GOALS: Ein Lern- und Übungssystem für Lehrveranstaltungen im Zeitalter generativer KI am Beispiel einer Einführungsveranstaltung der Informatik[1]

Sven Jacobs,[2] Marc Sauer[3] und Andreas Hoffmann[4]

Die heterogene Vorbildung der Studierenden in den Einführungsveranstaltungen der Informatik stellt Hochschulen vor besondere Herausforderungen, da Informatik als Schulfach in den Bundesländern unterschiedlich verankert ist. Um diesem Umstand zu begegnen, wurde an der Universität Siegen die Übungsplattform GOALS (Graph Oriented and AI Based Learning Siegen) entwickelt. GOALS ermöglicht es den Studierenden, selbst zu entscheiden, wann sie welche Aufgaben bearbeiten, und stellt die Lerninhalte transparent in einem Konzeptgraphen dar. Durch die Integration von Large Language Models (LLMs) wird unmittelbares, formatives Feedback zu Programmieraufgaben generiert. Die Plattform wurde im Sommersemester 2024 in zwei Phasen evaluiert. Die Ergebnisse zeigen eine positive Resonanz der Studierenden hinsichtlich der Nutzbarkeit von GOALS und des generierten Feedbacks.

GOALS: A Learning and Practice System for Lectures in the Age of Generative AI Using the Example of an Introductory Computer Science Lecture

The heterogeneous prior education of students in introductory computer science courses poses particular challenges for universities, as computer science is established differently as a school subject across the federal states. To address this issue, the University of Siegen developed the practice platform GOALS (Graph Oriented and AI Based Learning Siegen). GOALS allows students to decide for themselves when to work on which tasks

1 Basiert auf einem Impulsbeitrag im Rahmen der Tagung.

2 ORCID-ID: 0009–0000-5079-7941

3 ORCID-ID: 0000–0002-9217-3085

4 ORCID-ID: 0000–0002-6870-6451

and presents the learning content transparently in a concept graph. By integrating Large Language Models (LLMs), immediate, formative feedback on programming tasks is generated. The platform was evaluated in two phases during the summer semester of 2024. The results show a positive response from students regarding the usability of GOALS and the generated feedback.

Einleitung

Hochschulische Einführungsveranstaltungen der Informatik stehen vor der besonderen Herausforderung, dass das Schulfach Informatik in den Bundesländern höchst unterschiedlich verankert ist. Im Vergleich zur Forderung von sechs Pflichtstunden Informatik in der Sekundarstufe I der ständigen wissenschaftlichen Kommission der Kultusministerkonferenz berichtet der Informatik Monitor für Nordrhein-Westfalen, dass der Anteil von Schülerinnen und Schülern mit Pflichtfach Informatik in der Sekundarstufe 24 % und der Anteil mit Informatik in der Sekundarstufe II 14 % (Rheinland-Pfalz 8 %/24 %; Hessen 0 %/10 %) beträgt (Hellmig 2023: 77).

Aus diesem Grund sitzen in den meist synchronen Informatik Einführungsveranstaltungen sehr heterogene Lerngruppen. Im Sommersemester 2023 gaben an der Universität Siegen in der Einführungsveranstaltung »Objektorientierte und funktionale Programmierung« (OFP) 43 % (n=97) der Studierenden an, dass sie in der Schule keinen Informatikunterricht besucht haben.

Um diesen Herausforderungen zu begegnen, wurde der Übungsbetrieb für die Einführungsveranstaltung OFP mithilfe einer hierzu entwickelten Übungsplattform mit dem Namen GOALS (Graph Oriented and AI Based Learning Siegen) umgestellt. Statt wöchentlich abzugebenden Übungsaufgaben können die Studierenden nun selbst über das Semester hinweg entscheiden, wann sie welche Aufgaben bearbeiten. Damit trotz asynchroner Bearbeitung unmittelbar und formativ Feedback zu den Lösungen der Studierenden gegeben werden kann, werden die neuen Möglichkeiten von Large Language Models (LLMs) eingesetzt. Hierzu wurden mehrere Komponenten unter dem Namen »Tutor Kai« (Jacobs/Jaschke, 2024a) in GOALS entwickelt.

Methodik

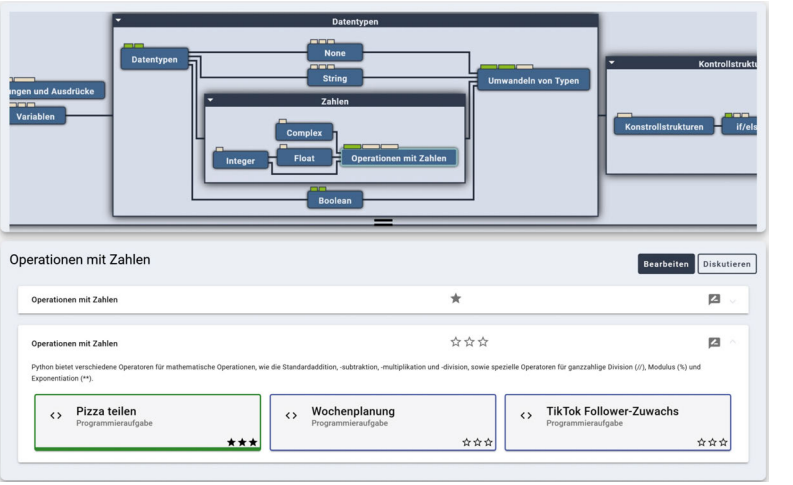
Die Evaluation der Übungsplattform fand im Sommersemester 2024 für die Einführungsveranstaltung OFP in zwei Phasen statt. An der Vorlesung nahmen circa 200 Studierende teil. Das Ziel der Evaluation war es zu untersuchen, wie die Studierenden die Nutzbarkeit von GOALS bewerten und inwiefern die anfänglichen Erwartungen der Studierenden am Ende des Semesters aus ihrer Perspektive erfüllt wurden.

Zu Beginn wurden nach erstmaliger Vorstellung von GOALS die Erwartungen der Studierenden an die Übungsplattform und deren LLM-Integrationen mit einem Fragebogen (Evaluation A) erhoben. Dieser wurde von 97 Studierenden vollständig ausgefüllt. Am Ende der Vorlesungszeit und noch vor der Klausur wurden die Erfahrungen der Studierenden sowie die Nutzbarkeit von GOALS mit einem zweiten Fragebogen (Evaluation B) erfasst (n=58). Die Teilnahme war freiwillig.

GOALS Übungsplattform

In der ersten Entwicklungsstufe von GOALS wurde zunächst ein Konzeptgraph entwickelt, welcher noch nicht für Knowledge Tracing (vgl. Abdelrahman/Wang/Nunes 2023), sondern vielmehr zur Übersicht und Navigation eingesetzt wird. Da die Studierenden selbst entscheiden sollen, wann sie welche Aufgaben bearbeiten, ist es dabei notwendig, die zugehörigen Konzepte und ihre Abhängigkeiten darzustellen. Je Inhaltsbereich wird das Niveau der angestrebten Kompetenzförderung anhand der Kategorien der kognitiven Prozessdimension nach Anderson und Krathwohl (2001) innerhalb eines Graphen dargestellt. Der initiale Konzeptgraph entstand durch die Expertenanalyse (Modulverantwortliche) der in der Form von Folien, Video-Aufzeichnungen und Aufgabenstellungen vorhandenen Vorlesungsmaterialien. Dabei wurden diese in Sinnabschnitte unterteilt und nach Konzepten gruppiert. Anschließend wurden Abhängigkeiten modelliert und die vorhandenen Aufgaben basierend auf den kognitiven Prozessdimensionen den Konzepten zugeordnet. Zu jedem Konzept wurden passende Multiple-Choice-Aufgaben basierend auf dem Transkript der Vorlesungsaufzeichnung durch ein LLM generiert, durch Tutoren geprüft und freigegeben (siehe Abschnitt 4.2). Anschließend wurde der initiale Graph in Zusammenarbeit mit zwei Modulverantwortlichen überarbeitet.

Abb. 1: User Interface von GOALS



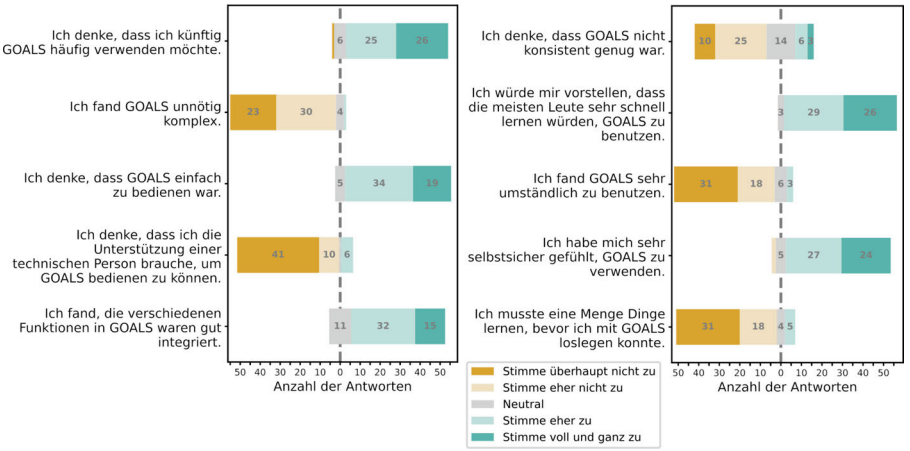
Damit der Konzeptgraph nicht nur zur Visualisierung, sondern auch zur Navigation genutzt werden kann, muss dieser übersichtlich gestaltet sein. Aus diesem Grund werden Abhängigkeiten vereinfacht dargestellt, indem beispielsweise Knoten innerhalb eines Subgraphen nur Abhängigkeiten untereinander haben (vgl. Abb. 1: oberer Bereich). Zusätzlich werden die Kategorien der kognitiven Prozessdimension vereinfacht als Level dargestellt und jeder Aufgabe visuell nur eines dieser Level zugeordnet (Abb. 1: unterer Bereich).

Zur Evaluation der Nutzbarkeit von GOALS wurde der System Usability Scale (vgl. Brooke 1996) in seiner deutschen Übersetzung (vgl. Gao/Kortum/Oswald 2020: 17) innerhalb Evaluation B verwendet. Die Einzelergebnisse (vgl. Abb. 2) und der Gesamtwert von 80,4 müssen unter der Einschränkung interpretiert werden, dass Studierende, die nicht mit GOALS oder der Vorlesung zurechtgekommen sind, gegebenenfalls nicht mehr auf Anfragen reagierten. Bezüglich des Konzeptgraphen wurden die 58 Studierenden in der Evaluation B außerdem gefragt, wie nachvollziehbar dessen Aufbau war und wie gut die Navigation durch diesen gelang. Hinsichtlich des Aufbaus äußerten 15 Studierende und bezüglich der Navigation 20 Studierende, dass Verbesserungspotenzial besteht.

Neben dieser Evaluation der gesamten Plattform und des Konzeptgraphen wurden die diversen Integrationen von generativer künstlicher Intelligenz in

Form von LLMs für die Generation von Aufgaben und formativen Feedback einzeln untersucht (vgl. Jacobs et al. 2025a, Jacobs et al. 2025b).

Abb. 2: System Usability Scale: Ergebnisse der Evaluation B (n=58)



Integration von Large Language Models

Die Möglichkeiten von generativer künstlicher Intelligenz (GenAI) für die Einführung in die Programmierung sind umfangreich (Brett et al. 2023, Prather et al. 2023 sowie Denny et al. 2024). In GOALS wurde das zu Beginn des Sommersemesters 2024 aktuelle Large Language Model »gpt-4-turbo-2024-04-09« von OpenAI (OpenAI 2023) verwendet, um automatisch formatives Feedback zu Programmieraufgaben zu geben sowie die Multiple-Choice- und Programmier-Aufgaben zu generieren.

Feedback

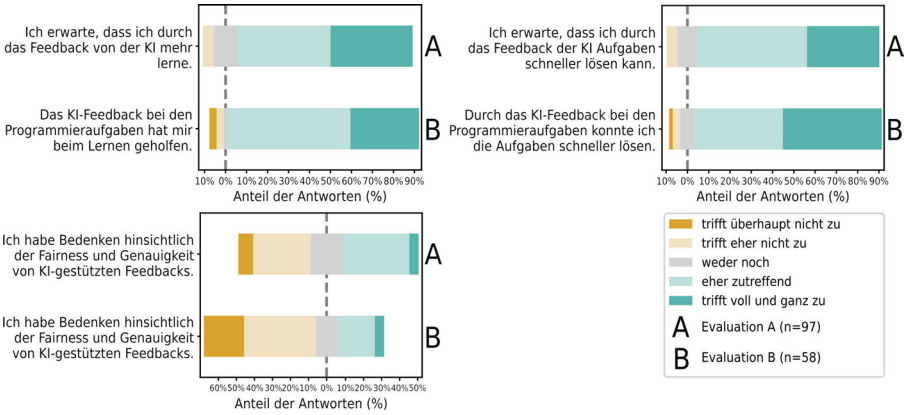
Die Wirksamkeit von Feedback ist hinlänglich belegt (vgl. Hattie/Timperley 2007) und kann durch fehlerspezifische strategische Informationen erheblich zum Lernfortschritt beitragen (vgl. Narciss 2006). Die automatische Generierung von Feedback bei Programmieraufgaben ist bereits seit geraumer Zeit Gegenstand von Forschung (vgl. Keuning/Jeuring/Heeren 2019). In diesem

Kontext werden häufig statische Tests eingesetzt, die sich nur begrenzt anpassen lassen und primär zur Lokalisierung von Fehlern dienen. Large Language Models können diese Funktionen erweitern, indem Probleme dynamischer adressiert werden und Feedback individuell formuliert wird.

Bei einem ersten Feedback-Prototypen für GOALS im Sommersemester 2023 wurden hierzu die Aufgabenstellung, die Lösung des Studierenden, die Ausgabe des Compilers sowie ein Prompt (vgl. Schulhoff et al. 2024: 5) an das LLM gesendet. Innerhalb des Prompts wurde das LLM (GPT-4-0314) dazu gebracht, ein kurzes Feedback zu geben, welches notwenige Konzepte erklärt aber keinesfalls Teile der Lösung verrät. Zur Evaluation bewerteten 51 Studierende 1243 Feedbacks, inwiefern Ihnen das Feedback geholfen hat. Auf einer Likert Skala von eins (überhaupt nicht hilfreich) bis sieben (sehr hilfreich) erreicht der Prototyp hierbei einen Wert von 5,05. Bei einer Expertenanalyse von 263 Feedbacks stellte sich jedoch heraus, dass in sechs Prozent der generierten Feedbacks halluzinierte Fehler beschrieben wurden und zwölf Prozent der Feedbacks zu einer falschen Lösung geführt hätten (vgl. Jacobs/Jaschke 2024a).

Im darauffolgenden Semester wurde das Feedback mittels Retrieval Augmented Generation (vgl. Gao et al. 2024) basierend auf Transkripten der Vorlesungsaufzeichnungen erweitert, sodass innerhalb des Feedbacks passende Stellen aus den Vorlesungsaufzeichnungen zitiert und verlinkt werden können (vgl. Jacobs/Jaschke 2024b). Auf diese Weise soll die Anzahl der Halluzinationen verringert werden und zudem sollen die bekannten Erklärungen aus der Vorlesung den Studierenden beim Erinnern unterstützen. In einem Klausurvorbereitungsworkshop mit 15 Studierenden in Wintersemester 2023/24 wurde dieser Ansatz erstmalig erprobt. Die Ergebnisse deuten darauf hin, dass die verlinkten Vorlesungsaufzeichnungen im Feedback hilfreich für die Studierenden sind (vgl. Jacobs/Jaschke 2024b). Da der Ansatz jedoch zwei aufeinander aufbauende Anfragen an das LLM benötigt, ist die Zeit bis zum ersten Wort des Feedbacks deutlich länger. Durch die Veröffentlichung des im Sommersemester 2024 verwendeten LLMs GPT-4 Turbo konnte diese Zeit deutlich reduziert werden.

Abb. 3: Erwartungen an und Erfahrungen mit dem generierten Feedback



Während der Vorlesungszeit des Sommersemesters 2024 wurden für die Programmieraufgaben insgesamt 9917 Feedbacks generiert. Zu Beginn und Ende des Semesters wurden zudem die Erwartungen (Evaluation A) und Erfahrungen (Evaluation B) der Studierenden an und mit dem generierten Feedback untersucht. Die Ergebnisse (vgl. Abb. 3: Evaluation A) legen nahe, dass die Studierenden nahezu ausschließlich davon ausgingen, durch das Feedback mehr zu lernen und die Aufgaben schneller bearbeiten zu können. Die Erwartungen der Studierenden wurden aus ihrer Perspektive erfüllt, wobei die Zustimmung hinsichtlich des Lerneffekts leicht gesunken ist und hinsichtlich der Bearbeitungszeit leicht gestiegen ist (vgl. Abb. 3: Evaluation B). Zudem sind die Bedenken hinsichtlich der Fairness und Genauigkeit im Vergleich von Evaluation A und Evaluation B gesunken.

In einer Think-Aloud-Studie, die zusätzlich zu Beginn des Sommersemesters 2024 durchgeführt wurde, wurde der Nutzen des Feedbacks für die Studierenden qualitativ untersucht. Dabei zeigte sich, dass Studierende mit Programmiererfahrung mehr vom Feedback profitieren als Studierende ohne Vorkenntnisse (vgl. Jacobs et al. 2025b).

Aufgaben generieren

In GOALS können sowohl Programmieraufgaben als auch Multiple- und Single-Choice (MC) Aufgaben automatisch mittels LLMs generiert werden.

Bei den MC Aufgaben wurde das jeweils relevante Transkript der Vorlesungsaufzeichnung als Kontext für das LLM genutzt (Retrieval Augmented Generation), damit die Fragen inhaltlich zu den tatsächlich behandelten Themen passen. Nach einer anschließenden Revision durch Tutoren und Modulverantwortliche wurden die Fragen für Studierende freigeschaltet. Über 90 % der so generierten MC Aufgaben konnten ohne Korrektur für das Sommersemester 2024 freigegeben werden.

Bei der Generierung von kontextuell personalisierten Programmieraufgaben wurden alle notwendigen Aufgabenbestandteile wie Aufgabenstellung, Programmgerüst, Musterlösung und Unit-Tests basierend auf Programmierkonzepten und einem frei wählbaren Kontext generiert. 200 generierte Python-Aufgaben wurden von Experten anhand unterschiedlicher Kriterien positiv bewertet (vgl. Jacobs et al. 2025a). Anschließend generierten 26 Studierende in einer Übungswoche 167 Programmieraufgaben basierend auf ihren individuell gewählten Kontexten (z. B. Fußball-EM 2024). Von diesen bewerteten sie 104 Aufgaben und gaben an, dass über 90 % der Aufgaben den gewünschten Kontext integrieren und genügend Informationen zur Lösung der Aufgaben gegeben sind. Außerdem bewerteten die Studierenden die Personalisierung der Aufgaben sowie die Möglichkeit, immer weitere Aufgaben generieren zu können, sehr positiv (vgl. Jacobs et al. 2025a).

Fazit

Die Einführung der Übungsplattform GOALS in ihrer ersten Entwicklungsstufe hat eine transparente und übersichtliche Darstellung von Inhalten und Aufgaben in Form eines Konzeptgraphen ermöglicht. Durch die Integration von Large Language Models (LLMs) zur Generierung von formativem Feedback können Studierende die Aufgaben selbstbestimmt und in ihrem eigenen Tempo bearbeiten, ohne an wöchentliche Abgabetermine gebunden zu sein. Die Studierenden bewerteten die Nutzbarkeit der Plattform sehr positiv und empfanden das generierte Feedback als hilfreich. Die hohen Erwartungen an das generierte Feedback wurden aus Sicht der Studierenden erfüllt. Auch die vorgestellten Ansätze zur Generation von Programmieraufgaben erscheinen vielversprechend. Obwohl die Aufgaben, die mit den derzeitigen LLMs (GPT-4o) generiert werden, noch eine manuelle Überprüfung erfordern, könnte dies in Zukunft mit verbesserten Systemen und LLMs nicht mehr der Fall sein.

In zukünftigen Weiterentwicklungen soll der Konzeptgraph zusammen mit dem darin abgebildeten Fortschritt der Studierenden als Lerner-Modell genutzt werden. Dies eröffnet die Möglichkeit, adaptive Feedbackstrategien zu entwickeln und Aufgaben zu generieren, die einen individuellen Lebensweltbezug aufweisen. Auf diese Weise soll GOALS noch besser auf die heterogenen Vorkenntnisse und Bedürfnisse der Studierendenschaft zugeschnitten werden, was zu einem personalisierten und effektiveren Lernprozess beitragen soll.

Literatur

Abdelrahman, Ghodai/Wang, Qing/Nunes, Bernardo (2023): »Knowledge Tracing: A Survey«, in: ACM Computing Surveys 55(11), S. 1–37.

Anderson, Lorin W./Krathwohl, David R. (2001): A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, New York: Addison-Wesley.

Becker, Brett A./Denny, Paul/Finnie-Ansley, James/Luxton-Reilly, Andrew/Prather, James/Santos, Eddie Antonio (2023): »Programming Is Hard – Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation«, in: SIGCSE 2023: The 54th ACM Technical Symposium on Computer Science Education, S. 500–506, https://doi.org/10.1145/3545945.3569759

Brooke, John (1996): »SUS: A ›Quick and Dirty‹ Usability Scale«, in: Patrick W. Jordan/Bruce Thomas/Ian L. McClelland/Bernard Weerdmeester (Hg.), Usability Evaluation in Industry, London: Taylor & Francis, S. 189–194.

Denny, Paul/Prather, James/Becker, Brett A./Finnie-Ansley, James/Hellas, Arto/Leinonen, Juho/Luxton-Reilly, Andrew/Reeves, Brent N./Santos, Eddie Antonio/Sarsa, Sami (2024): »Computing Education in the Era of Generative AI«, in: Communications of the ACM 67, S. 56–67.

Gao, Meiyuzi/Kortum, Philip/Oswald, Frederick L. (2020): »Multi-Language Toolkit for the System Usability Scale«, in: International Journal of Human–Computer Interaction 36(20), S. 1883–1901.

Gao, Yunfan/Xiong, Yun/Gao, Xinyu/Jia, Kangxiang/Pan, Jinliu/Bi, Yuxi/Dai, Yi/Sun, Jiawei/Guo, Qianyu/Wang, Meng/Wang, Haofen (2024): Retrieval-Augmented Generation for Large Language Models: A Survey, https://doi.org/10.48550/arXiv.2312.10997

Hattie, John/Timperley, Helen (2007): »The Power of Feedback«, in: Review of Educational Research 77, S. 81–112.

Hellmig, Lutz/Schieckoff, Bentley/Schwarz, Richard/Süßenbach, Felix (2023): Informatik-Monitor 2023/24: Zur Situation des Informatikunterrichts in Deutschland, Berlin: Gesellschaft für Informatik e.V, https://informatik-monitor.de/2023-24

Jacobs, Sven/Jaschke, Steffen (2024a): »Evaluating the Application of Large Language Models to Generate Feedback in Programming Education«, in: 2024 IEEE Global Engineering Education Conference (EDUCON), S. 1–5, https://doi.org/10.1109/EDUCON60312.2024.10578838

Jacobs, Sven/Jaschke, Steffen (2024b): »Leveraging Lecture Content for Improved Feedback: Explorations with GPT-4 and Retrieval Augmented Generation«, in: 2024 36th International Conference on Software Engineering Education and Training (CSEE&T), S. 1–5, https://doi.org/10.1109/CSEET62301.2024.10663001

Jacobs, Sven/Peters, Henning/Jaschke, Steffen/Kiesler, Natalie (2025a): »Unlimited Practice Opportunities: Automated Generation of Comprehensive, Personalized Programming Tasks«, in: Proceedings of the 2025 Conference on Innovation and Technology in Computer Science Education Vol. 1 (ITiCSE), https://doi.org/10.48550/arXiv.2503.11704

Jacobs, Sven/Kempf, Maurice/Kiesler, Natalie (2025b): »That's Not the Feedback I Need! – Student Engagement with Compiler and GenAI Feedback in Tutor-Kai«, Manuskript eingereicht.

Keuning, Hieke/Jeuring, Johan/Heeren, Bastiaan (2019): »A Systematic Literature Review of Automated Feedback Generation for Programming Exercises«, in: ACM Transactions on Computing Education 19, S. 1–43.

Narciss, Susanne (2006): Informatives tutorielles Feedback. Entwicklungs- und Evaluationsprinzipien auf der Basis instruktionspsychologischer Erkenntnisse, Münster: Waxmann.

OpenAI (2023): GPT-4 Technical Report, https://doi.org/10.48550/arXiv.2303.08774

Prather, James/Denny, Paul/Leinonen, Juho/Becker, Brett A./Albluwi, Ibrahim/Craig, Michelle/Keuning, Hieke/Kiesler, Natalie/Kohn, Tobias/Luxton-Reilly, Andrew/MacNeil, Stephen/Petersen, Andrew/Pettit, Raymond/Reeves, Brent N./Savelka, Jaromir (2023): »The Robots Are Here: Navigating the Generative AI Revolution in Computing Education«, in: Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education, New York: ACM, S. 108–159.

Schulhoff, Sander/Ilie, Michael/Balepur, Nishant/Kahadze, Konstantine/Liu, Amanda/Si, Chenglei/Li, Yinheng/Gupta, Aayush/Han, HyoJung/Schulhoff, Sevien/Dulepet, Pranav Sandeep/Vidyadhara, Saurav/Ki, Dayeon/Agrawal, Sweta/Pham, Chau/Kroiz, Gerson/Li, Feileen/Tao, Hudson/Srivastava, Ashay/Da Costa, Hevander/Gupta, Saloni/Rogers, Megan L./Goncearenco, Inna/Sarli, Giuseppe/Galynker, Igor/Peskoff, Denis/Carpuat, Marine/White, Jules/Anadkat, Shyamal/Hoyle, Alexander/Resnik, Philip (2024): The Prompt Report: A Systematic Survey of Prompting Techniques, https://doi.org/10.48550/arXiv.2406.06608