Desarrollo y Evaluación de un Tutor Inteligente para el aprendizaje de programación basado en los Modelos de Lenguaje Extenso

M.C. Oleksiy Levchuk

Departamento de Ciencias de la Computación – Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE) – Ensenada, Baja California – México

levchuk@cicese.edu.mx

Abstract. *The emergent behavior of automatic programming following the popularization of Generative Artificial Intelligence has raised uncertainty about the future of programming and its teaching. This doctoral work proposes the design and evaluation of an architecture for the development of Intelligent Tutoring Systems for programming learning, integrating Large Language Models to provide a personalized user experience. The architecture is developed using a design-based research methodology, assessing its effect on cognitive engagement and learning through formative prototype evaluations and a summative evaluation conducted via an intervention study.*

Resumen. *El comportamiento emergente de la programación automática tras la popularización de la Inteligencia Artificial Generativa ha generado incertidumbre sobre el futuro de la programación y su enseñanza. Este trabajo doctoral propone diseñar y evaluar una arquitectura para el desarrollo de Sistemas de Tutoría Inteligente para el aprendizaje de programación que integran Modelos de Lenguaje Extenso para ofrecer una experiencia de usuario personalizada. La arquitectura se desarrolla con una metodología de investigación basada en diseño, evaluándose su efecto en el compromiso cognitivo y el aprendizaje mediante pruebas formativas del prototipo y una evaluación sumativa a través de un estudio de intervención.*

1. Introducción

Los recientes avances en Aprendizaje Profundo, impulsados por el uso de GPUs para realizar cálculos intensivos de manera altamente paralela (Goodfellow et al., 2016) y la introducción de la arquitectura de red neuronal Transformador (Vaswani et al., 2017) han permitido el desarrollo de Modelos de Lenguaje Extenso (**LLMs**) con la capacidad de procesar y generar información de manera coherente y contextualmente relevante. El acceso a grandes volúmenes de datos provenientes de diversas fuentes textuales facilitó el preentrenamiento (fase en la que un LLM aprende estructuras del lenguaje antes de su ajuste para tareas específicas) de estos modelos para capturar patrones lingüísticos complejos, haciendo posible el desarrollo de la Inteligencia Artificial Generativa (**IAG**), IA diseñada para crear contenido nuevo en diferentes formatos (Radford et al., 2019).

Los LLMs pueden utilizarse para desarrollar agentes conversacionales capaces de generar respuestas coherentes y contextuales (Brown et al., 2020), útiles en tareas como traducción, creación de contenido, diseño de prototipos y asistencia educativa (Rahman & Watanobe, 2023). En el ámbito de la programación, estos modelos pueden actuar como asistentes inteligentes que sugieren fragmentos de código, explican conceptos y ofrecen soluciones a errores comunes. Por ejemplo, GitHub Copilot y Llama-Code son capaces de generar código de manera eficiente, facilitando la escritura y depuración de programas. Sin embargo, aunque los LLMs están transformando la práctica del desarrollo de software, la evidencia sugiere que no reemplazan a programadores humanos, sino que elevan los requisitos de su labor y preparación profesional (Vaithilingam et al., 2022).

Los Sistemas de Tutoría Inteligente (**ITS**) son aplicaciones de software diseñadas para proporcionar instrucción y retroalimentación personalizada, adaptándose a las necesidades específicas de cada usuario. Esta área, con varias décadas de desarrollo, ha demostrado en ciertos contextos igualar o superar la efectividad de un tutor humano (Zhai & Wiebe, 2023). Sin embargo, su arquitectura tradicional carece de conciencia contextual (limitando su cobertura temática) y está asociada con altos costos y tiempos de desarrollo (Liu et al., 2023). El reciente éxito de los LLMs en tareas de programación abre oportunidades para fortalecer la arquitectura y aumentar la efectividad de los ITS.

Un estudio reciente que analiza más de 4 millones de conversaciones entre humanos y el asistente conversacional Claude (Tamkin et al., 2025) señala que el uso de la IAG se concentra en el desarrollo de software y las tareas de escritura, con un mayor impacto en ocupaciones relacionadas con la informática y las matemáticas. Dado que los procesos de desarrollo ya están siendo transformados por la IAG, es crucial que los futuros ingenieros de software tengan acceso a herramientas educativas de vanguardia. No obstante, la integración de LLMs en ITS presenta desafíos clave en la depuración y evaluación de su comportamiento debido a la naturaleza probabilística de los LLMs (Bommasani et al., 2021), además de la dificultad de combinarlos eficazmente con los módulos y las subrutinas de un ITS, aspectos que se abordarán en este trabajo doctoral.

2. Planteamiento del problema

Esta investigación busca responder las siguientes preguntas de investigación y alcanzar los objetivos establecidos para abordar la cuestión de integración de LLMs en ITS.

1. ¿En qué áreas de un ITS para programación los LLMs pueden contribuir a la personalización del contenido y en cuáles presentan limitaciones?
2. ¿Qué características de diseño de un ITS para programación basado en LLMs promueven el compromiso cognitivo de los estudiantes?
3. ¿Cómo se pueden ajustar los LLMs en el contexto de un ITS para programación para potenciar el aprendizaje en comparación con otros enfoques de enseñanza?

2.1. Objetivo General

Diseñar y evaluar una arquitectura para el desarrollo de un ITS para programación que integre LLMs con el objetivo de personalizar el contenido educativo, fomentar el compromiso cognitivo y mejorar el aprendizaje, proporcionando asistencia a los estudiantes en la resolución de problemas computacionales.

2.2. Objetivos Específicos

- Diseñar estrategias para integrar LLMs en ITS, identificando componentes clave para personalizar el contenido según distintos perfiles de estudiantes.
- Explorar y comparar estrategias de ajuste fino de LLMs para fomentar el compromiso cognitivo de los estudiantes durante el aprendizaje de programación.
- Implementar evaluaciones formativas e iterativas de la arquitectura desarrollada, integrando principios de diseño instruccional y experiencia de usuario.
- Realizar una evaluación sumativa del prototipo desarrollado en contextos educativos reales para medir su impacto en el aprendizaje de programación.

3. Trabajo relacionado

El compromiso cognitivo (grado de esfuerzo mental y atención que se invierte en el proceso de aprendizaje o resolución de problemas) de los estudiantes se debe tomar en cuenta para comprender los desafíos en la enseñanza y el aprendizaje de programación, donde las altas tasas de fracaso reflejan la necesidad de enfoques que no solo enseñen a producir código, sino que también fomenten una mayor atención en la comprensión de los conceptos subyacentes (Halverson & Graham, 2019). Un metaanálisis de más de 5000 artículos sobre la enseñanza y el aprendizaje de programación (Scherer et al., 2020) identificó que los enfoques constructivistas que combinan autonomía con orientación docente en entornos colaborativos con retroalimentación inmediata y el uso de herramientas tecnológicas son más efectivos que los métodos educativos tradicionales.

En el contexto de los ITS basados en LLMs existen varios prototipos que intentan alinearse con los enfoques educativos constructivistas, como Ruffle&Riley, que genera guiones de tutoría a partir de textos de lecciones y fomenta un modelo de aprendizaje basado en la enseñanza a otros: su evaluación mostró altos niveles de compromiso y comprensión por parte de los estudiantes (Schmucker et al., 2023). Sin embargo, la investigación se ha centrado principalmente en evaluar las capacidades de los LLMs de manera independiente, fuera del contexto de un ITS, observando su efecto positivo en estudiantes de programación (Qureshi, 2023; Denny et al., 2023), así como su inevitable integración en la educación, señalando la necesidad de establecer límites y promover un uso responsable de los LLMs (Rahman & Watanobe, 2023).

La integración de un LLM dentro de la arquitectura de un ITS depende del ajuste fino, un proceso que adapta el modelo a tareas específicas mediante su entrenamiento con datos especializados. Sonkar y colaboradores (2024) emplearon un conjunto de datos pedagógicos sintéticos para ajustar varios LLMs a la tarea de proporcionar ayuda personalizada, logrando mejoras significativas en la calidad de las explicaciones y la capacidad de guiar a los estudiantes. Gao y colaboradores (2024) ajustaron un LLM al contexto de la enseñanza de Inteligencia Artificial, mejorando su capacidad para asistir en el aprendizaje autorregulado dentro de un sistema de visualización interactiva.

Una revisión bibliográfica realizada en el marco de esta investigación identificó que las primeras referencias a ITS que integran LLMs en alguno de sus módulos o subrutinas datan de 2023, lo que evidencia la reciente emergencia de este campo de estudio. Además, la literatura existente no ofrece un consenso sobre la metodología óptima para el diseño y desarrollo de estos sistemas, ya que los estudios disponibles presentan limitaciones o insuficiencias en la evaluación de su diseño arquitectónico.

4. Metodología de investigación

La estrategia de investigación adoptada se fundamenta en el enfoque metodológico *Design Science Research* (Johannesson & Perjons, 2021), el cual se utiliza para crear artefactos innovadores que resuelven problemas prácticos, mientras se genera conocimiento teórico sobre su funcionamiento y posibles mejoras. En este contexto, los pasos fundamentales incluyen: identificación del problema, diseño y desarrollo, demostración, evaluación y, finalmente, comunicación (Figura 1).

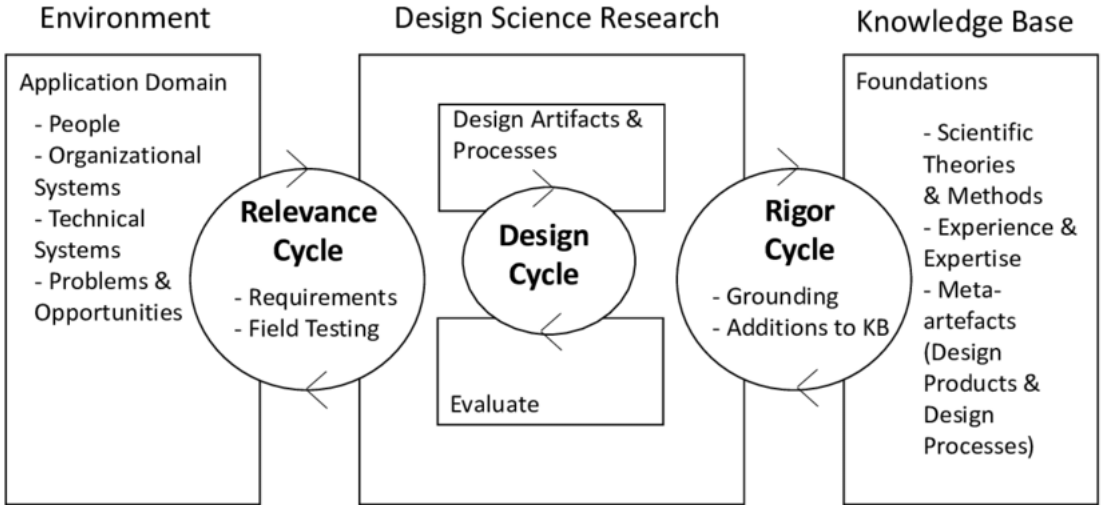

Figura 1: *Design Science Research* (Johannesson & Perjons, 2021)

Algunas de las fases de desarrollo previamente mencionadas se abordaron de manera parcial durante mi estancia de maestría, por ejemplo:

1. Se realizaron entrevistas a 10 profesores-investigadores y se aplicaron encuestas a 150 estudiantes de programación de diversos niveles educativos para explorar su percepción y uso de agentes conversacionales en la enseñanza y el aprendizaje de programación. Los resultados revelaron un marcado uso de estas herramientas entre los estudiantes, así como una actitud receptiva por parte de los profesores. Esta tendencia generalmente positiva hacia su adopción en el entorno educativo destaca la necesidad de avanzar hacia soluciones más estructuradas y personalizadas, como los ITS basados en LLMs (Levchuk, 2024).
2. Se llevó a cabo una revisión de la literatura sobre las capacidades de distintos LLMs para realizar tareas relacionadas con la programación en entornos escolares, identificando el ajuste de instrucciones como una técnica factible para trabajar con LLMs de forma remota sin modificar directamente los parámetros del modelo base. Además, bajo las métricas propuestas, el modelo GPT-4 fue seleccionado como el más pertinente para proporcionar retroalimentación personalizada a los estudiantes usando la técnica previamente mencionada.
3. Se exploro la técnica de ajuste de instrucciones para crear el prototipo de un ITS para programación que combina la funcionalidad de un agente conversacional con un cuaderno computacional. Emplea una API privada de GPT-4 para comunicarse directamente con el LLM seleccionado y una serie de instrucciones prediseñadas para controlar el contenido textual generado y su interacción con el usuario para prevenir plagio académico (Levchuk et al., 2024).

5. Solución propuesta y trabajo a futuro

Para abordar una cuestión tan multimodal como el diseño de una arquitectura que permita la incorporación de LLMs dentro de los ITS, se propone dividir el proyecto en varios pasos alineados directamente con las tres preguntas de investigación:

1. **Revisión bibliográfica:** Para identificar y mapear la literatura disponible mediante una revisión de alcance, se analizarán artículos que evidencien el uso de LLMs para potenciar la funcionalidad de los ITS. Los resultados se presentarán en una investigación registrada en la plataforma *Open Science Framework* y contribuirán al desarrollo de una arquitectura basada en el estado del arte.
2. **Métricas de personalización:** Con base en la primera pregunta de investigación y en la evidencia bibliográfica que respalda el uso de LLMs para la generación de material didáctico (Denny et al., 2023), el apoyo en la resolución de ejercicios (Qureshi, 2023) y la automatización del proceso de tutoría (Schmucker et al., 2023), se pretende establecer directrices de diseño que garanticen no solo la eficiencia técnica en la incorporación de un LLM dentro de cierto módulo o subrutina del ITS, sino también la maximización de su capacidad de personalización de contenido educativo para estudiantes de programación.
3. **Diseño de la arquitectura:** Con base en la segunda pregunta de investigación y en la evidencia bibliográfica de que niveles altos de personalización se asocian con una disminución de la carga cognitiva intrínseca (Lange, 2021), así como en estudios que indican que la personalización del aprendizaje mejora el rendimiento (59 %) y el compromiso cognitivo (36 %) de los estudiantes (du Plooy et al., 2024), se diseñarán prototipos que aprovechen esta capacidad para fomentar el compromiso cognitivo. Dado que la programación demanda habilidades cognitivas avanzadas, es crucial analizar la carga cognitiva (cantidad de esfuerzo mental requerido para procesar información en la memoria de trabajo) que enfrentan los estudiantes y su capacidad para gestionarla.
4. **Ajuste de los LLMs:** Se abordarán y evaluarán diversas técnicas de ajuste de LLMs, como el ajuste fino, la recuperación y generación, la ingeniería de instrucciones y el aprendizaje por refuerzo, entre otras. Se emplearán librerías como Hugging Face Transformers y PyTorch, con su posterior evaluación mediante herramientas como PandaLM. Este proceso implicará la interacción con modelos de código abierto (Llama, DeepSeek, etc.) y de código cerrado (GPT-4, Bard, etc.), con el objetivo de identificar combinaciones que permitan una mayor personalización y, por consiguiente, un mayor compromiso cognitivo.
5. **Enfoque en la cognición:** Con base en la tercera pregunta de investigación y en la evidencia de que promover procesos cognitivos entre estudiantes mejora la adquisición de conocimientos en programación (Singh & Rajendran, 2024), así como en estudios que reportan una fuerte correlación positiva entre el compromiso cognitivo y el éxito/productividad académica (Khan et al., 2023), se diseñará una interfaz de usuario que facilite interacciones intuitivas pero desafiantes. Esta interfaz ofrecerá apoyo sin revelar directa ni indirectamente la solución al problema planteado, mediante la implementación de elementos visuales dinámicos, retroalimentación inmediata y personalizada, así como mecanismos para el monitoreo y la intervención del profesor.

6. **Evaluaciones formativas:** Se realizarán evaluaciones formativas de los módulos y componentes del prototipo de ITS basado en la arquitectura propuesta, integrando tanto retroalimentación cualitativa obtenida a través de entrevistas y encuestas dirigidas a estudiantes y profesores, como datos cuantitativos relacionados con el rendimiento funcional del sistema (pruebas A/B).
7. **Evaluación sumativa:** Se llevará a cabo un estudio de intervención con evaluación sumativa para medir el impacto del ITS en el compromiso cognitivo y el aprendizaje de programación en los estudiantes. El estudio incluirá un grupo de control y un grupo de tratamiento, con una duración de ocho semanas dentro del currículo de clases de programación. Se aplicarán pruebas de rendimiento antes y después de la intervención, así como una prueba de retención un mes después. Se analizarán métricas clave como el progreso en habilidades, el tiempo de resolución, la precisión y la complejidad de las tareas.

6. Ventajas de la solución propuesta

A diferencia de investigaciones centradas en evaluar la capacidad de los LLMs para generar contenido específico, desempeñar tareas concretas o simular roles particulares, este trabajo doctoral propone el diseño y la evaluación de una arquitectura para el desarrollo de ITS basados en LLMs para el aprendizaje de programación. En contraste con la aplicación de estos modelos para suplir funcionalidades específicas dentro de módulos individuales (Fernández et al., 2024), se busca integrar la IAG de manera holística, con el objetivo de optimizar múltiples subrutinas y módulos del sistema, siempre que se identifiquen beneficios significativos en esta sinergia, representando una aplicación más amplia y unificada de los LLMs en la educación de programación, superando las limitaciones de enfoques tradicionales.

La arquitectura propuesta se centra en fomentar la autonomía de los estudiantes, permitiéndoles resolver problemas por sí mismos mientras adquieren un entendimiento más profundo a través de retroalimentación personalizada. Simultáneamente, el sistema proporciona herramientas para monitorear y gestionar las interacciones entre estudiantes y LLMs, asegurando un control efectivo del proceso educativo y facilitando intervenciones pedagógicas en momentos críticos. Este enfoque integral y centrado en la cognición tiene el potencial de transformar la forma en que los ITS basados en LLMs se diseñan y utilizan en el aprendizaje de programación y otros dominios educativos.

7. Conclusiones

Esta investigación propone generar conocimiento sobre el diseño, implementación y evaluación de una arquitectura para el desarrollo de ITS para programación que integren LLMs para personalizar el contenido educativo, fomentar el compromiso cognitivo y mejorar el aprendizaje. Se busca identificar la configuración óptima de los componentes que permiten la personalización del contenido, determinar características de diseño que fomenten el compromiso cognitivo y explorar distintas estrategias de ajuste fino para maximizar el aprendizaje adquirido. Los resultados esperados contribuirán al avance de la Ingeniería de Software, proporcionando lineamientos para la integración efectiva de LLMs en ITS y su aplicación en la enseñanza de programación.

Dado que se trata de un campo emergente, esta investigación no busca crear una herramienta definitiva, sino generar conocimiento y responder la pregunta controversial: ¿Como podemos asegurarnos de que la integración de LLMs en los ITS fomente un aprendizaje activo en lugar de generar dependencia? Explorar algo trascendental en lugar de centrarse en un LLM en particular ayudaría a preservar la relevancia de este trabajo, además de que se hará público el prototipo desarrollado en base a la arquitectura diseñada a través de la plataforma de GitHub y se compartirá con la comunidad científica a través de publicaciones en revistas de impacto.

8. Referencias

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., ... & Liang, P. (2021). "On the opportunities and risks of foundation models". arXiv preprint. https://doi.org/10.48550/arXiv.2108.07258

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). "Language models are few-shot learners". Advances in neural information processing systems, 33, 1877-1901. https://doi.org/10.48550/arXiv.2005.14165

Denny, P., Prather, J., Becker, B. A., Finnie-Ansley, J., Hellas, A., Leinonen, J., ... & Sarsa, S. (2024). "Computing education in the era of generative AI". Communications of the ACM, 67(2), 56-67. https://doi.org/10.1145/3624720

Du Plooy, E., Casteleijn, D., & Franzsen, D. (2024). "Personalized adaptive learning in higher education: a scoping review of key characteristics and impact on academic performance and engagement". Heliyon, 10(21), e39630. https://doi.org/10.1016/j.heliyon.2024.e39630

Fernández, L. R., Mena, A. L. F., Magaña, M. P. T., Magaña, M. A. R., & Fernández, M. A. R. (2024). "Inteligencia artificial en la educación: Modelo de lenguaje de gran tamaño (LLM) como recurso educativo". Revista IPSUMTEC, 7(2), 157-164. https://doi.org/10.61117/ipsumtec.v7i2.321

Gao, L., Lu, J., Shao, Z., Lin, Z., Yue, S., Ieong, C., ... & Chen, S. (2024). "Fine-tuned large language model for visualization system: A study on self-regulated learning in education". IEEE Transactions on Visualization and Computer Graphics. http://dx.doi.org/10.1109/TVCG.2024.3456145

Goodfellow, I., Bengio, Y., & Courville, A. (2016). "Deep learning". MIT Press.

Halverson, L. R., & Graham, C. R. (2019). "Learner Engagement in Blended Learning Environments: A Conceptual Framework". Online Learning, 23, 145-178. https://doi.org/10.24059/olj.v23i2.1481

Johannesson, P., & Perjons, E. (2021). "An Introduction to Design Science". In Springer eBooks. https://doi.org/10.1007/978-3-030-78132-3

Khan, H., Gul, R., & Zeb, M. (2023). "The Effect of Students' Cognitive and Emotional Engagement on Students' Academic Success and Academic Productivity". Journal Of Social Sciences Review, 3(1), 322-334. https://doi.org/10.54183/jssr.v3i1.141

Lange, C. (2021). "The relationship between e-learning personalization and cognitive load". Open Learning the Journal of Open Distance And e-Learning, 38(3), 228-242. https://doi.org/10.1080/02680513.2021.2019577

Levchuk, O. (2024). Diseño y evaluación de un tutor inteligente basado en Inteligencia Artificial Generativa para la adquisición de habilidades de programación. Tesis de Maestría en Ciencias. CICESE, Baja California, México. 92 pp.

Levchuk, O., Sánchez, C., Pacheco, N., López, I., & Favela, J. (2024). "Interaction Design (IxD) of an Intelligent Tutor for Programming Learning Based on LLM". Avances en Interacción Humano-Computadora, 9(1), 1–10. https://doi.org/10.47756/aihc.y9i1.137

Liu, Z., He, X., Liu, L., Liu, T., & Zhai, X. (2023). "Context matters: A strategy to pre-train language model for science education". In International Conference on Artificial Intelligence in Education, 666-674. Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-36336-8_103

Qureshi, B. (2023). "Exploring the use of chatgpt as a tool for learning and assessment in undergraduate computer science curriculum: Opportunities and challenges". ArXiv preprint. https://doi.org/10.48550/arXiv.2304.11214

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. & Sutskever, I. (2019). "Language Models are Unsupervised Multitask Learners". OpenAI.

Rahman, M. M., & Watanobe, Y. (2023). "ChatGPT for Education and Research: Opportunities, Threats, and Strategies". Applied Sciences, 13(9), 5783. https://doi.org/10.3390/app13095783

Scherer, R., Siddiq, F., & Viveros, B. S. (2020). "A meta-analysis of teaching and learning computer programming: Effective instructional approaches and conditions". Computers In Human Behavior, 109, 106349. https://doi.org/10.1016/j.chb.2020.106349

Schmucker, R., Xia, M., Azaria, A., & Mitchell, T. (2023). "Ruffle&riley: Towards the automated induction of conversational tutoring systems". ArXiv preprint. https://doi.org/10.48550/arXiv.2310.01420

Singh, D., & Rajendran, R. (2024). "Cognitive engagement as a predictor of learning gain in Python programming". Smart Learning Environments, 11(1). https://doi.org/10.1186/s40561-024-00330-9

Sonkar, S., Ni, K., Chaudhary, S., & Baraniuk, R. G. (2024). "Pedagogical alignment of large language models". arXiv preprint. https://doi.org/10.48550/arXiv.2402.05000

Tamkin, A., Liu, K., Valle, R., & Clark, J. (2025). "Which Economic Tasks are Performed with AI? Evidence from Millions of Claude Conversations". Anthropic. assets.anthropic.com/m/2e23255f1e84ca97/original/Economic_Tasks_AI_Paper.pdf

Vaithilingam, P., Zhang, T., & Glassman, E. L. (2022). "Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models". CHI EA '22: CHI Conference on Human Factors in Computing Systems Extended Abstracts, Article 332, 1–7. https://doi.org/10.1145/3491101.3519665

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). "Attention is all you need". Advances in Neural Information Processing Systems, 30, 5998–6008. https://doi.org/10.48550/arXiv.1706.03762

Zhai, X., & Wiebe, E. (2023). "Technology-based innovative assessment". In Classroom-Based STEM Assessment: Contemporary Issues and Perspectives, 99–125.