

Article

A Generative AI-Empowered Digital Tutor for Higher Education Courses

Hila Reicher ¹, Yarden Frenkel ¹, Maor Juliet Lavi ¹, Rami Nasser ¹ , Yuval Ran-milo ¹ , Ron Sheinin ¹ , Mark Shtai² and Tova Milo ^{1,*}

¹ School of Computer Science, Tel Aviv University, Tel Aviv 69778, Israel; hilareicher@mail.tau.ac.il (H.R.); yardenfo@gmail.com (Y.F.); maorlavi@tauex.tau.ac.il (M.J.L.); raminasser@tauex.tau.ac.il (R.N.); yuv.milo@gmail.com (Y.R.-m.); ronsheinin@mail.tau.ac.il (R.S.)

² School of Electrical Engineering, Tel Aviv University, Tel Aviv 69778, Israel; shtai@tauex.tau.ac.il

* Correspondence: milo@cs.tau.ac.il

Abstract: This paper explores the potential of AI-based digital tutors to enhance student learning by providing accurate, course-specific answers to complex questions, anchored in validated course materials. The Tel Aviv University Digital Tutor (TAUDT) exemplifies this approach, enabling students to navigate and comprehend academic content with ease. By citing specific passages in course materials, TAUDT ensures pedagogical accuracy and relevance while fostering independent learning. Its modular design allows for the seamless integration of advancements in AI and state-of-the-art technologies, ensuring long-term adaptability and performance. Designed to integrate effortlessly into existing academic workflows, TAUDT requires no technological expertise from instructors, addressing barriers posed by technophobia among faculty. A pilot study demonstrated high levels of student engagement, highlighting its potential as a scalable, adaptive solution for higher education.

Keywords: digital tutoring; LLM; AI; higher education



Academic Editor: Leon Sterling

Received: 12 February 2025

Revised: 16 March 2025

Accepted: 19 March 2025

Published: 26 March 2025

Citation: Reicher, H.; Frenkel, Y.; Lavi, M.J.; Nasser, R.; Ran-milo, Y.; Sheinin, R.; Shtai, M.; Milo, T. A Generative AI-Empowered Digital Tutor for Higher Education Courses. *Information* **2025**, *16*, 264. <https://doi.org/10.3390/info16040264>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Artificial intelligence (AI) presents an unprecedented opportunity to scale higher education and enhance its quality. Significant efforts are being directed toward realizing this potential [1–10]; however, the rapid and effective adoption of AI faces several key challenges. One major challenge is that while AI systems, particularly those leveraging generative models, are powerful, they are also prone to inaccuracies—commonly referred to as “hallucinations” [11]. Newer AI models reduce such errors, but maintaining accuracy, especially in alignment with the pedagogical structure and terminology set by educators, remains an ongoing challenge. The other challenge lies in effectively engaging the educational staff and keeping them involved [12], despite the characteristic conservatism and technophobia often found in large research universities, where faculty members tend to resist adopting new technologies or altering their established teaching practices [13].

In this context, we developed the Tel Aviv University Digital Tutor (TAUDT), an AI-empowered chatbot designed to enhance student engagement with course materials outside the classroom. TAUDT integrates seamlessly into existing academic workflows, allowing students to ask questions at their own pace and receive personalized, relevant answers drawn from a course’s specific content. Platforms with similar capabilities have recently been reported, both in the academic literature [3–7] and in commercial products [8–10], as outlined in Section 2. However, they do not yet provide sufficiently robust solutions to the two challenges mentioned above, on which this work concentrates.

Firstly, to address the issue of accuracy and hallucinations, TAUDT anchors all responses in validated course materials, ensuring that students receive accurate and contextually relevant answers. While reviewing course materials—whether through watching recorded lectures or reading lecture notes—students can pause their studies and ask the digital tutor questions related to the material. TAUDT's answers include not only a textual explanation but also references to relevant sections of the course content. By clicking on these references, students can access the precise points in the lecture video or reading materials that support the answer, allowing them to verify the information.

It is important to note that in a course's context, answers must not only be correct but also align with the pedagogical structure of the curriculum. They must also adhere to a desired notation and style and follow the specific teaching sequence determined by faculty. For instance, answers generated by AI must not rely on concepts or content that students have not yet encountered. TAUDT respects this structure by retrieving content from course materials and generating responses that align with the topics and concepts presented in class.

Secondly, addressing the issue of university conservatism and the reluctance of faculty to modify their teaching styles or invest in learning new technologies, TAUDT integrates effortlessly into existing course management systems. Course materials such as lecture video-recordings, notes, slides, and reading materials are uploaded into TAUDT through a simple interface. Faculty can continue their usual teaching practices while the system processes the materials, making them available for interactive learning.

However, building a system like TAUDT comes with its own technical hurdles. The rapid development of large language models (LLMs), where competing models with new capabilities are released periodically, necessitates a modular design to allow TAUDT to incorporate newer models without requiring significant re-engineering. This adaptability is key to ensuring that the system remains at the forefront of AI advancements. Another hurdle is operational cost. Commercial LLMs often charge based on input size, meaning that feeding unnecessary information into the system can be expensive. Additionally, when too much context is provided, LLM responses can become less accurate. These issues are addressed in TAUDT through a retrieval-augmented generation (RAG) architecture [14], which retrieves only the most relevant portions of the course materials in response to student queries, thereby minimizing cost while ensuring more precise and focused answers.

By addressing the above listed challenges and technical hurdles, TAUDT provides a powerful solution for improving student engagement and learning outcomes without placing additional burdens on educators. Its intuitive interface and seamless integration into existing systems make it an effective tool for supporting independent learning, while its design ensures contextually accurate responses and state-of-the-art performance.

To evaluate the system's operation, a pilot study was conducted in an undergraduate course at the Faculty of Engineering of Tel Aviv University on Computer Architecture, with 100 enrolled students. TAUDT was employed as an interactive tool, allowing students to query course-related materials outside of the classroom environment. Based on the positive outcomes of this pilot, the system is now being deployed in 30 additional courses across the university. This expansion will provide further insights into TAUDT's scalability and adaptability across a variety of academic disciplines.

The remainder of this paper is structured as follows: Section 2 provides a review of related work in the field, highlighting existing efforts and identifying gaps in current technology. In Section 3, we delve into user interaction models for both educators and students. We then describe the system architecture and design choices in Section 4. Section 5 outlines the operational deployment and evaluation methodology. A pilot study with real-

world application results is presented in Section 6. Finally, we provide conclusions in Section 7, discussing the implications of our findings and provides a roadmap for future research directions.

2. Review of Related Work

Many studies have been conducted in the attempt to assess the implications of AI-based chat-bot technologies in higher education [15–20]. These include consideration of the effects on student learning methods [16], exploration of the potential of AI-tools in enhancing the learning experience [17,19–22], empirical assessment of the impact and effectiveness of available AI tools [21–25], and aspects of ethics and academic integrity [15,17,18,20].

Nonetheless, the implementation of a platform that functions as a custom-designed virtual tutor capable of handling complex academic materials in higher education is still at its infancy. Early attempts to generate such platforms were conducted in the pre-LLM era, i.e., before tools like Open AI's ChatGPT [9] (https://openai.com/index/introducing-gpts/?utm_source=chatgpt.com, accessed on 24 November 2024) and others became available [1,2]. Prominent examples include the work of Knobloch et al. in 2018 [1], reporting automatic handling of student questions in the process of a lecture. However, although it contains AI components, the answers to most student queries required contributions from a human expert. Somewhat later in 2020 [2], Zylich et al. demonstrated a truly autonomous system built around a self-made recurrent neural network (RNN) designed to answer queries by students in higher education. That work contains many of the essential building blocks deployed in later systems (including ours); however, due to the limited power of the RNN, it focused primarily on peripheral aspects such as course logistics, and not on the academic content itself.

A significant leap forward occurred in the post-LLM era, with the introduction of commercial generative AI products. These developments provided widespread access to powerful generative AI models, enabling the design of applications across various fields. In particular, chat-bots that are based on commercial LLM engines and are specifically designed for higher education began to emerge [3–7]. The first attempt was reported by Sajja et al. [3] in early 2023, very shortly after ChatGPT's appearance, and it offered students intelligent interaction with a chat-bot on curriculum, syllabi and course logistics. Subsequently, Liu et al. [4] presented an AI tool functioning as an instructor in a student forum, responding to queries on a variety of course materials, including course logistics and content. The later work by Sajja et al. [5] reports a virtual tutor that answers student queries on the course materials; in this sense, this work is closest to ours. It also answers queries related to logistics, offering students self-testing possibilities and homework evaluations. Finally, Google has reported on its experimental AI-driven educational platform, Shiffbot, powered by the Gemini application programming interface (API), which showcases the sophisticated integration of generative AI into interactive learning environments [7]. Although Shiffbot exemplifies advances in leveraging AI for adaptive learning and content delivery, it relies heavily on the educator's active configuration and intervention, contrasting with TAUDT's focus on minimizing educator workload.

In addition to the academic efforts described above, there has been a recent surge in commercial, customized general-purpose chatbots that have potential applications as teaching assistants. A few examples of such chat-bots are Open AI's GPT creator [9], Poe [8], and CustomGPT.ai [10]. These products allow the teacher to upload course materials and specify the style in which the chat-bot replies to queries. Although such commercial chat-bot design platforms are rapidly evolving, they are not optimized for the academic scene, and the answers that they provide are not characterized by the level of reliability and rigor required in higher education.

The innovation in this work is that we place extra emphasis on the subject of relevance and reliability that was outlined in the introduction. We argue that it is vital to equip students with mechanisms to verify the accuracy of the information provided and ensure that it is firmly grounded in their course materials. Additionally, it is important to distinguish between statements that are based on general knowledge and those grounded in course-specific materials, thereby fostering trust and clarity. As we demonstrate in what follows, TAUDT addresses this by marking statements based on general knowledge and by including direct citations to relevant sections of the course content, allowing students to trace responses back to authoritative sources.

Another notable advantage of the system described in this paper is its ability to log all queries, responses, and student feedback in a format amenable to automated analysis. By examining these student–tutor interactions, valuable insights can be gained into student comprehension, enabling targeted optimizations of course materials and teaching methods to better address student needs.

Finally, it is important to review some of the basic ideas behind LLMs and LLM-based systems, so that the choices that we make in TAUDT can be properly justified.

Large Language Models (LLMs): Large language models (LLMs) are a category of artificial intelligence systems that use vast amounts of textual data to generate human-like responses [26]. These models are trained on diverse data sources and can understand and answer questions based on provided input. A common method for adapting LLMs for specific tasks is by carefully crafting the input prompts to guide the model's output without altering the underlying model. This approach is known as prompt engineering and it is particularly useful with closed source commercial LLMs such as OpenAI's GPT series, Anthropic's Claude, Google's Gemini, and others [27].

Retrieval-Augmented Generation (RAG): Retrieval-Augmented Generation (RAG) [14] is a method that combines two techniques: information retrieval and language generation. In this method, given a query, the system first retrieves relevant documents or data from a specific corpus (such as the course materials in TAUDT) and then uses an LLM to generate a coherent response based on that retrieved information. RAG is particularly suitable for systems like TAUDT, where cost and accuracy are important considerations. By retrieving only the most relevant pieces of information before querying the LLM, the system minimizes unnecessary processing and improves the accuracy of the generated response. An essential component of RAG is embeddings, which are numerical representations of text that capture its semantics, so that texts with similar embeddings have similar meanings. By converting both the query and the database content into embeddings, the system can efficiently match queries with relevant content, ensuring that only the pertinent sections of the course material are retrieved.

Frameworks for Developing LLM-Based Applications: Developing LLM-based applications like TAUDT often requires integrating multiple AI functions, such as handling prompts, managing data retrieval, as well as interacting with external systems. Several frameworks have been designed to simplify this process. Common examples include LangChain, LlamaIndex, Haystack and others, all of which offer tools to build applications that connect LLMs with data sources like databases and APIs. Langchain is used in the development of TAUDT [28].

3. User Interaction with the TAUDT System

TAUDT was designed to seamlessly integrate into existing university workflows, providing both teaching staff and students with an intuitive and efficient interface for engaging with course materials. This section outlines the interaction models for both educators and students. A link to a short video tutorial can be found here [29].

3.1. Teaching Staff Interaction

Teaching staff interact with the system primarily through the course setup phases. Course materials are uploaded directly from the university course management system through a simple interface. Supported formats include lecture notes, slides, and video recordings, which are then transcribed automatically. The current TAUDT version processes only text data, utilizing the transcriptions rather than raw video content. In the courses that we have examined, working with only this textual data was found to be surprisingly effective in providing valuable results.

After uploading the materials, the teaching staff is prompted to input a set of 20 examples of typical student questions along with corresponding answers. These serve as a benchmark for automatic testing, ensuring that the system's responses are aligned with the lecturer's expectations. Intuitively, the system uses these examples to test itself: the TAUDT generates responses to the questions, which are compared to the provided answers using dedicated metrics. System parameters are fine-tuned based on this analysis to improve accuracy (for details, see Section 5).

Upon completing this setup, the course is ready for deployment. The teaching staff have access to anonymized summaries of student interactions, offering opportunities for enhancing future lectures.

3.2. Student Interaction

Student interactions with TAUDT occur through a secure, user-friendly interface, which is depicted in Figure 1. After signing in using authorized university credentials, students can access the main interface where they can create new conversations linked to their registered courses or resume previous conversations. The conversational interface is inspired by popular chat-based tools, allowing students to pose questions through a text box at the bottom of the window.

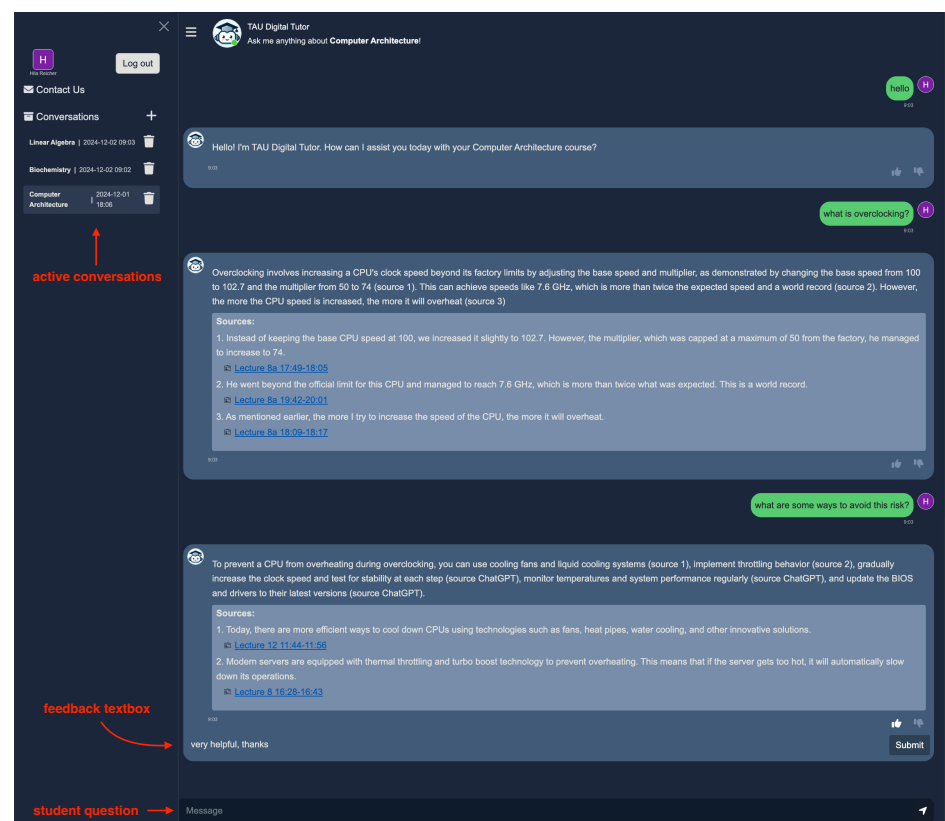


Figure 1. An example of TAUDT's conversational UI.

TAUDT's responses are structured to enhance learning and provide references to the course materials. Each response includes embedded citations, such as "(Source 1)", "(Source 2)" or "(source ChatGPT)". Numbered citations like "Source 1" or "Source 2" link back to specific parts of the course content which support the answer. The corresponding citations—pointers to the material—are listed below the answer. By clicking on these citations (pointers), students are directed to the relevant section of the lecture notes or video, enabling them to verify the relevance and accuracy of the response. A non-numbered citation like "ChatGPT" indicates that the preceding text was derived using general knowledge of the employed LLM. Pointing out ChatGPT's general knowledge contributions is meant to encourage the student to treat these contributions cautiously until their correctness is verified independently using other sources.

Example: To better illustrate the interaction flow between a student and TAUDT, consider the scenario depicted in Figure 1. A student is reviewing lecture materials from the "Computer Architecture" course. While watching a recorded lecture, the student hears a term, "overclocking", which she does not recall. She pauses the video and types the question, "What is overclocking?", into the TAUDT interface. The system searches through the lecture materials, finding relevant segments in the course video which explain the concept and responding with a concise summary. The answer explains that overclocking is a method to increase the CPU's clock's speed (pointing to Sources 1 and 2—two specific points in the course video where this information is described), a process that may potentially cause the CPU to overheat (pointing to Source 3). The student wants to dive deeper to understand how such overheating may be avoided; hence, she asks a follow-up question, "What are some ways to avoid this risk?". TAUDT's response here includes two cooling methods that were mentioned in class (sources 1 and 2) and three additional ones suggested by ChatGPT. Observe that distinguishing the answer parts originating from the course material vs. ChatGPT enables critical reading—the student can easily follow and examine the links to the supporting course extracts, while exercising further caution for information provided by ChatGPT only.

Feedback is a critical component of the interaction model. Students can rate responses using thumbs up/down icons and submit more detailed comments through an additional text box. This input is crucial for refining the system and identifying areas for improvement. The sidebar offers options for managing ongoing conversations, accessing previous queries, and logging out of the system.

4. System Architecture and Technical Design

The Tel Aviv University Digital Tutor (TAUDT) leverages the power of LLMs and a retrieval-augmented generation (RAG) framework and is structured to support modular integration of different AI components. This section outlines the high-level design of the system, details its key components, and describes the query processing pipeline. The system consists of three main layers, which are depicted in Figure 2:

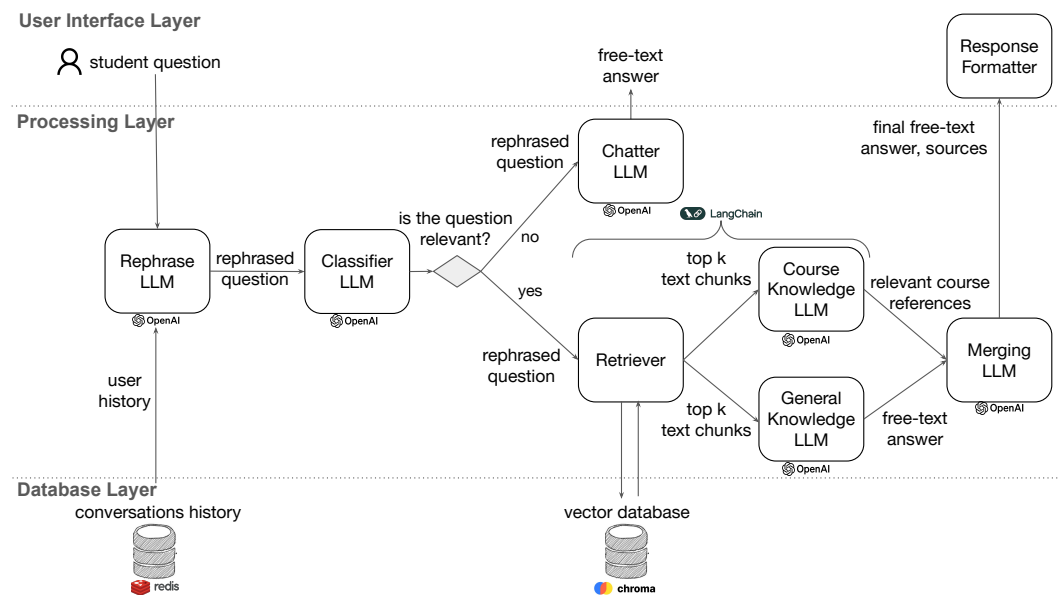


Figure 2. TAUDT's architecture.

Frontend User Interface Layer: The frontend provides students with an intuitive user interface (UI) to submit questions, view responses, and manage conversations. Each interaction is associated with a specific course, ensuring that answers remain contextually relevant.

Data Storage Layer: This layer manages course materials and user data and includes two corresponding databases. Course materials are split into chunks (e.g., course video intervals, sections of transcribed material, etc.), each of which is processed into a numerical vector representation, a process known as embedding [30], and the embedding vector is stored in a database. This embedding vector will be later used by the system's RAG for the retrieval of the chunks most relevant for the student query. Student interaction histories are stored in another database for efficient query-time context management, as described below. All student ID's are anonymized to fully preserve users' privacy.

Backend Processing Layer: The backend query processor constitutes the heart of the system, and its functionality is detailed in what follows. It manages the processing of user queries and includes modules for preprocessing, routing, query refinement, and response synthesis. This layer also handles the interaction with the two databases and external services like LLMs.

The processing pipeline is designed around a series of interconnected components, each serving a specific role in transforming a raw student question into a structured and informative response. The key components involved in this process are the following.

4.1. Input Handling and Rephrasing

A question that is part of a conversation may be phrased in an ambiguous or context-dependent manner. For example, a question like "What are some ways to handle this risk?" cannot be understood without relating it to prior stages in the conversation. To address this, the first step involves rephrasing the question to resolve ambiguities and ensure it can be understood independently. This is handled by the Rephrase LLM module which examines the stored conversation history and transforms the student's input into a standalone question. When there is no conversation history, the rephrased question remains identical to the submitted query. The question in the above example is rephrased to something like "What are some ways to prevent a CPU from overheating during overclocking?",

assuming that the CPU overheating while overclocking was the subject of the earlier stage in the conversation.

4.2. Query Classification

Once the question has been rephrased, it enters the *classifier LLM*. The primary function of this component is to distinguish between actionable questions and interactions that do not require an extensive response, such as greetings or acknowledgments (“hello”, “thanks”, etc.). If the query is classified as actionable and relevant, it proceeds to the retrieval phase described below. Otherwise, the system generates a direct response (e.g., “Hello! I am Tel Aviv University Digital Tutor. How can I assist you today?”) using the *chatter LLM* module, which emulates a conventional LLM interaction, without invoking more complex processing.

4.3. Data Retrieval

For actionable questions, the system initiates two parallel RAG-based computations.

Given a query, the process starts by searching the database, which contains the pre-processed course materials, to identify the most relevant chunks of information from the course, such as a specific chunk of the lecture notes or of the transcribed lecture videos. This is performed by the *retriever* module. The search uses embeddings. Namely, the query and each chunk in the course materials are represented as vectors (each being a set of numerical values) that capture the semantic meaning of the corresponding text. By comparing the embedding of the query with the embeddings of the course material chunks, the system retrieves the chunks that best match the meaning of the student’s question.

The retrieved chunks are then processed in parallel by two LLMs that differ in their prompts.

Course Knowledge LLM: The course knowledge LLM is prompted to further refine the retrieval and retain a subset of the chunks which appear contextually most relevant for generating a query’s answer.

General Knowledge LLM: The general knowledge LLM, on the other hand, is prompted to generate a new virtual chunk which contains the LLM answer to the question. In the generation of this virtual chunk, the LLM is instructed to utilize both the retrieved chunks of the course materials and its own general knowledge as well as supplement, if needed, the course-specific data with additional context, examples, or related concepts. For instance, consider our running example where the student asks for ways to avoid CPU overheating. The general knowledge LLM may respond with a virtual chunk that includes some of the methods described in class, but it may also include additional, broader industry practices not mentioned there (e.g., the possibility of gradually increasing the clock speed or system monitoring). As another example, if a student asks, ‘Can you give me an example of [a concept or calculation]?', and an example was not directly presented in class, the general knowledge LLM can provide a relevant example that complements the course material.

One of the deliberate decisions we made during the system design process was to input the retrieved chunks of the course materials to the general knowledge LLM as well, rather than simply prompting it to answer based on its general knowledge. The main reason for this choice is to ensure that the virtual chunk generated by the general knowledge LLM aligns with the language, terminology, and context used in the course material, thereby reducing the risk of discrepancies.

4.4. Response Merging

The *merging LLM* module is then responsible for synthesizing an answer to the student’s question out of the outputs of the course knowledge and general knowledge LLMs. It

is prompted to ensure that the answer is cohesive and includes both detailed course-specific information and relevant supplementary knowledge when needed. If the two outputs contain overlapping or contradictory information, the merging LLM resolves conflicts by prioritizing the course-specific content and using general knowledge to fill in gaps or provide clarifications. This is also performed by prompting the LLM with corresponding instructions. The merged response is post-processed by the *response formatter* module to include references and links to the course material. These references are formatted as clickable links that direct students to the precise section of the lecture notes or video, allowing for easy verification. More specifically, the response formatter first identifies the quotes appearing in the answer and matches them to the relevant retrieved chunk. The chunks are then mapped to their respective “locations” in the course materials. Since the embedded chunks are stored in the database along with their origin information, we can reliably trace back each retrieved chunk: For textual sources (e.g., PDFs or text files), the formatter retrieves page numbers. For video-based materials, it identifies timestamps within associated subtitle (.srt) files and constructs direct links to the relevant video segments. The links in the response are then generated using this metadata. Finally, To improve readability, the formatter numbers the sources (e.g., Source 1, Source 2, Source 3) and ensures they appear in the same order as in the free-text answer.

4.5. Answer Delivery and Feedback Handling

The final synthesized response is sent back to the student through the frontend UI, followed by citations from the course materials as described earlier and a feedback prompt giving the students an option to rate the response (e.g., thumbs up/down) and provide additional comments. This feedback is logged in the database and used by the system administrator to analyze and refine the system operation, and by the teaching staff for the purpose of improving pedagogical aspects when relevant.

It is important to note that when a student asks a question that does not have even a partial answer within the course material, the system still processes the query through the entire flow; however, if no relevant course references are retrieved, the system responds by informing the student that there is not enough information available to answer the question, even in cases where the general knowledge LLM might know the answer. This again is a deliberate design choice intended to prevent students from abusing the system as a general-purpose Q&A tool. An alternative solution may be to extend the capabilities of the classifier to identify course-unrelated questions. However, while saving processing efforts, this approach entails the risk of false negatives which may result in missed opportunities to provide valuable answers to students. We leave the examination of this alternative for future study.

4.6. Technological Stack

For readers interested in the technological aspects of the implementation, the application is developed using a Python-based Flask 3.0.0 backend and a React 18.3.1 frontend. Both components are containerized using Docker 26.1.1 and deployed on AWS EC2 instances. The core storage and processing components include Chroma 0.4.21 (for vector database management), Redis 7.2 (for user interaction history), and LangChain 0.2.5 (for managing LLM interactions). User authentication is securely handled via OAuth 2.0, ensuring access is restricted to authorized university accounts. Monitoring is managed through AWS CloudWatch, allowing for real-time performance tracking and alerting. In our system’s deployment, we utilize OpenAI’s GPT-4o model for all LLM tasks.

5. System Setup

To rigorously assess the performance of TAUDT and the impact of key design decisions throughout its development, we implemented a structured, automated evaluation process inspired by the RAGAS [31] approach. The evaluation inputs consist of ground-truth question-and-answer pairs, typically provided by the course teaching staff during the course setup (as described in Section 3). TAUDT generates responses to these questions, which are compared to the lecturer-provided answers using several metrics. These metrics are adaptations of corresponding metrics presented in RAGAS [31] and adjusted to our context, as detailed below.

Agreement Score: An answer consists of two parts: a textual reply and a set of references. The Agreement Score evaluates the textual response, measuring how closely TAUDT's output aligns with the lecturers, providing an indication of how well the system mimics expert responses.

Reference Score: This metric assesses the similarity between the system's retrieved references and the ground-truth references provided by the teaching staff.

Groundedness Score: This metric measures the extent to which TAUDT's textual answers are based on the listed references, ensuring that responses are grounded in the relevant content retrieved during the query processing.

Question-Answer Relevance Score: This metric gauges how well TAUDT's responses address the student query, ensuring their relevance.

All metrics are calculated using large language models (LLMs) prompted with specific instructions corresponding to each evaluation criterion.

Furthermore, we track the system's response time to ensure that performance remains efficient even as the system evolves.

This evaluation process allows for continuous fine-tuning of system parameters to improve performance. Each time a modification is made, the same set of ground-truth questions is used to assess how the change impacts performance. Key system parameters that are iteratively optimized include the following:

Chunk number: To determine the number k of course chunks to be retrieved in the retrieval step.

Chunk size: To optimize the granularity of the retrieved content.

Embedding model selection: To identify the text embedding model that best captures the course material's semantics.

LLM model selection: To choose the most appropriate LLM model for different stages of the query-processing pipeline.

We used the Weights & Biases platform [32] to manage and visualize the results of each evaluation run. This platform allows us to track system performance across various configurations, run parameter sweeps, and systematically optimize combinations of settings to enhance accuracy and system behavior. Examples for tradeoffs observed in parameter adjustments are that higher k -values (resp. chunk size) yield improved retrieval accuracy but increase the response time (cost). We opted for $k = 5$ and (non-overlapping) chunks of 1000 characters, for all materials, resulting in an average response time of 11 s.

6. Pilot Study

To evaluate the system's effectiveness, a pilot study was conducted in the "Computer Architecture" undergraduate course of the Faculty of Engineering at Tel Aviv University with 100 enrolled students.

6.1. Study Setup and Execution

The course videos were transcribed automatically and uploaded to TAUDT. A link to the digital tutor was made available on the course website, and students were encouraged to use it. Student interactions were logged, and feedback was collected using a thumbs-up/down mechanism along with a text box for additional comments.

The initial system version was rather basic and included only the *retriever* LLM, the *course knowledge* LLM and the *answer formatter*. The feedback collected from student interactions was used to iteratively improve the system and add new functionalities. This includes support for the conversational chat (implemented by the *rephrase* LLM), treatment of non-actionable interactions (implemented by the *classifier* LLM and the *chatter* LLM modules), and for the enrichment of answers with general knowledge (implemented by the *general knowledge* LLM and the *merging* llm). More technical improvements included bug fixes and an upgrade to the embedding model. See Table 1 for the list of updates.

Table 1. System updates: features and release dates.

Feature Name	Release Date
Launch	3 June 2024
Conversational chat	3 July 2024
Upgraded embedding model	6 July 2024
General knowledge from ChatGPT	14 August 2024
Question classifier	21 August 2024

The evaluation was primarily based on the student's feedback, including the frequency and consistency of use, the relevance of the answers provided (as assessed by the course lecturer), and the system's reliability. Recall that all student ids are anonymized to fully preserve user privacy. Quantitative metrics such as the average response time per query, the system's ability to provide any answer at all, and the frequency of errors were also tracked. Additionally, we measured user engagement, examining how many students consistently returned to use the tool throughout the semester.

6.2. Results

The pilot study yielded valuable insights into student engagement, system performance, and areas for improvement.

Student Engagement and Usage The digital tutor experienced substantial adoption and use throughout the semester. Of the 100 enrolled students, 78 actively used the system at least once, collectively submitting 2107 queries. Among these users, 74.36% returned for multiple sessions, showcasing consistent engagement with the tool as a valuable learning resource.

On average, each session lasted 1.42 min, during which students asked approximately 2.35 questions. A session was defined as a series of interactions separated by no more than five minutes. This highlights the system's utility in facilitating quick, focused learning interactions, making it well suited for on-the-go or time-constrained studying. Figure 3 depicts the queries' number per week throughout the semester, showing peak usage in the final week of the semester and for exam preparation. Interestingly, some queries were submitted even after the exam, with students using TAUDT to verify or clarify issues encountered during the exam. This pattern underscores TAUDT's utility not only for ongoing learning but also for retrospective review and comprehension.

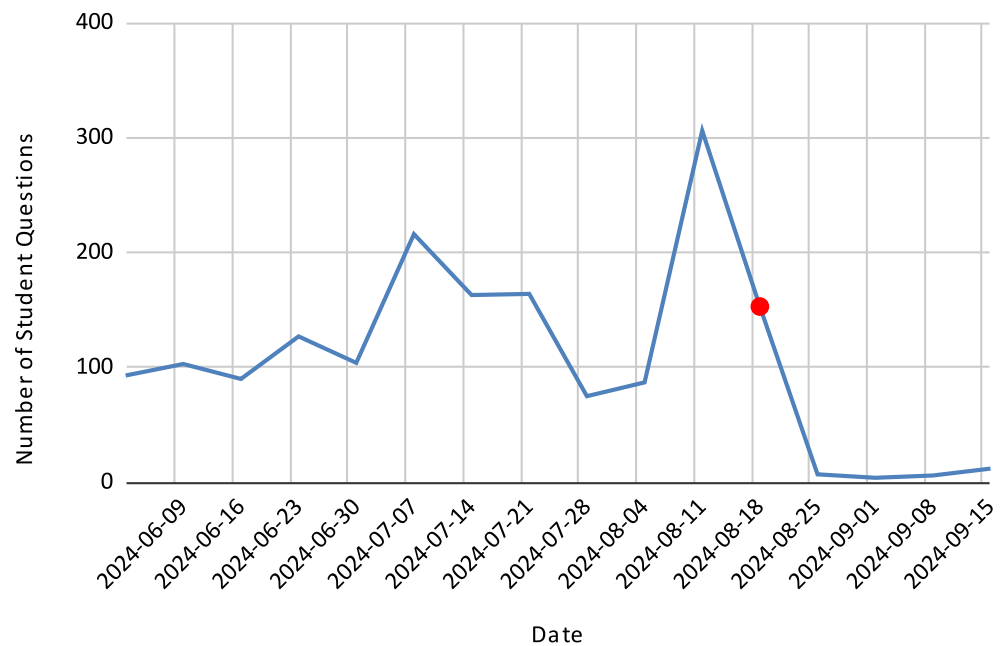


Figure 3. Number of student questions over time.

Query Types Using an LLM-based classification, the queries were categorized into six primary types:

1. **Concept Clarification (50%):** Students sought explanations or definitions of specific terms, concepts, or principles covered in the course.
2. **Problem-Solving Requests (13%):** Students requested assistance with calculations, procedural tasks, or problem-solving approaches.
3. **Follow-Up Questions (12%):** These inquiries built on prior responses, asking for additional details, clarifications, or variations.
4. **Example Requests (12%):** Students asked for illustrative examples to better understand a topic or method.
5. **Non-Actionable Interactions (8%):** Casual exchanges (e.g., “hello,” “thank you”) or unrelated queries.
6. **Administrative Queries (5%):** Course logistics, such as lesson schedules or overall structure (We note that administrative information is uploaded to the system when it is provided in the form of accompanying documents (e.g., PDF) or in the lecture transcript. Either way it is included in the RAG and retrieved from there).

This diversity highlights the breadth of student needs addressed by TAUDT, ranging from conceptual understanding to practical problem solving. It is noteworthy that the vast majority of queries addressed the various aspects of the actual learning process as opposed to course administration.

System Performance The system answered 78.69% of submitted queries successfully, with the success rate improving steadily over the semester, reaching 90% during the critical exam preparation period (Figure 4). The average response time was 11.63 s. The remaining 21.31% of queries failed for various reasons. An analysis of failure types revealed the following breakdown:

1. **Failed to Match Quotes in Response Generation Phase (85.39% of errors):** This error occurs when the *response formatter* fails to correctly align the generated answer with specific references from the course materials. Namely, it cannot

match the quotes in the answer to the original retrieved chunks. (This may happen due to a lack of direct quotes, formatting inconsistencies, mismatches between the generated text and retrieved sources, etc.)

2. Failed Processing Sources (12.13%): Issues arose when source documents could not be processed, typically due to parsing or mapping challenges.
3. Failed to Extract Content in Curation Phase (1.12%): This reflected difficulties in finding relevant material for vague or ambiguous questions.
4. Quote Pipeline Miscellaneous Errors (0.90%): General errors in the content processing pipeline that could not be categorized further.
5. Failed Reordering Quotes (0.45%): Errors occurred when reordering quotes for logical flow or sequence in the response.

These findings have encouraged iterative improvements to the system, particularly around content retrieval and error-handling mechanisms.

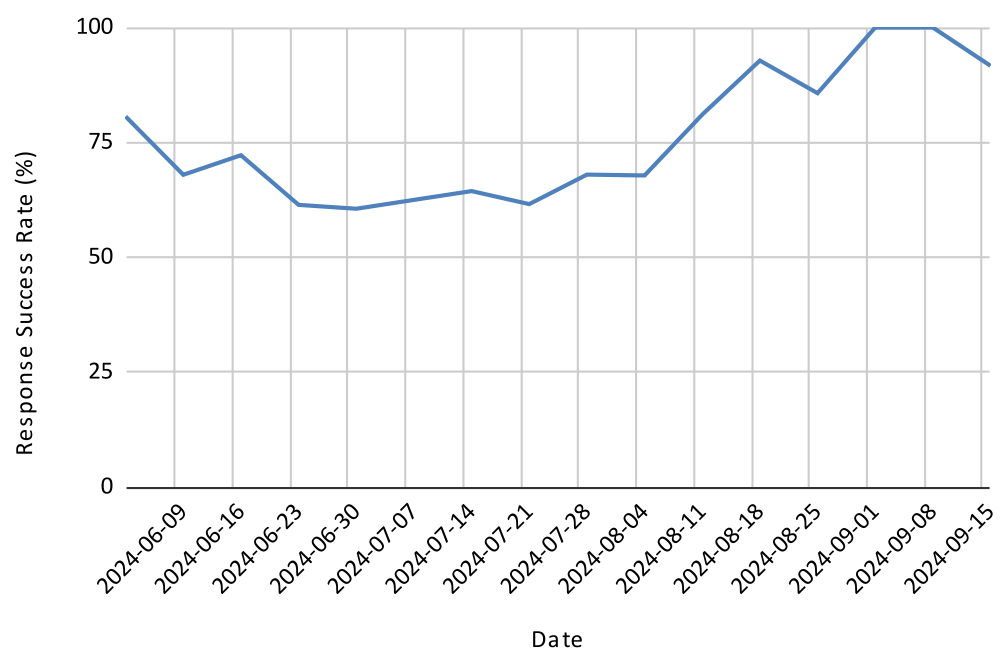


Figure 4. Percentage of successful responses over time.

Student Feedback Of the feedback received, 62.86% was positive, with an upward trend in satisfaction as new features were introduced, reflecting general satisfaction with the tool. Detailed comments were submitted in 78.20% of interactions, offering constructive input for refinement. Key themes included the following:

1. Content Relevance: Students occasionally noted that answers did not fully address their queries or were unrelated to the topic.
2. Completeness: Some students requested more detailed explanations or additional context.
3. Technical Issues: Reports of system errors, such as timeouts or inability to generate responses, were noted.
4. Usability Suggestions: Students proposed features to enhance the user experience, such as improved formatting (e.g., breaking complex answers into bullet points) or additional system functionality.

Overall, the pilot study underscored TAUDT's potential to enhance student engagement and learning outcomes. Based on the positive outcomes of this pilot, the system

is now being deployed in 30 additional courses across the university. This expansion will provide further insights into TAUDT's scalability and adaptability across a variety of academic disciplines, paving the way for its continued refinement and evolution as a robust and versatile educational tool.

7. Discussion and Conclusions

The emergence of commercial LLMs, such as ChatGPT, Gemini, Claude, and others, presents an unprecedented opportunity to enhance higher education by constructing AI-driven systems enabling offline interactive teaching and learning environments. The potential of such systems, particularly in reaching wider audiences, underscores their transformative role in democratizing access to quality education.

We argue that these systems are required to have three important features, whose implementation is demonstrated in the TAUDT platform.

The first is ensuring the correctness and reliability of the platform when responding to student queries. We achieve that by anchoring all responses in validated course materials, ensuring accuracy and pedagogical alignment. This approach provides students with mechanisms to verify information through direct references to formal content, fostering trust and educational rigor.

The second required feature is seamless integration into existing academic workflows while requiring no significant technical expertise from the teaching staff. The designed system must minimize barriers to adoption and support its scalability across institutions.

Thirdly, the rapid advancements in LLM technologies dictate the need in modular architecture that enables seamless integration of the best available LLM at every stage. This approach ensures that the platform remains state-of-the-art while maintaining system stability and adaptability to technological progress.

We maintain that these core features are mandatory for achieving a robust and scalable solution, addressing critical challenges in higher education while leveraging the transformative potential of LLMs. While the current emphasis in TAUDT is on these key requirements, several areas remain open for further exploration:

- **System Logs for Insights:** The potential of leveraging system logs extends beyond improving the platform itself. By analyzing these logs, educators could identify common student challenges and refine their teaching strategies accordingly [33].
- **Human–Platform Interactions:** Future work should focus on optimizing the interactions between students, teachers, and the TAUDT platform. Clearly defining the roles of human participants in this digital ecosystem is crucial for maintaining balance and maximizing benefits [12].
- **Promoting Interactive Learning:** By framing the TAUDT platform not only as a tool for solving problems but also as a means to actively involve students in the learning process, its impact could be further amplified [34].

The use of external large language models (LLMs) raises ethical considerations, particularly regarding data security and content generation biases. The platform upholds student privacy through strict anonymization of interaction logs, preventing any personally identifiable information from being stored or analyzed. TAUDT minimizes reliance on the knowledge base of external LLMs by employing a retrieval-augmented generation (RAG) approach, ensuring that course-specific knowledge remains the foundation of responses. However, as the system expands into disciplines where discussions involve politically sensitive topics, gender, race, or sociological contexts, additional safeguards may be required, a subject that is currently being evaluated for future system versions. Ongoing feedback mechanisms allow for continuous ethical oversight, addressing emerging concerns. The research and deployment of TAUDT were conducted in compliance with Tel Aviv Univer-

sity's Institutional Review Board guidelines, and informed consent was obtained from all participating students.

Following the points outlined above, the TAUDT platform represents a promising direction for promoting independent learning across various frameworks, including lifelong education [35] and beyond. It demonstrates the transformative potential of LLMs in higher education and lays a robust foundation for scalable and impactful educational innovation. As the platform evolves and its application extends to additional disciplines and learning contexts, TAUDT and similar platforms are poised to play a key role in shaping the future of higher education.

Author Contributions: M.S. and T.M. conceived and initiated the project. They also functioned as scientific supervisors. H.R., Y.F., R.N., Y.R.-m. and R.S. jointly developed the TAUDT platform and accompanied its deployment and evaluation. M.J.L. helped with developing automated testing and evaluation capabilities. This paper was written jointly by H.R., M.S. and T.M. All authors have read and approved the final manuscript.

Funding: The work reported in this paper was funded Tel Aviv University.

Institutional Review Board Statement: Ethical review and approval were waived for this study, since an analysis of an anonymous cohort derived from pedagogical project is not a human subject research. According to Israeli regulations, this study does not require IRB approval and was approved by the head of Tel Aviv University Ethics Committee.

Informed Consent Statement: Informed consent was obtained from all subjects involved in this study.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Acknowledgments: We extend our gratitude to Jonathan Ostrometzky for his enthusiastic introduction of TAUDT to his class and for providing insightful feedback. We also thank Anat Cohen for her valuable comments on the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial intelligence
TAUDT	Tel Aviv University Digital Tutor
LLM	Large language model
RAG	Retrieval-augmented generation
API	Application programming interface
UI	User interface
NLP	Natural language processing

References

1. Knobloch, J.; Kaltenbach, J.; Bruegge, B. Increasing student engagement in higher education using a context-aware Q&A teaching framework. In Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training, Gothenburg, Sweden, 30 May–1 June 2018; pp. 136–145.
2. Zylich, B.; Viola, A.; Toggerson, B.; Al-Hariri, L.; Lan, A. Exploring automated question answering methods for teaching assistance. In Proceedings of the Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, 6–10 July 2020; Proceedings, Part I 21; Springer: Berlin/Heidelberg, Germany, 2020; pp. 610–622.
3. Sajja, R.; Sermet, Y.; Cwiertny, D.; Demir, I. Platform-independent and curriculum-oriented intelligent assistant for higher education. *Int. J. Educ. Technol. High. Educ.* **2023**, *20*, 42. [\[CrossRef\]](#)
4. Liu, X.; Pankiewicz, M.; Gupta, T.; Huang, Z.; Baker, R.S. A Step Towards Adaptive Online Learning: Exploring the Role of GPT as Virtual Teaching Assistants in Online Education. *Preprint* **2024**. [\[CrossRef\]](#)

5. Sajja, R.; Sermet, Y.; Cikmaz, M.; Cwiertny, D.; Demir, I. Artificial intelligence-enabled intelligent assistant for personalized and adaptive learning in higher education. *Information* **2024**, *15*, 596. [CrossRef]
6. Sumanth, N.S.; Priya, S.V.; Sankari, M.; Kamatchi, K. AI-Enhanced Learning Assistant Platform. In Proceedings of the IEEE 2024 International Conference on Inventive Computation Technologies (ICICT), Lalitpur, Nepal, 24–26 April 2024; pp. 846–852.
7. Google Developers Blog. How It's Made: Exploring AI x Learning Through Shiffbot, an AI Experiment Powered by the Gemini API. 2024. Available online: <https://app.daily.dev/posts/how-it-s-made---exploring-ai-x-learning-through-shiffbot-an-ai-experiment-powered-by-the-gemini-api-8z3etgmsk> (accessed on 24 November 2024).
8. Poe. Poe-Fast, Helpful AI Chat. 2024. Available online: <https://poe.com/> (accessed on 24 November 2024).
9. OpenAI. Introducing GPTs. 2024. Available online: https://openai.com/index/introducing-gpts/?utm_source=chatgpt.com (accessed on 24 November 2024).
10. CustomGPT. CustomGPT: Build Your Own AI-Powered Chatbot. 2024. Available online: https://customgpt.ai/?fpr=justin10&gad_source=1&gclid=CjwKCAiA9IC6BhA3EiwAsbltOHu_zAqo9IYsS6Bok9tgNKVM-7WszdypEsoIphVAtmBf6y7T_p5-ChoCooUQAvD_BwE (accessed on 24 November 2024).
11. Azamfirei, R.; Kudchadkar, S.R.; Fackler, J. Large language models and the perils of their hallucinations. *Crit. Care* **2023**, *27*, 120. [CrossRef] [PubMed]
12. Te'eni, D.; Yahav, I.; Zagalsky, A.; Schwartz, D.; Silverman, G.; Cohen, D.; Mann, Y.; Lewinsky, D. Reciprocal human-machine learning: A theory and an instantiation for the case of message classification. *Manag. Sci.* **2023**, 1–26. [CrossRef]
13. Rehman, A.U.; Mahmood, A.; Bashir, S.; Iqbal, M. Technophobia as a Technology Inhibitor for Digital Learning in Education: A Systematic Literature Review. *J. Educ. Online* **2024**, *21*, n2. [CrossRef]
14. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459–9474. [CrossRef]
15. Perkins, M. Academic Integrity considerations of AI Large Language Models in the post-pandemic era: ChatGPT and beyond. *J. Univ. Teach. Learn. Pract.* **2023**, *20*, 1–26. [CrossRef]
16. Liu, H.; Liu, C.; Belkin, N.J. Investigation of users' knowledge change process in learning-related search tasks. *Proc. Assoc. Inf. Sci. Technol.* **2019**, *56*, 166–175. [CrossRef]
17. Akgun, S.; Greenhow, C. Artificial intelligence in education: Addressing ethical challenges in K-12 settings. *AI Ethics* **2022**, *2*, 431–440. [CrossRef] [PubMed]
18. Ewing, G.; Demir, I. An ethical decision-making framework with serious gaming: A smart water case study on flooding. *J. Hydroinform.* **2021**, *23*, 466–482. [CrossRef]
19. Neumann, M.; Rauschenberger, M.; Schön, E.M. "We need to talk about ChatGPT": The future of AI and higher education. In Proceedings of the 2023 IEEE/ACM 5th International Workshop on Software Engineering Education for the Next Generation (SEENG), Melbourne, Australia, 16 May 2023; pp. 29–32.
20. Lee, H. The rise of ChatGPT: Exploring its potential in medical education. *Anat. Sci. Educ.* **2024**, *17*, 926–931. [CrossRef] [PubMed]
21. Huang, J.; Saleh, S.; Liu, Y. A review on artificial intelligence in education. *Acad. J. Interdiscip. Stud.* **2021**, *10*, 206. [CrossRef]
22. Crompton, H.; Song, D. The potential of artificial intelligence in higher education. *Rev. Virtual Univ. Católica Del Norte* **2021**, *62*, 1–4. [CrossRef]
23. Jensen, L.X.; Buhl, A.; Sharma, A.; Bearman, M. Generative AI and higher education: A review of claims from the first months of ChatGPT. *High. Educ.* **2024**, 1–17. [CrossRef]
24. Essel, H.B.; Vlachopoulos, D.; Tachie-Menson, A.; Johnson, E.E.; Baah, P.K. The impact of a virtual teaching assistant (chatbot) on students' learning in Ghanaian higher education. *Int. J. Educ. Technol. High. Educ.* **2022**, *19*, 57. [CrossRef]
25. Tack, A.; Piech, C. The AI Teacher Test: Measuring the Pedagogical Ability of Blender and GPT-3 in Educational Dialogues. In Proceedings of the 15th International Conference on Educational Data Mining, Durham, UK, 24–27 July 2022; pp. 522–529. [CrossRef]
26. Kasneci, E.; Seßler, K.; Küchemann, S.; Bannert, M.; Dementieva, D.; Fischer, F.; Gasser, U.; Groh, G.; Günemann, S.; Hüllermeier, E.; et al. ChatGPT for good? On opportunities and challenges of large language models for education. *Learn. Individ. Differ.* **2023**, *103*, 102274. [CrossRef]
27. Caffagni, D.; Cocchi, F.; Barsellotti, L.; Moratelli, N.; Sarto, S.; Baraldi, L.; Baraldi, L.; Cornia, M.; Cucchiara, R. The Revolution of Multimodal Large Language Models: A Survey. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, 11–16 August 2024; pp. 13590–13618. [CrossRef]
28. Wikipedia Contributors. LangChain. 2024. Available online: https://en.wikipedia.org/wiki/LangChain?utm_source=chatgpt.com (accessed on 28 November 2024).
29. TAUDT-Team. TAUDT Tutorial. 2025. Available online: <https://www.youtube.com/watch?v=IC2bCnEv5X0> (accessed on 13 March 2025).
30. Wikipedia Contributors. Word Embedding—Wikipedia. The Free Encyclopedia. 2024. Available online: https://en.wikipedia.org/wiki/Word_embedding (accessed on 27 December 2024).

31. Es, S.; James, J.; Espinosa-Anke, L.; Schockaert, S. RAGAS: Evaluation Framework for Retrieval-Augmented Generation Systems. 2023. Available online: <https://docs.ragas.io/en/stable/> (accessed on 13 March 2025).
32. Biases, W. Weights & Biases: Machine Learning Experiment Tracking. 2020. Available online: <https://wandb.ai> (accessed on 3 December 2024).
33. Hase, A.; Kuhl, P. Teachers' use of data from digital learning platforms for instructional design: A systematic review. *Educ. Technol. Res. Dev.* **2024**, *72*, 1925–1945. [[CrossRef](#)]
34. Doherty, K.; Doherty, G. Engagement in HCI: Conception, Theory and Measurement. *ACM Comput. Surv.* **2019**, *51*, 99. [[CrossRef](#)]
35. Field, J. *Lifelong Learning and the New Educational Order*; Trentham Books: Stoke-on-Trent, UK, 2000.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.