the feedback for B is misleading, the decision is '-'.

Kontext Akzeptabel

The task is asking for the properties of a subvector space. According to the list of mathematical concepts provided, 'Untervektorraum [Vektorraum]' is the appropriate concept.

Bewertung Flaws: 2

Drag-and-Drop

In diesem Abschnitt ist die erstellte *Drag-and-Drop*-Aufgabe zum mathematischen Konzept *Gerade* [*Polynom*] sowie deren Evaluationen dargestellt.

ONYX

Abb. 13: Screenshot der Drag-and-Drop-Aufgabe in ONYX

Gerade [Polynom] (DD)

Punkte: 1Keine Antwort

Ordnen Sie die Begriffe den Definitionen zu.

Lineares Polynom	Korrektes Element hier ablegen	Die konstante Ableitung $f'(x) = a$, die die Steigung der Geraden darstellt.
Steigung	Korrektes Element hier ablegen	Der Wert b in der Gleichung $p(x) = ax + b$, der den Schnittpunkt der Geraden mit der y-Achse angibt.
Ableitung einer Geraden	Korrektes Element hier ablegen	Der Koeffizient a in der Gleichung $p(x) = ax + b$, der die Neigung der Geraden angibt.
Quadratisches Polynom	Korrektes Element hier ablegen	Ein Polynom vom Grad 1, das die Form $p(x) = ax + b$ hat.
Nullstelle		
y-Achsenabschnitt		
Scheitelpunkt		

Antwort abgeben

JSON

Codeausschnitt 23: Darstellung der Drag-and-Drop-Aufgabe in JSON

```
1 {
2   "question_type": "drag-and-drop",
3   "question": {
4     "title": "Gerade [Polynom] (DD)",
5     "question_text": "Ordnen Sie die Begriffe den Definitionen zu.",
6     "pairs": [
7       {"left": "Lineares Polynom", "right": "Ein Polynom vom Grad 1, das die Form  $p(x)$ 
8          $\hookrightarrow = ax + b$  hat."},
9       {"left": "Steigung", "right": "Der Koeffizient  $a$  in der Gleichung  $p(x) = ax +$ 
10         $b$ , der die Neigung der Geraden angibt."},
11       {"left": "y-Achsenabschnitt", "right": "Der Wert  $b$  in der Gleichung  $p(x) = ax$ 
12         $\hookrightarrow + b$ , der den Schnittpunkt der Geraden mit der y-Achse angibt."},
13       {"left": "Ableitung einer Geraden", "right": "Die konstante Ableitung  $f'(x) =$ 
14         $a$ , die die Steigung der Geraden darstellt."}
15     ],
16     "distractors": [
17       {"left": "Quadratisches Polynom"},
18       {"left": "Nullstelle"},
19       {"left": "Scheitelpunkt"}
20     ],
21     "feedback": {
22       "correct": "Super, läuft!",
23       "incorrect": "Gucken Sie noch einmal im Skript nach, um die Begriffe besser zu
24          $\hookrightarrow$  verstehen."
25     }
26   }
27 }
```

Evaluationen*Tabelle 8: Evaluationen der Drag-and-Drop-Aufgabe*

	Akzeptabel?	
	Expertin	Maschinell
Kognitive Prozessdimension	ja	ja
Wissensdimension	ja	ja
Aufführbarkeit	ja	ja
Aufgabenstellung	ja	ja
Auswertung	ja	ja
Feedbackzweige	ja	ja
Kontext	ja	ja
Bewertung	ja	ja

Expertin Leichte Mehrdeutigkeit bei Steigung, aber ok.

Verwendung von $\text{\LaTeX}$ ist ausbaufähig und Antwortpaare könnten ergänzt werden.

Maschinell

Kognitive Prozessdimension Erinnern

The task requires the student to recall specific definitions and properties related to linear polynomials, coefficients, and derivatives. The task involves matching these definitions to the correct terms, which necessitates retrieving previously learned information from memory. Therefore, the task falls under the cognitive process dimension 'remember'.

Wissensdimension Faktenwissen

The task requires understanding the specific details and technical terminology related to linear polynomials, coefficients, and derivatives. This involves matching terms with their precise definitions, which falls under the factual knowledge dimension. Specifically, it aligns with the category of understanding concepts and technical terminology.

Ausführbarkeit Akzeptabel

Question type is an implemented question type

Title is okay

Question text is okay

Pair(s) (and Distractor(s)) is/are okay

Feedback is okay

Aufgabenstellung Akzeptabel

The task is well defined, precise, and solvable. The mathematical terms are correctly used

and the task is formulated in technical language. The variables and equations are properly surrounded with $\$$. Therefore the decision is 'okay'.

Auswertung ['1A', '2B', '3C', '4D']

The task requires matching the given definitions to the appropriate terms. A linear polynomial of degree 1 is described by the equation $p(x) = ax + b$, which matches definition A. The coefficient 'a' in the equation $p(x) = ax + b$ represents the slope of the line, which matches definition B. The value 'b' in the equation $p(x) = ax + b$ represents the y-intercept, which matches definition C. The constant derivative $f'(x) = a$ represents the slope of the line, which matches definition D. Therefore, the correct matches are: 1A, 2B, 3C, and 4D.

Feedbackzweige Akzeptabel

The feedback provided is generally helpful and motivating. The correct feedback 'Super, läuft!' is positive and encouraging, reinforcing the correct choice. The incorrect feedback 'Gucken Sie noch einmal im Skript nach, um die Begriffe besser zu verstehen.' is constructive, encouraging the learner to review the relevant material without being discouraging. Therefore, the decision is 'okay'.

Kontext Akzeptabel

The task involves matching terms related to linear polynomials and their properties, such as slope, y-intercept, and derivative. According to the list of mathematical concepts provided, 'Gerade [Polynom]' is the appropriate concept.

Bewertung Flaws: 0

Textbox

In diesem Abschnitt ist die erstellte Aufgabe vom Aufgabentyp *Textbox* zum mathematischen Konzept *Approximation [Funktionswert]* sowie deren Evaluationen dargestellt.

ONYX

Abb. 14: Screenshot der Textboxaufgabe in ONYX

Approximation [Funktionswert] (TB) Punkte: 2 Keine Antwort

Die Approximation eines Funktionswertes kann durch erfolgen, wobei nur Folgen von Treppenfunktionen zugelassen sind.

beliebige
monoton fallende
monoton wachsende

JSON

Codeausschnitt 24: Darstellung der Textboxaufgabe in JSON

```
1 {  
2   "question_type": "textbox",  
3   "question": {  
4     "title": "Approximation [Funktionswert] (TB)",  
5     "question_text": "Die Approximation eines Funktionswertes kann durch [gap1] erfolgen,  
6       ↳ wobei nur [gap2] Folgen von Treppenfunktionen zugelassen sind.",  
7     "gap": [  
8       [  
9         {"text": "Treppenfunktionen", "is_correct": true},  
10        {"text": "Polynome", "is_correct": false},  
11        {"text": "Exponentialfunktionen", "is_correct": false}  
12      ],  
13      [  
14        {"text": "monoton wachsende", "is_correct": true},  
15        {"text": "monoton fallende", "is_correct": false},  
16        {"text": "beliebige", "is_correct": false}  
17      ],  
18     "feedback": {  
19       "correct": "Super, läuft!",
```

```

20 "incorrect_gap1": "Schauen Sie sich die Methoden zur Approximation von
    ↪ Funktionswerten noch einmal an.",
21 "incorrect_gap2": "Nur eine bestimmte Art von Folgen ist für die Approximation
    ↪ zugelassen. Schauen Sie noch einmal im Skript nach."
22 }
23 }
24 }

```

Evaluationen

Tabelle 9: Evaluationen der Textboxaufgabe

	Akzeptabel?	
	Expertin	Maschinell
Kognitive Prozessdimension	ja	ja
Wissensdimension	ja	ja
Aufführbarkeit	ja	ja
Aufgabenstellung	nein	nein
Auswertung	ja	ja
Feedbackzweige	ja	ja
Kontext	nein	ja
Bewertung	nein	nein

Expertin Fourier- und Taylorpolynome sind auch Approximationen.
Kontext wohl zu eng für das Unterintegral gedacht.

Maschinell

Kognitive Prozessdimension Erinnern

The task requires the student to recall specific information about the approximation of function values and the type of sequences allowed. This involves retrieving previously learned information from memory. Therefore, the task falls under the 'remember' cognitive process dimension, specifically the 'recall' category.

Wissensdimension Faktenwissen

The task requires understanding the specific details and technical terminology related to the approximation of function values using step functions and the conditions under which these approximations are valid. This falls under the factual knowledge dimension.

Ausführbarkeit Akzeptabel

Question type is an implemented question type

Title is okay

Question text is okay

Gap(s) is/are okay

Feedback is okay

Aufgabenstellung Unkzeptabel

The task is not fully clear and precise. The term 'Treppenfunktionen' (step functions) is used, but it is not clear in what context the approximation is being discussed. Additionally, the term 'Folgen von Treppenfunktionen' (sequences of step functions) is not a standard term in mathematical literature. The task could be more precise by specifying the context, such as integration or approximation methods. Therefore, the decision is '-'.

Auswertung ['1A', '2A']

The approximation of a function value can be done using step functions, where only monotonically increasing sequences of step functions are allowed. Therefore, the correct answers are 1A and 2A.

Feedbackzweige Akzeptabel

The feedback for gap1 is appropriate and helpful. For the correct answer, 'Super, läuft!' is motivating and positive. For the incorrect answer, 'Schauen Sie sich die Methoden zur Approximation von Funktionswerten noch einmal an.' is constructive and encourages the learner to review the material. Similarly, for gap2, the feedback is also appropriate. The correct feedback is positive and encouraging, while the incorrect feedback provides a clear next step. Therefore, the decision is 'okay'.

Kontext Akzeptabel

The task is about approximating a function value using step functions, which is a concept related to the approximation of function values. According to the list of mathematical concepts provided, 'Approximation [Funktionswert]' is the appropriate concept.

Bewertung Flaws: 1

SELBSTSTÄNDIGKEITSERKLÄRUNG

Ich erkläre hiermit, dass ich diese Bachelorarbeit selbstständig ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst habe. Alle den benutzten Quellen wörtlich oder sinngemäß entnommenen Stellen sind als solche einzeln kenntlich gemacht.

Diese Arbeit ist bislang keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht worden.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Leipzig, den 31.08.2024