Computer Standards & Interfaces 92 (2025) 103940

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

LAMB: An open-source software framework to create artificial intelligence assistants deployed and integrated into learning management systems

Marc Alier[a], Juanan Pereira[b], Francisco José García-Peñalvo[c,*], Maria Jose Casañ[a], Jose Cabré[a]

[a] UPC Universitat Politècnica de Catalunya, Barcelona Tech, Barcelona, Spain
[b] Computer Science Faculty, UPV/EHU, San Sebastián, Spain
[c] Computer Science Department, Research Institute for Educational Sciences (IUCE), Universidad de Salamanca, Salamanca, Spain

ARTICLE INFO

Keywords:
Generative artificial intelligence
Education domain
Learning assistant
Retrieval-Augmented Generation (RAG)
Large language model (LLM)
IMS learning tools interoperability (LTI)

ABSTRACT

This paper presents LAMB (Learning Assistant Manager and Builder), an innovative open-source software framework designed to create AI-powered Learning Assistants tailored for integration into learning management systems. LAMB addresses critical gaps in existing educational AI solutions by providing a framework specifically designed for the unique requirements of the education sector. It introduces novel features, including a modular architecture for seamless integration of AI assistants into existing LMS platforms and an intuitive interface for educators to create custom AI assistants without coding skills. Unlike existing AI tools in education, LAMB provides a comprehensive framework that addresses privacy concerns, ensures alignment with institutional policies, and promotes using authoritative sources. LAMB leverages the capabilities of large language models and associated generative artificial intelligence technologies to create generative intelligent learning assistants that enhance educational experiences by providing personalized learning support based on clear directions and authoritative fonts of information. Key features of LAMB include its modular architecture, which supports prompt engineering, retrieval-augmented generation, and the creation of extensive knowledge bases from diverse educational content, including video sources. The development and deployment of LAMB were iteratively refined using a minimum viable product approach, exemplified by the learning assistant: "Macroeconomics Study Coach," which effectively integrated lecture transcriptions and other course materials to support student inquiries. Initial validations in various educational settings demonstrate the potential that learning assistants created with LAMB have to enhance teaching methodologies, increase student engagement, and provide personalized learning experiences. The system's usability, scalability, security, and interoperability with existing LMS platforms make it a robust solution for integrating artificial intelligence into educational environments. LAMB's open-source nature encourages collaboration and innovation among educators, researchers, and developers, fostering a community dedicated to advancing the role of artificial intelligence in education. This paper outlines the system architecture, implementation details, use cases, and the significant benefits and challenges encountered, offering valuable insights for future developments in artificial intelligence assistants for any sector.

1. Introduction

Over the past few years, Artificial Intelligence (AI) has transformed various industries, with education being no exception. The emergence of large language models (LLMs) and their application in creating AI assistants offers unprecedented opportunities to enhance personalized learning experiences. However, current AI tools in education fall short of providing flexible, customizable, and ethically sound solutions that integrate seamlessly with institutional Learning Management Systems (LMS). This gap creates a pressing need for an AI framework that not only empowers educators to design custom learning assistants but also adheres to institutional privacy policies and fosters a reliable, scalable approach to education.

Existing AI-based educational tools often lack modularity and are

* Corresponding author.
E-mail addresses: marc.alier@upc.edu (M. Alier), juanan.pereira@ehu.eus (J. Pereira), fgarcia@usal.es (F.J. García-Peñalvo), ma.jose.casan@upc.edu (M.J. Casañ), jose.cabre@upc.edu (J. Cabré).

https://doi.org/10.1016/j.csi.2024.103940
Received 26 July 2024; Received in revised form 14 October 2024; Accepted 28 October 2024
Available online 30 October 2024

difficult for non-technical educators to customize or deploy. Moreover, these tools often fail to address critical concerns like privacy, scalability, and seamless integration into existing LMS platforms. While large-scale AI models offer tremendous potential, their current application in education is either too generic or insufficiently aligned with institutional needs. There is a gap in providing educators with easy-to-use, highly customizable, and ethically responsible AI-powered learning assistants.

The goal of this paper is to introduce LAMB (Learning Assistant Manager and Builder), an open-source software framework that enables educators to create AI-powered learning assistants that are fully integrated with LMS platforms. The tasks undertaken include developing a modular architecture, ensuring compliance with educational privacy policies, and offering an intuitive interface for educators to design assistants without the need for programming expertise.

The key contributions of this work are as follows

1. A novel open-source framework, LAMB, designed for educators to create AI-powered learning assistants without requiring programming knowledge.
2. A modular architecture that enables seamless integration with Learning Management Systems (LMS) through the IMS Learning Tools Interoperability (LTI) standard.
3. A robust focus on privacy, ethical AI usage, and adherence to institutional policies to ensure safe deployment in educational environments.
4. A validated case study, the "Macroeconomics Study Coach," which demonstrates the practical application and benefits of the LAMB framework in real-world educational settings.

The remainder of the paper is structured as follows: Section 1 provides background information on LLM-based chatbots, Intelligent Learning Assistants, the concept of Artificial Intelligence Assistants, and outlines the research goals. Section 2 presents a review of related work concerning Generative AI in education. Section 3 details the architecture and system components of LAMB. Section 4 describes the implementation of the Macroeconomics Study Coach on LAMB. Section 5 focuses on the validation of the framework through various case studies. Section 6 analyzes the running costs of LAMB. Section 7 provides a comprehensive discussion of the findings, and finally, Section 8 concludes the paper with a summary of the conclusions and directions for future work.

1.1. Background

In the last two years, Artificial Intelligence (AI) has significantly transformed the knowledge economy, mainly due to the success of ChatGPT [1]. This Generative Artificial Intelligence (GenAI) tool has seen organic adoption and rapid integration across various sectors. It swiftly transitioned AI from a theoretical research field into an immediate and tangible reality. This development carries a sense of urgency and fear of missing out if not addressed and utilized effectively. The shift occurred almost instantaneously, capturing the attention of professionals, companies, and policymakers, who quickly recognized its profound implications and the urgent need for strategic engagement. The new crop of AI tools is supposed to be an opportunity and a challenge [2] for whole industries, with a perceived urgency to be addressed. However, for most sectors, a definitive path forward remains unclear [3].

ChatGPT is just one example of this emerging class of GenAI tools. Throughout 2023 and into early 2024, multiple Large Language Models (LLMs) [4] based on Generative Pre-Trained Transformers [5] have been introduced. These models vary in their offerings, with some provided as services through user interfaces or Application Programming Interfaces (APIs), such as Anthropic's Claude 3 or Google's Gemini. Others have been open-sourced, including datasets, training codes, model weights, and fine-tuning scripts, such as Meta's Llama 3 series and Mistral's Mistral and Mixtral series, among many others. These new models are frequently compared in rankings based on their performance across several metrics and evaluations [6].

With the introduction of chatbots built on top of LLMs, users immediately recognized the emergence of a new class of products. LLMs exhibit the emerging characteristics listed in Table 1.

In the education sector, the implications of these advancements are particularly profound. The introduction of GenAI tools is transforming teaching and learning methodologies. Students, educators, and administrators increasingly leverage these technologies to enhance educational experiences, personalize learning, and improve administrative efficiency. Despite the rapid adoption, a critical need remains for strategic engagement and clear pathways to integrate these tools fully into the educational framework.

The rapid ascent of ChatGPT in the mainstream has piqued the interest of many educators and institutions. A significant number of teachers are now pursuing courses on how to incorporate ChatGPT and similar technologies into their teaching. Their motivations vary but primarily include the potential to automate academic tasks within their disciplines, the possibilities for enhancing student learning experiences, and plagiarism concerns on the assignments [8].

ChatGPT, known for its broad applicability, enriches the educational landscape by offering diverse educational tools to create instructional materials, facilitate inclusive discussions, design quizzes, and provide personalized feedback. It also aids in simplifying complex concepts and can generate coding examples [9]. Additionally, ChatGPT supports research by suggesting innovative ideas and methods, offering insights from past studies, and assisting in statistical analyses, broadening the scope for more inclusive and interconnected research [10]. Furthermore, it aids in writing by critiquing drafts, organizing content, and strengthening arguments [11].

However, the deployment of LLMs in educational settings also introduces considerable risks. The effectiveness of these AI models heavily

Table 1
Main characteristics of LLM-based chatbots.

Characteristic	Description
Advanced Natural Language Processing Abilities	Shows understanding of the user's input, quality of text generated, summarization, and synthesis abilities
Few-shot Learning	LLMs can quickly adapt to new tasks with minimal examples, learning effectively from limited interaction data
Chain-of-Thought Reasoning	LLMs can articulate step-by-step reasoning processes, enhancing their ability to handle complex tasks by mimicking the way humans explain their thought processes
Unforeseen Skills (Emerging Capabilities)	Including programming, arithmetic, correcting misconceptions, and answering exam questions across various domains, improving as the model size increases
Internal Representations	LLMs show internal representations allowing for abstract reasoning beyond mere linguistic structure, such as understanding colors, geography, object properties, and game states
Misconception Differentiation	They can distinguish between misconceptions and facts, showing an internal calibration of truth likelihood
Common-Sense Reasoning	LLMs pass tests like the Winograd Schema Challenge [7], indicating a sophisticated level of common-sense reasoning beyond textual clues
Privacy Concerns	Privacy is an issue in most LLMs provided as a service; for example, only Enterprise or Team ChatGPT customers have guaranteed confidentiality under the terms of service
Multimodality	Since 2023, LLMs have been able to handle inputs and generate outputs in the combined form of images, audio, and even video
Hallucinations	LLM responses are always peppered with the chance of hallucinations or incorrect answers, which the LLM delivers with the same confidence as correct answers

depends on the quality of the input prompts, requiring users to craft their questions carefully to obtain precise and relevant answers [12]. Response quality can vary dramatically depending on the training data's depth and breadth in specific domains, which may result in subpar or incorrect information [13]. The phenomenon of AI "hallucinations," where the model generates convincing yet factually incorrect or irrelevant content, poses significant risks in an academic context where accuracy is critical [14]. Over-reliance on such technology might affect learning or diminish students' creativity and critical thinking skills [15]. Additionally, there is a risk of propagating inherent biases and stereotypes in the training datasets [16]. AI tools cannot substitute for the essential human interactions vital for students' emotional and social growth [17]. Integrating AI into educational frameworks raises ethical issues concerning data ownership and consent, and it increases security concerns, especially since sensitive information might be included in training datasets for future AI models [18–20].

Facing these challenges, educational institutions must carefully navigate the integration of generative AI into their Information Technology (IT) policies. Issues like system compatibility, server reliability, and the development of ethical guidelines must be addressed to harness AI's benefits responsibly. Additionally, adherence to data protection laws, especially in regions with stringent privacy regulations, remains a paramount concern [21–24].

Currently, generative AI tools are extensively used by students, teachers, researchers, etc., but for personal rather than strategic purposes, aimed more at increasing individual work efficiency than improving the quality of the teaching/learning process [25].

Because of its surprising emerging capabilities, an LLM is not designed for any concrete task and has an inherent disruptive quality when introduced in a business process. However, LLMs can also be considered a platform for building specific applications that take advantage of the features of LLMs in a controlled, predictable, and safe manner.

1.2. Artificial intelligence assistants

Since the widespread availability of machine learning technologies in the mid-2010s, particularly with the development of high-level APIs such as TensorFlow (introduced in 2015) and PyTorch (introduced in 2016), a new category of software has emerged: the assistant. An assistant is a conversational software, using text or voice interfaces, that leverages Natural Language Processing (NLP) technology to provide access to a limited number of system features.

When an LLM-based chatbot is combined with a set of features from traditional software, it results in what is described as an AI assistant. Since late 2022, queries for "AI Assistant" on search engines have increased almost tenfold, indicating the emergence of a new kind of AI assistant implemented on top of LLMs.

An example of an AI assistant is the search engine perplexity.ai (https://perplexity.ai), which provides an answer to the user based on the contents of the pages retrieved by the search query and provides references within its answer to the links obtained and uses LLM technology at its best: natural language processing, content analysis, summarization, and content generation to provide a comprehensive response with specific links and citations that enable the verification of the response with authoritative sources.

In 2024, OpenAI launched a new product based on its LLMs named "assistants," moving away from previous attempts to enhance ChatGPT, such as plugins (https://openai.com/index/chatgpt-plugins/) and GPTs (https://openai.com/index/introducing-gpts/). The OpenAI assistants feature a dedicated API and a user interface, providing a distinct approach to building assistants.

A clear picture of what an AI assistant involves can be seen in the OpenAI assistants' documentation: *"The Assistants API allows you to build AI Assistants within your own applications. It provides instructions and can leverage models, tools, and files to respond to user queries effectively"* [26].

Distinctly, an AI assistant differs from an LLM-based chatbot in its purpose and behavior. It utilizes reliable sources of information, which it can accurately cite and link while maintaining the natural language processing, reasoning, and instruction-following capabilities of an LLM.

It is important to note that the quality of an assistant is not merely based on the volume of information baked into the LLM training or the cut-off date of its knowledge updates. Instead, it depends on the assistant's ability to manage reasonably large contexts, retrieve relevant information and make sense of it, follow directions, and structure outputs. Therefore, the ideal LLM for building an AI assistant might not necessarily be the highest-performing model in all areas.

While LLMs play a crucial role in AI assistant (from now on "assistant") development, they are not the only emerging technology involved. As outlined in Table 2, a range of other technologies and disciplines contribute to creating a comprehensive assistant. These complementary components augment the LLM's capabilities, adding new features, predictability, and scope.

Beyond the technologies listed in Table 2, assistants demand strong software engineering skills and best practices, focusing on deployment, scalability, and security. Ensuring the security of an assistant requires a deep understanding of information security principles and addressing

Table 2
Technologies involved in the creation of an AI assistant (in addition to LLMs).

Technology	Description
Retrieval-Augmented Generation (RAG)	RAG combines LLM's ability to generate responses and pull information from external databases or documents, improving the accuracy and relevance of responses. The retrieved data is inserted in the conversation with the LLM, usually called Context, so the LLM can use it to generate an accurate response [27]
Semantic Search in Embeddings Databases	Utilizes embeddings to organize and retrieve information semantically, improving the assistant's ability to understand and respond to queries accurately across different modalities. An embedding is a numerical representation of data that captures its meaning and relationships in a high-dimensional space. An embeddings database can perform a similarity search and retrieve semantic-related objects to a given query. With RAG embeddings, databases confer an AI assistant capable of effectively using large amounts of unstructured information [28]
Very Large Contexts	Modern LLMs have started to allow for extensive contexts. The context is the window of attention of LLM to a conversation. LLMs have displayed the emerging ability to learn new skills from the information and examples provided in a conversation.
Code Interpreters	The LLMs are not designed to perform calculations or complex tasks. While they can fake it with reasonable accuracy, especially the larger models, they are prone to error and hallucination. However, LLMs are increasingly proficient at generating code that can be passed to an interpreter, and they then use the execution output to complete their response
Function Calling	Function calling is a feature introduced by OpenAI in June 2023 [26]. It provides the LLM with the option to respond with an invocation to the function of an API defined in the context. Function calling enables the LLM to interact with external information systems based on user commands
Prompt Engineering	Prompt engineering [29] refers to crafting and optimizing prompts to guide the LLM responses, improving the quality and relevance of generated content
Evaluation Frameworks and Metrics	Utilizing tools and techniques for assessing assistant performance, including monitoring systems, accuracy metrics, and benchmark datasets, to ensure reliability and effectiveness.
Fine Tuning	Refers to retraining the underlying LLM on specific datasets to enhance its performance in particular domains or tasks [30,31]

unique concerns that arise from incorporating an LLM into the technology stack. Particular attention must be paid to mitigating risks such as prompt injection and LLM jailbreaking, among other potential threats [32,33].

1.3. Research goals

The objective is to design, develop, and evaluate a software framework for the education sector that enables educators to create LLM-powered learning applications without coding and deploying them within the context of their learning institutions' processes and information systems.

This type of LLM-powered application is called a Learning Assistant. The software framework is Learning Assistant Manager and Builder, abbreviated as LAMB. The objective is to select and validate the technologies required to create a learning assistant.

The academic setting means that safety and ethical issues will be of the essence. So, a new goal will be added: examining the implications of AI safety in the context of education for the design of LAMB and also what specific requirements and constraints the educational context introduces. The exploration of the concepts of Safe AI in Education was taken from Alier, Garcia-Peñalvo, and Camba [34] and Smart Learning Applications (SLAs) [35], which will have substantial implications for the requirements and design of LAMB:

- A learning assistant must be interoperable with the Learning Management Systems (LMS) and presented to students and teachers as part of the educational institution's software tools and processes.
- Substantial compliance with the educational institution's privacy policies is a legal requirement in the education sector.
- Alignment of the learning assistant with the educational institution culture, using only authoritative sources of information and tools for content generation, providing proper citations and transparency.

2. Related work

Integrating generative AI into learning management systems (LMS) is an emerging field with several applications and ongoing research. While there are no comprehensive studies specifically focused on theoretical frameworks for generative AI applications in LMS, existing work has provided foundational insights into the applications, challenges, and potential benefits of AI-driven educational technologies.

2.1. Theoretical foundations

Generative AI applications in LMS are grounded in key educational theories such as personalized learning, adaptive instruction, constructivism, and connectivism. These theories emphasize tailoring education to individual learners' needs, creating knowledge through active learning and social connections, and adapting educational content dynamically. For instance, Pesovski et al. [36] discuss how generative AI enables customizable learning experiences by automatically generating learning materials tailored to students' preferences and learning styles, which aligns with these educational frameworks.

In a broader context, generative AI supports constructivist learning by providing diverse and personalized content, such as quizzes, lesson plans, and interactive exercises. Virtual tutoring systems powered by AI reflect both personalized learning and connectivist principles, offering individualized support to students and enhancing engagement.

2.2. Applications of generative AI in LMS

Generative AI has been applied across various key areas in education:

Content Generation: AI has been increasingly used to create personalized learning materials, quizzes, and lesson plans, aligning with constructivist educational approaches. For example, the work of Pesovski et al. [36] illustrates how generative AI can create multiple formats of educational content, allowing students to choose between different learning styles and enhancing their engagement and understanding.

Virtual Tutoring: AI-powered virtual tutors offer personalized support, simulate one-on-one instruction, and adapt to individual learning needs dynamically. Noroozi et al. [37] highlight the transformative potential of generative AI in providing personalized feedback, enhancing both learner engagement and educational outcomes

Assessment and Feedback: AI-driven assessments provide competency-based feedback, supporting constructivist notions of building knowledge through experience and reflection. This capability is crucial for delivering real-time, adaptive feedback to students, as Su and Yang [38] noted in their framework for applying generative AI in education.

2.3. Theoretical challenges

Despite the promise of generative AI, several theoretical challenges must be addressed. Wu (2023) discusses potential issues such as cognitive overload, where the rapid information processing capabilities of generative AI models like GPT may overwhelm learners, hindering their ability to engage with the material critically.

Additionally, balancing AI-generated content with human expertise remains a critical concern, as over-reliance on AI could diminish students' critical thinking and social learning experiences.

Furthermore, there are ethical concerns regarding bias and the quality of AI-driven personalization, as Noroozi et al. [37] highlighted**.** The authors emphasize the need for governance frameworks to ensure the responsible use of generative AI in educational contexts, particularly concerning privacy, transparency, and the potential biases inherent in AI models.

Comparative Analysis of LMS with Generative AI Integration

Several LMS platforms have begun integrating generative AI, offering functionalities that enhance the learning experience. A comparative analysis by Arghir [39] demonstrates how platforms such as Moodle, Blackboard, and Tutor LMS support AI-driven tools. Moodle, for example, integrates plugins for dynamic content creation and AI-driven assessments, while Blackboard offers extensions for test creation and content generation. These integrations highlight the increasing role of generative AI in automating educational tasks and facilitating personalized learning.

In contrast, LMS platforms like Canvas rely on third-party integrations to incorporate AI capabilities, limiting their native generative AI functionalities. This variability in AI integration across LMS platforms underscores the need for more standardized, robust AI functionalities to support various educational needs.

2.3.1. How LAMB contributes to solve the key challenges

The integration of generative AI in LMS, while promising, still requires further exploration to develop robust theoretical frameworks that can support the evolving educational landscape. As generative AI continues to grow in capability, several key areas need further investigation:

Personalized Learning and Long-Term Impact: While generative AI has proven effective in delivering personalized learning experiences, more research is needed to evaluate its long-term effects on student engagement and learning outcomes. LAMB directly addresses this challenge by providing an open-source framework that empowers educators to create AI-powered learning assistants tailored to their specific needs. By allowing educators to integrate high-quality, authoritative content into AI-driven learning tools, LAMB contributes to more personalized, contextually relevant educational experiences. The framework's modular design also allows for continuous improvement and adaptation to meet diverse educational goals

Human-AI Collaboration: One of the critical challenges in AI integration within education is determining the optimal balance between AI automation and human oversight. LAMB supports human-AI collaboration by giving educators control over the prompts, knowledge bases, and AI assistant behaviors. This allows for a partnership where AI enhances instruction without replacing the critical role of the educator. By enabling educators to actively design and oversee the AI assistants within their LMS environments, LAMB ensures that human expertise remains central to the learning process while benefiting from AI's capabilities to automate tasks like content generation and student feedback.

Ethical and Governance Frameworks: As Noroozi et al. [37] noted, the ethical challenges surrounding generative AI in education—such as privacy concerns, bias, and the quality of AI-generated content—require robust governance frameworks. LAMB addresses these concerns through built-in safeguards, ensuring alignment with institutional privacy policies and promoting using authoritative sources for content generation. By adhering to strict ethical guidelines, LAMB offers a responsible approach to deploying AI-powered tools in education, ensuring that AI-driven personalization does not compromise learner autonomy or integrity.

Scalability and Integration with Existing Educational Frameworks: The LAMB framework exemplifies how generative AI can be scaled and integrated with existing LMS platforms, addressing the need for interoperable, adaptable educational tools. LAMB's support for the IMS Learning Tools Interoperability (LTI) standard ensures seamless integration with common LMS platforms like Moodle and Blackboard, making it a scalable solution that can be deployed across diverse educational environments. This adaptability is crucial for ensuring that generative AI tools can be used effectively in various contexts, from small classrooms to large institutions.

3. Learning assistant manager and builder (LAMB) system architecture

The LAMB architecture is designed to enable educators to create AI-powered learning assistants (deployed as SLAs) without requiring coding skills, while ensuring seamless integration within institutional Learning Management Systems (LMS). This modular system provides the flexibility, scalability, and security necessary for diverse educational applications, from automating routine queries to enhancing personalized learning experiences.

The architecture was iteratively developed, following an agile software development approach to refine and deploy a Minimum Viable Product (MVP). The Macroeconomics Study Coach was initially created as a case study to address specific student challenges in navigating lengthy lecture materials. This foundational use case informed the broader architectural design of LAMB, demonstrating the system's ability to provide relevant, accurate, and context-aware responses.

3.1. Architecture overview

The LAMB architecture overview will be described by a schema (Fig. 1) and sequence diagrams (Figs. 2-4). The architecture utilizes two external key components:

- **The AI infrastructure**. This provides access to embeddings models and LLM inference. Vendors like OpenAI, Google, Anthropic, or Mistral can provide the AI infrastructure or can provide it by running open-source models like Llama3 or Mixtral in one's infrastructure.
- **The institutional ecosystem**. The LMS is a virtual campus used in learning institutions like Moodle, Sakai, Canvas, or Blackboard.

As illustrated in the architecture schema in Fig. 1, the LAMB system revolves around the LAMB Core, which acts as the central hub, orchestrating communication between the User Interface, the AI Infrastructure, the knowledge base components (RAG Engine, Content Library, Augmentation functions, and Embeddings Database), and the LTI LMS integration. The containers of the LAMB architecture are named LAMB-X (LAMB-Core et al., LAMB LTI Learning Assistant Provider, and LAMB Knowledge Base), and the components within the LAMB Core container are named after their function.

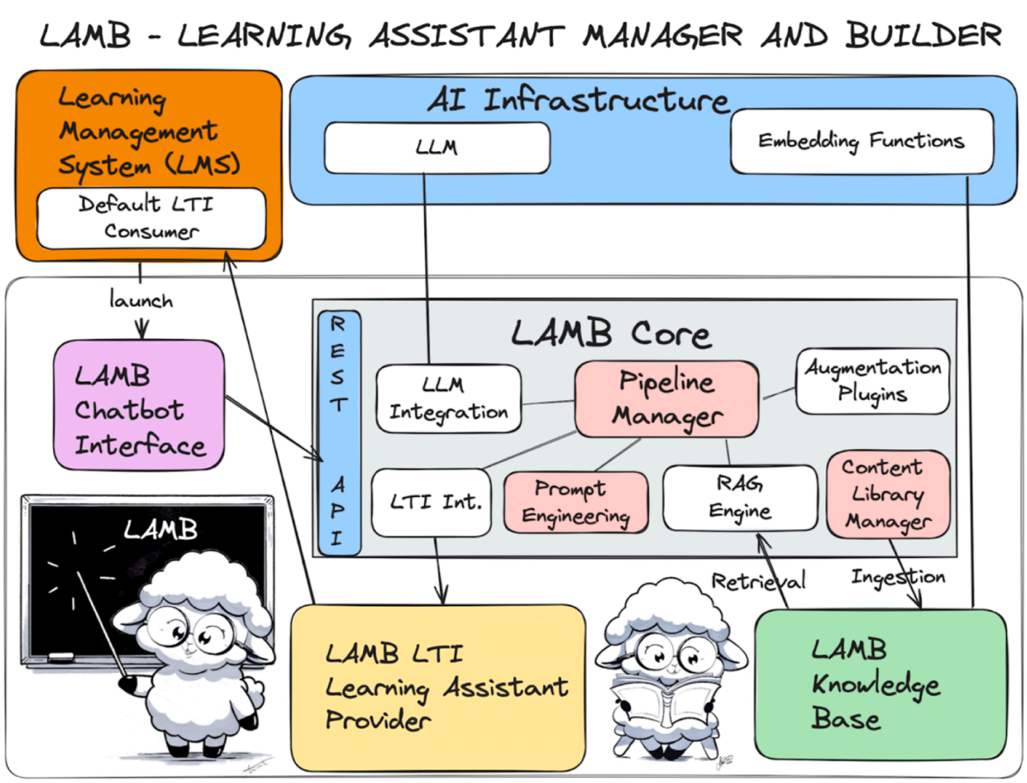

Fig. 1. LAMB architecture schema.

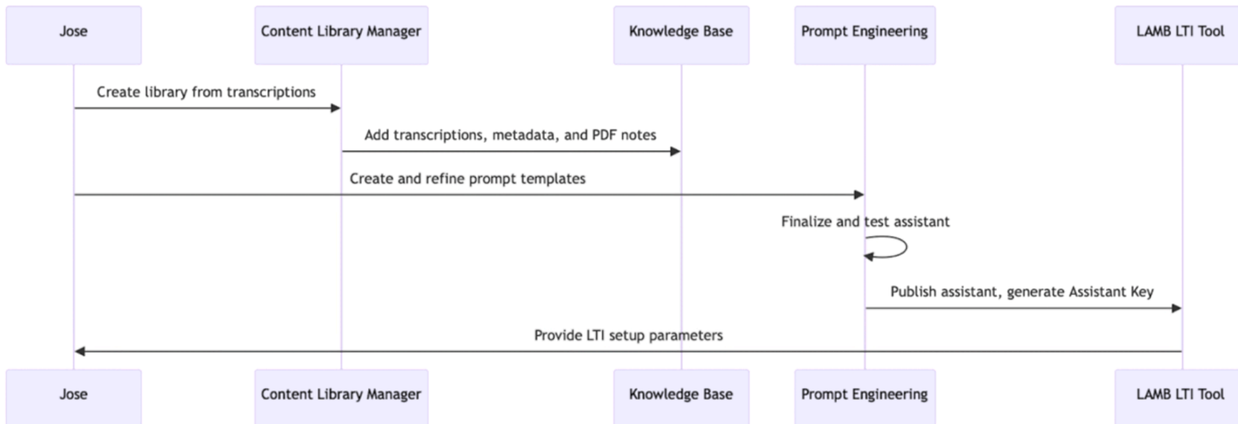

Fig. 2. Sequence diagram of the creation of the "Macroeconomics Study Coach".

It follows a description of the LAMB architecture containers and components.

LAMB Core. It is the main container and interacts with the other containers. It addresses many of the core features of LAMB. Likewise, it is implemented in Node.js with an SQLite database for persistence and follows a modular architecture, as depicted in Fig. 1:

- **Prompt engineering**. This component allows the user to create, test, and manage prompts used to define the behavior of the assistants. The prompts are parameterized for the *RAG Engine* and the *Augmentation Plugins*. Prompt engineering allows testing and evaluating the assistant in an iterative design process. When considered suitable, the user can publish the learning assistant via the *LAMB LTI Integration* by defining an *Assistant Key*.
- **Content library manager**. This component allows users to create and manage learning materials (video transcripts, documents, webpages, etc.) in libraries. The learning assistants will use the contents as authoritative sources of information (ground truth). In an "ingestion process," the Content Library Manager breaks down the sources into relevant chunks of data and metadata, encoded as embeddings, creating *Knowledge Bases*. The chunking strategies are critical to refine the effectiveness of the RAG [40]. Hence, LAMB allows plugins for chunking strategy.
- **REST API.** It provides access to the learning assistants through various endpoints implementing the OpenAI API call structure. The REST API will require the *Assistant Key* to authorize using a learning assistant.
- **Pipeline manager (PM).** This is invoked by the REST API, which provides a learning assistant identifier for the messages received. The PM handles the pipeline of execution of the learning assistant, according to its configuration (done at the *Prompt Engineering* module), using retrieved chunks from the *LAMB Knowledge Base*, the *Augmentation Plugins,* and the *LLM Integration Layer*. Next, the output of the LLM will be processed and returned to the REST API.
- **Retrieval Augmented Generation (RAG) engine.** It retrieves appropriate information from the *Knowledge Base* to produce more accurate and contextually relevant responses from the LLM.
- **Augmentation plugins**. LAMB allows the definition of custom *Augmentation Plugins* to shape the behavior of the learning assistant in the PM.
- **LLM integration layer**. This layer manages the interaction with the chosen LLM, sending prompts and receiving generated responses. It supports multiple LLMs and infrastructure provider configurations. Users can set up their AI infrastructure configurations, including API URLs, API Keys, and models.
- **LTI integration**. It creates an *LTI Learning Tool* for each assistant in the LAMB *LTI Provider*, assigning the proper *Assistant key*. The *LAMB LTI Integration* can also be used by the *Augmentation Plugins* to send grades, participation reports, and feedback to the LMS.
- **Web-based user interface and administration**.

The LAMB Knowledge Base stores semantic collections of content chunks and metadata ingested from the Content Library Manager. It uses embeddings databases like ChromaDB to organize and retrieve information efficiently, supporting the LAMB system's RAG capabilities. This database underpins the system's ability to provide detailed and contextually relevant answers to user queries.

LAMB integrates learning assistants into institutional ecosystems through the **LTI Learning Assistant Provider**, which publishes assistants as LTI-compatible tools. This ensures that educators and students can interact with learning assistants directly within their LMS environment using the institution's existing authentication methods.

The LMS LTI Consumer (our LAMB client) binds the execution of an LTI tool to a specific learning activity in a given LMS course. The protocol allows the LTI tool to get authentication, execution context, roles, and authorizations for the user and report back grades and feedback to the LMS. The administrator of the LMS can also establish privacy policies for the LTI tools and determine whether the tool will receive the student's identification and personal information or be an anonymized façade.

LAMB chatbot interface. This is the frontend through which final users (teachers and students) interact with the learning assistants created with the LAMB framework. The chatbot interface can be a standalone web application or integrated with an existing LMS via the LTI interoperability protocol. The current implementation is based on a customized version of OpenWebUI. Still, since LAMB Core implements the OpenAI syntax, it can be easily customized to various clients, such as Copilot, using a code text editor.

LAMB's architecture prioritizes scalability and security. The system uses Docker containers and microservices, enabling easy deployment, maintenance, and scalability across various educational institutions. Each container operates independently, allowing for isolated updates without disrupting the entire system.

- **Scalability**: The architecture supports multiple instances of AI assistants and large content libraries, ensuring smooth performance even as user demand grows.
- **Security**: The system employs robust security measures, including Assistant Keys that protect access to learning assistants, ensuring compliance with institutional data privacy policies.
- **Extensibility**: The modular structure allows for the easy integration of new LLMs, features, and plugins, enabling the system to evolve alongside advances in AI technologies.

4. Implementing the macroeconomics study coach on LAMB

To demonstrate how the different containers and components of the architecture interact to provide functionality, the learning assistant "Macroeconomics Study Coach" will be used as a detailed example. The macroeconomics course will serve as a case study to illustrate the functionality of the LAMB architecture.

Jose (teacher) has 30 video recordings of lectures of his macroeconomics course. He was acting as Assistant Creator. He will create the learning assistant, Macroeconomics Study Coach, using the LAMB Core. Fig. 2 illustrates the sequence of interactions that occurred during the creation of the learning assistant.

1. With the *Content Library Manager,* he will create a library with all the transcriptions of his lectures. The metadata will include the URL where each video is available for streaming. He will make a *Knowledge Base* with all the videos, plus a PDF file with some lecture notes.
2. He will create prompt templates for his learning assistant with the *Prompt Engineering* module. Then, he will test and refine the prompts and the assistant. After that, the assistant was published. Finally, *LAMB Core* will create an *Assistant Key* and a *LAMB LTI Tool provider* and show Jose the required parameters to set up an *IMS LTI activity* on his LMS (Moodle).
3. Now acting as a teacher (see Fig. 3), Jose will open his macroeconomics course on the Moodle instance of his university, adding an *External Tool Activity* (Macroeconomics Study Coach) with the parameters he received in step 2. He will enter the activity and define the privacy policies and requirements.
4. Jose's students will interact with the "Macroeconomics Study Coach" (see Fig. 4). On the first time they enter, they will have to accept the terms and conditions (if any are set up) and choose their confidentiality settings, like deciding whether they want to share their names with the "Macroeconomics Study Coach" or allow or not their activity with the learning assistant to be anonymously accessed by learning analytics processes. Then, they will be redirected to the *LAMB chatbot user interface*, where they will interact with the learning assistant and ask questions.
5. To answer those questions, the *LAMB Core* accesses the *Knowledge Base* and the LLM infrastructure to send back any response to the chatbot user interface.
6. On the chatbot user interface, the "Macroeconomics Study Coach" will answer the students' questions using the *Knowledge Base* as ground truth and provide excerpts of the sources used and links to the streaming videos of the lectures where the questions are answered.

5. Validation

The initial validation of the LAMB system was the "Macroeconomics Study Coach" learning assistant. This assistant has been subject to intensive evaluations of its performance during several system iterations. These evaluations have informed the iterations in the design of the ingestion processes, the design of the prompt engineering module, the prompting strategies, chunking strategies, the selection of embeddings models, and the workings of the augmentation functions.

The "Macroeconomics Study Coach" has become a valuable demonstration use case, showcasing the use of online videos and PDFs as sources of information for a knowledge base and how the assistant becomes more accurate and aligned in its responses, providing citations for the sources and links to the specific videos and timestamps used or pages of the PDF sources.

Two learning assistants were created after the LAMB system reached a production-stable version.

5.1. The learning assistant is the expert

The first learning assistant was designed as a resource to help students work on a case of evaluation of a technology-driven business project. The students working in groups have to analyze a given case applying the PESTLE (Political, Economic, Social, Technological, Legal, and Environmental) methodology, which requires the students to ask questions about the case from the point of view of several given dimensions (Political, Economic, Social, Technological, Legal, and Environmental) before they write their report [41]. In previous courses, the students had to role-play being such experts with the help of search engines, specific documentation, and even ChatGPT in the fall of 2023.

The professors had identified a weakness in the methodology: the quality of the reports depended heavily on the ability of the students to gather information and role-play as experts in the different dimensions.

To address this issue, the teachers interviewed real experts about the case. These interviews, averaging 30 min, were transcribed and inserted into a knowledge base. Complementary documents - like excerpts from laws, technical documents, and news articles - were also added to the knowledge base.

Then, a learning assistant was created with access to information from experts and documents, and its prompts contained a description of the case and instructions to provide expert information to the students but not perform the analysis for them.

The assistant was tested in two graduate courses on different campuses, with good results. First and foremost, the LAMB system proved stable and production ready. The students could access the learning assistant from the course on the institution's LMS. The system could cope with the workload of a couple of dozen students with concurrent access.

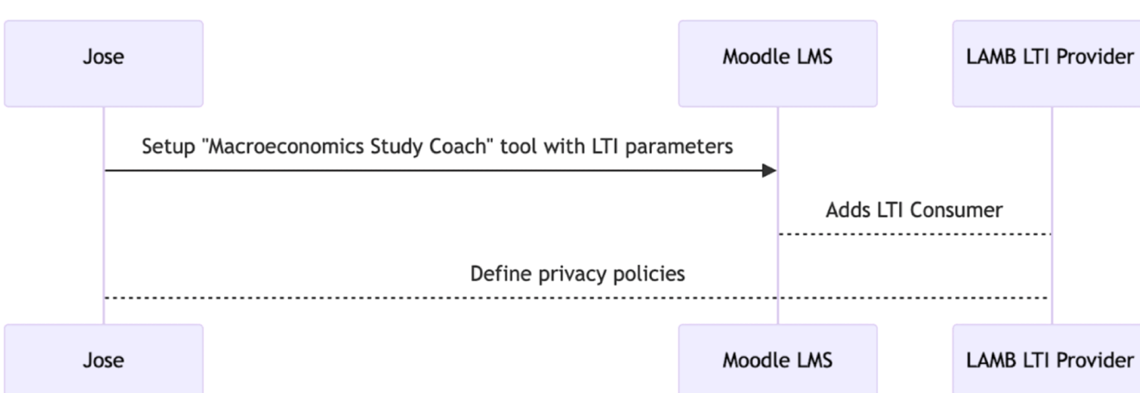

Fig. 3. The teacher creates the LTI Activity that wraps the LAMB Learning Assistant.

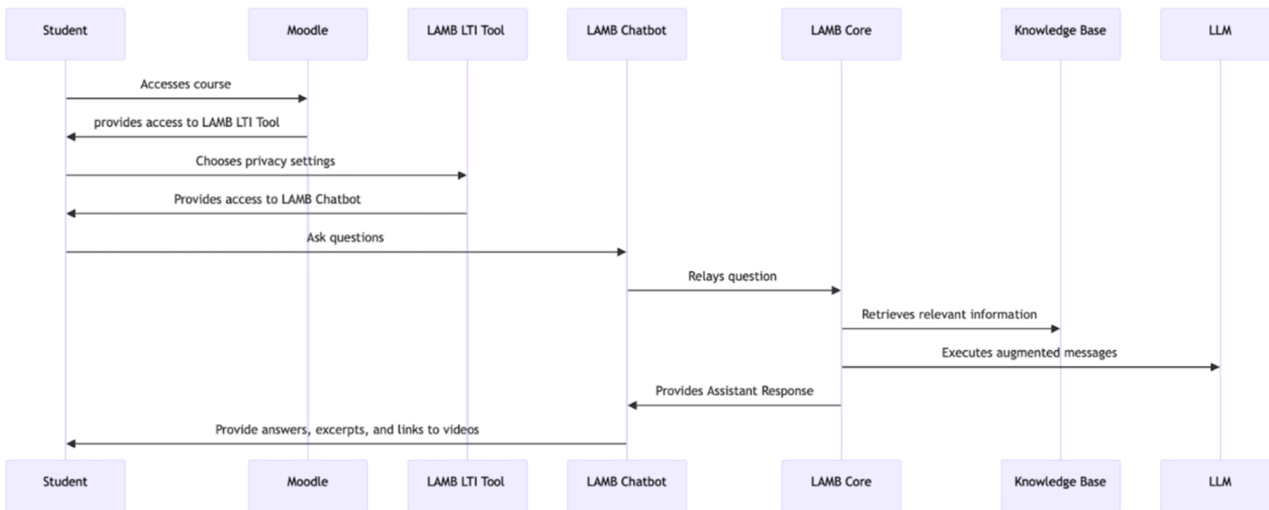

Fig. 4. The student integration with the chatbot via the LMS and the LAMB LTI Tool.

A qualitative analysis of the reports delivered by the students reveals that the reports created with access to the Learning Assistant were noticeably deeper, more detailed, and more nuanced insights in the case analysis than the reports of previous semesters.

The students also answered positively about the usefulness of the learning assistant. The learning assistant was designed to end every answer with proposals for further questions to continue working. The students also considered this feature helpful. See Fig. 5.

5.2. The teacher's assistant

Another assistant was created with a knowledge library containing transcripts of training videos and articles about GenAI in education intended for faculty training. The assistant was made available as complementary course material in faculty training courses in four universities.

The faculty participating in the courses showed great interest in creating their assistants, but their enthusiasm for using them varied greatly depending on the university. A survey on one of the courses (see Fig. 6) shows a good perception of the tool, but only 8 out of 30 participants made it inconclusive.

While student feedback is valuable, we acknowledge the need for more objective validation. To address this, we gathered feedback from 6 professors and teaching assistants who utilized LAMB and the learning assistants in their courses. The responses highlight both the perceived benefits and limitations from an expert perspective.

Positive Feedback:

- **Consistency and Alignment:** Experts emphasized the chatbot's consistent performance, noting its ability to stay within the provided knowledge base and avoid generating irrelevant information. For example, one expert stated: *"At the moment, the chatbot seems very consistent in its responses: it doesn't say anything it shouldn't, and it uses the documentation we have provided, almost always referencing it as a source."*
- **Reduced workload:** Experts believed the learning assistant could handle repetitive student queries about course procedures and deadlines, freeing their time for more complex tasks.
- **Enhanced student learning:** In the case of the expert-based learning assistant, professors observed a noticeable improvement in the depth and quality of student reports compared to previous semesters. They attributed this improvement to the assistant's ability to provide students with access to expert knowledge and facilitate more profound engagement with the case study.
- **Promoting student autonomy:** Some experts saw the potential of learning assistants to encourage students to develop their independence and digital competencies by providing them with a tool for independent learning and exploration.

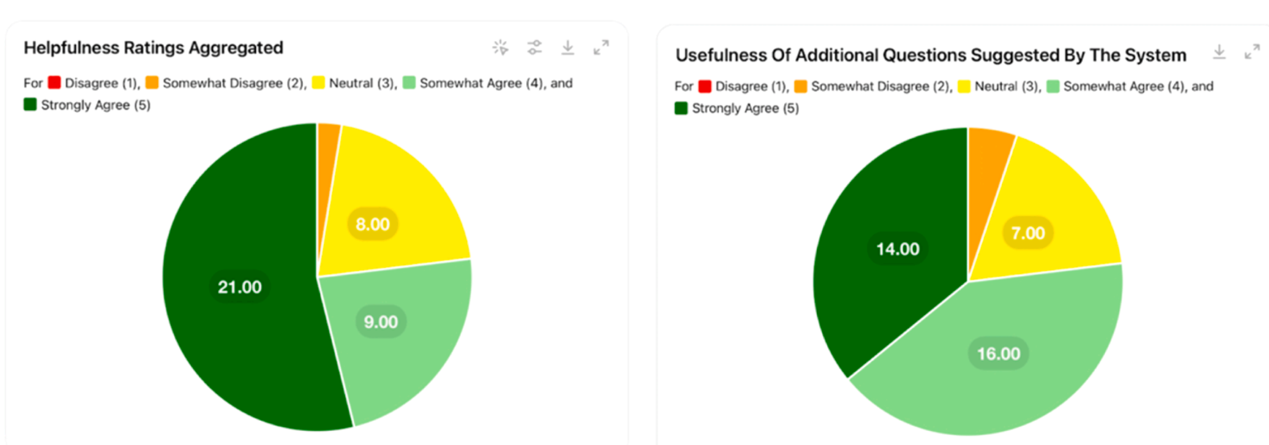

Fig. 5. . Answers to the questions about the helpfulness of the assistant (left) and the usefulness of the additional questions suggested by the system. The survey aggregates 36 answers from students in the two courses participating.

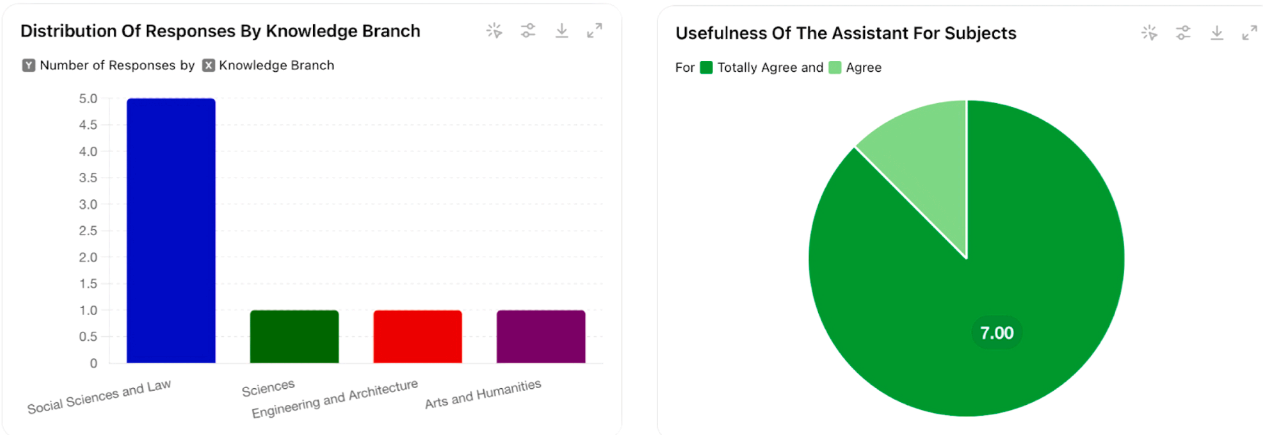

Fig. 6. Survey on the perceived usefulness of a learning assistant as an educational tool.

Challenges and Limitations:

- **Potential misuse and over-reliance:** Concerns were raised about students using the assistant as a shortcut to avoid engaging with core course materials or accepting information uncritically. One expert highlighted the need for clear guidance and training for students to ensure proper use.
- **Integration and usability:** Suggestions were made to improve the integration of the chatbot interface within the LMS to create a more seamless user experience. For example, one expert suggested: *"Another type of interaction: for questions for which it has not been trained and does not have information, you could add an option to make a query to the teacher of the course (e.g., "I cannot answer that question, do you want me to send the query to the teacher of the course? They will answer you through the messaging…")."*
- **Language support:** One expert mentioned the need for the assistant to support minority languages, like Basque or Catalan, particularly for contexts with diverse student populations.

Overall, expert feedback indicates that LAMB and the learning assistants show significant promise as educational tools. However, successful implementation requires careful consideration of the content provided, student training, and integration within existing learning environments. Future iterations of LAMB will address these challenges by providing more robust content curation and management tools, developing user-friendly interfaces, and exploring methods to promote critical thinking and responsible use of AI assistants in education.

5.3. *Learner evaluation and assessment*

While LAMB primarily focuses on enhancing learning through personalized AI-powered assistance, it also contributes indirectly to learner evaluation and assessment. Through its integration with Learning Management Systems (LMS), LAMB records student interactions with learning assistants, such as questions asked and responses provided. This data can help instructors gauge student engagement and identify common confusion or knowledge gaps. For instance, the type and complexity of questions students ask can give educators insights into their understanding of key concepts, enabling targeted interventions where necessary.

LAMB can also facilitate formative assessment by offering personalized feedback based on student queries. This immediate feedback helps learners better grasp concepts in real time, fostering a more profound understanding. Although LAMB currently supports indirect assessment methods, more direct integration of assessment tools and learning analytics will be an area of future development, focusing on providing educators with robust tools for evaluating student learning outcomes in a more structured and measurable way.

6. Running costs of LAMB

LAMB is designed to run on standard servers that can execute Docker containers. The current production server runs without problems on a Xeon (2021) processor with 24GB RAM and 2 × 1 TB SSD (Solid-State Drive); no GPU is used for computing. The AI Infrastructure is externalized to cloud providers like OpenAI, Anthropic, Mistral, Groq, and Google.

The authors have used development laptops with Apple Silicon M processors with +32GB RAM to run the LAMB containers with local execution of LLMs Lama2 13B, Lama3 7B, and Mixtral 8 × 7B models with the ollama.ai software with reasonable speeds of +50 tokens per second.

LAMB is designed to be flexible and cost-effective, offering options for utilizing both commercial and open-source LLMs. The current cost calculations in Tables 3 and 4 are based on commercial LLMs like ChatGPT due to their widespread availability and advanced capabilities.

However, LAMB also supports the integration of open-source LLMs, which can significantly reduce running costs. Open-source models, like Llama 3 or Mistral, can be deployed on local infrastructure or through cloud-based services, offering a more cost-effective alternative for institutions with limited budgets or specific privacy requirements.

The choice between commercial and open-source LLMs involves balancing costs, control, and performance. Commercial models like ChatGPT offer ease of use and higher out-of-the-box performance but come with ongoing token costs. In contrast, open-source models such as Llama 3 or Mistral may offer lower long-term expenses and more significant customization potential. Still, they require more infrastructure and expertise to deploy effectively. Each institution must weigh these factors based on its budget, technical capabilities, and specific application needs.

6.1. *Cost of creation of a learning assistant*

The creation of a Learning Assistant implies two main computing costs:

- **Cost of pre-processing documentation**. For instance, processing the "Macroeconomics Study Coach" video lectures involved several steps. The Whisper 3 model, which can be run locally, transcribes the lectures. Diarization models were employed to separate the professor's speech from the students' questions and responses in the recordings. Additionally, the transcriptions were processed using GPT-4 to enhance the clarity of the text. Notably, the total cost of the GPT-4 API for this purpose was only $3.

Table 3
Running costs of the learning assistant macroeconomics study coach with GPT-3.5.

Prompts per Student	Input Tokens	Output Tokens	Input Cost	Output Cost	Cost per 100 Students	Cost per Student
0.5	123,750	41,250	$0.06	$0.06	**$0.12**	**$0.001**
1	247,500	82,500	$0.12	$0.12	**$0.25**	**$0.002**
2	495,000	165,000	$0.25	$0.25	**$0.50**	**$0.005**
4	990,000	330,000	$0.50	$0.50	**$0.99**	**$0.010**
8	1980,000	660,000	$0.99	$0.99	**$1.98**	**$0.020**
12	2970,000	990,000	$1.49	$1.49	**$2.97**	**$0.030**
20	4950,000	1650,000	$2.48	$2.48	**$4.95**	**$0.050**
50	12,375,000	4125,000	$6.19	$6.19	**$12.38**	**$0.124**

Table 4
Running costs of the learning assistant macroeconomics study coach with GPT-4o.

Prompts per Student	Input Tokens	Output Tokens	Input Cost	Output Cost	Total per 100 Students	Total per Student
0.5	123,750	41,250	$0.62	$0.62	$1.24	**$0.01**
1	247,500	82,500	$1.24	$1.24	$2.48	**$0.02**
2	495,000	165,000	$2.48	$2.48	$4.95	**$0.05**
4	990,000	330,000	$4.95	$4.95	$9.90	**$0.10**
8	1980,000	660,000	$9.90	$9.90	$19.80	**$0.20**
12	2970,000	990,000	$14.85	$14.85	$29.70	**$0.30**
20	4950,000	1650,000	$24.75	$24.75	$49.50	**$0.50**
50	12,375,000	4125,000	$61.88	$61.88	$123.75	**$1.24**

- **Cost of computing embeddings during ingestion**. Embeddings can be locally calculated with small models that do not even require GPU. However, one lesson learned is that the learning assistant benefits significantly from using the best and most expensive embeddings model. For example, the OpenAI embeddings model "text-embedding-3-small" could not recognize technical terms like "stagflation" and would perform poorly if the user switched to the Catalan language on queries. The solution was to use "text-embedings-3-large," which significantly improved the results in English and Spanish and performed well with queries in Catalan. While the first model costs (1/62,500$) = $0.000016/page of text, the second (1/9200$) = $0.0001/page of text. Since ingestion occurs only once for a knowledge base, the cost of embeddings is considered negligible even with the top embedding models. (https://d66z.short.gy/nRVZUo).

Future versions of the LAMB framework are expected to use multimodal embeddings, and costs may vary.

6.2. Cost of running a learning assistant

Running a learning assistant implies querying an LLM via API. As previously stated, an embedding is also calculated at a negligible cost.

The cost of running an LLM query displays a clear pattern of diminishing costs over time. The cost is calculated in dollars per token (approximately 0.75 words). The GPT-3 API in early 2023 had a price of $12/M tokens (M stands for Million tokens) with a max request (or context window) of 4 K tokens, while on June 20, 2024, a better model, GPT-3.5 costs $0.5/M input tokens (24 times cheaper) and $1.5/M for a 16 K context. The latest flagship model of OpenAI, GPT4o (May 2024), costs 5$/M tokens with a context of 128 K tokens (see https://openai.com/api/pricing/).

Google's Gemini 1.5 Pro model allows for 2 M tokens of context [42]. The latest models are multimodal, meaning they can process text, images, sound, and frame-to-frame video inputs [43].

Consider the implications for a learning activity requiring students to use a learning assistant. Two university groups of 50 students totaling 100 participated. They will use a learning assistant during a learning activity that lasts 4 h. The "Macroeconomics Study Coach" logs indicate an average of 2475 tokens per query and an average output of 825 tokens (see Table 5).

Table 5
Average token usage for the Macroeconomics Learning Coach activity as measured on system logs.

Number of Students	100
Input tokens per augmented prompt	2475
Output tokens	825
GPT-3.5 input $/M tokens	0,5
GPT-3.5 output $/M Tokens	1,5
GPT-4o input $/M tokens	5
GPT-4o output $/M Tokens	15

Now, consider the intensity of students' use of learning assistants. In one experiment with groups of five students over four hours, there was an average of four questions per student. However, usage may vary, and different scenarios are presented in Table 3.

Suppose the learning assistant uses a flagship model like GPT4o. In that case, the costs increase significantly, but only up to a point with a fee of $1.24 for students in a group that makes intensive use of the learning assistant (see Table 4).

The costs of running a learning assistant depend significantly on the size of the prompts sent to the LLM, which are determined by its design by the user who creates it. LAMB can allow for the creation of assistants that take advantage of significant contexts. The LAMB user interface offers estimated costs of an assistant to its creator.

LAMB allows for running the LLMs on one's infrastructure. So, running open-source LLMs in one's data center is possible. However, setting up an LLM server for several users is way more complex than running it in batch mode.

7. Discussion

The work presented by the LAMB project and the learning assistants developed and deployed in educational settings have provided several key insights.

First and foremost, GenAI is driving the emergence of a new family of technologies (LLMs, embedding models, embedding databases, etc.) extending far beyond chatbots' simple interactions. These technologies should be viewed as platforms for developing a whole new category of applications.

Traditionally, software development has focused on creating deterministic, highly predictable software, maintaining data integrity, and accurately enforcing business rules. However, this type of software struggles with uncertainty, unstructured information, nuanced data entry, and contextual information. In contrast, the new generation of software can process unstructured information, understand nuances, and manage context. Still, it is nondeterministic and probabilistic, making it prone to hallucinations, with reliability measured statistically.

LAMB is a system that integrates deterministic and probabilistic software, leveraging each other's strengths while mitigating their weaknesses. Essential techniques like prompt engineering, RAG, strategies for ingesting information in knowledge bases, and chunking strategies are being uncovered and perfected as this project develops and this paper is written.

Despite its complex architecture and design, the LAMB system provides a simple user interface to craft a learning assistant and deploy it as a learning tool in an LMS course. LAMB extends the use of GenAI beyond simple chatbot interactions, providing knowledge bases (authoritative sources) and instructions that give the learning assistant a role and purpose in a learning and teaching strategy. The open sourcing of LAMB offers educators the ability to experiment with the design and creation of learning assistants and how they are used to improve teaching and learning.

The core strength of LAMB lies in its ability to leverage the power of LLMs while addressing the specific needs and challenges of educational contexts:

- **Breaking the access gap to AI in Education**. LAMB empowers educators, regardless of their technical expertise, to harness the potential of AI. This democratization of AI technology can bridge the gap between cutting-edge research and practical classroom application.
- **Contextualized learning support**. Unlike generic AI chatbots, like ChatGPT, LAMB-powered assistants are deeply integrated with course content through their knowledge bases. This ensures student interactions are grounded in accurate, relevant, and authoritative information, minimizing the risk of AI hallucinations and promoting a more reliable and trustworthy learning experience.
- **Flexibility and customization**. The modular design of LAMB, particularly the prompt engineering module, enables educators to tailor learning assistants to their specific pedagogical approaches, course objectives, and student needs. This flexibility fosters a more personalized and adaptive learning environment.

The adoption of LAMB in educational settings has the potential to drive significant improvements across multiple dimensions:

- **Personalized learning**. LAMB facilitates the creation of personalized learning paths for individual students. Learning assistants can adapt to different learning styles, paces, and levels of understanding, providing customized support and guidance.
- **Enhanced student engagement**. The interactive nature of learning assistants, coupled with their ability to provide immediate feedback and answer student queries conversationally, fosters greater student engagement with course material. This can lead to a deeper understanding of concepts and increased learning motivation.
- **We are empowering educators**. By automating routine tasks such as answering frequently asked questions and providing essential explanations, LAMB frees up valuable time for educators. This allows them to focus on higher-order teaching activities, such as facilitating discussions, mentoring students, and developing engaging learning materials.
- **Data-driven insights**. LAMB's ability to track and analyze student interactions with learning assistants offers valuable insights into learning patterns, common misconceptions, and areas where students struggle. This data can inform pedagogical strategies, curriculum development, and interventions to improve learning outcomes.

The open-source availability and a simple user interface are insufficient to guarantee successful educational innovation experiences. Just like in the early 2000s, the availability of open-source LMSs like Moodle and Sakai, with intuitive user interfaces and an assortment of default and third-party learning tools, required several years of experimentation and sharing of experiences among educators for best practices to surface and be acknowledged by the community [44]. Online learning communities like Moodle.org, where educators, managers, and developers shared their experiences, research, and code, were of the utmost importance.

Similarly, using wikis [45] or forums as educational tools was not evident in 2003. Designing and using learning assistants for effective educational practice is not obvious and requires research and experimentation. To succeed, LAMB and similar systems must gather a community of learning assistant creators, educators, managers, researchers, and developers. Learning assistants can only reach their potential as a new, helpful category of educational software with such a community.

Based on the successful deployment and validation of LAMB in real-world educational settings, we estimate its Technology Readiness Level (TRL) to be between 6 and 7. The system has moved beyond laboratory testing and has demonstrated its functionality and stability in operational environments. This indicates a high level of maturity and readiness for broader adoption in educational institutions.

Finally, it is essential to acknowledge the limitations of this study. As the integration of generative AI into LMS is a relatively new field, there is a scarcity of comprehensive studies focused on theoretical frameworks for such applications. This limits our ability to directly compare our findings with similar previous work and assess the generalizability of our results across different educational contexts and technologies. Future research should focus on exploring the effectiveness of LAMB and similar frameworks in diverse learning environments and comparing their performance with alternative approaches to AI integration in education.

8. Conclusions and future work

This paper introduces the LAMB project's architecture, design, and implementation. This work successfully leverages LLMs and new technologies and practices, such as embeddings databases, RAG, and prompt engineering, to enable the creation of learning assistants. Combined with standards and interoperability protocols, the learning assistant can be seamlessly deployed on the learning institution's LMS.

The learning assistant is a powerful tool that combines the language processing and generation capabilities of an LLM with specific directions and knowledge retrieval that can be referenced and validated, tailored for specific learning and teaching situations safely and ethically, as stated in the objectives.

The validation of LAMB through real-world implementations, including the "Macroeconomics Study Coach" and the expert-based case study assistant, demonstrates its potential to enhance student engagement and deepen understanding of complex topics. Moreover, the framework's design allows for scalability and adaptability to various educational contexts and subjects.

To further develop and refine LAMB, the following steps are proposed:

1. Expanding LAMB's capabilities to incorporate multimodal inputs and outputs, enhancing the versatility of learning assistants in various educational scenarios.
2. Integrating advanced LLM features such as code interpreters and function calling to extend the functionality of learning assistants.
3. Implementing robust learning analytics integration to provide educators with valuable insights into student learning patterns and outcomes.

4. Integrating Spaced Repetition Based Adaptive E-Learning techniques to enhance long-term retention by personalizing review sessions based on individual student needs and progress.

LAMB aims to make learning more personal and engaging by providing educators with easy-to-use AI tools, enhancing students' learning across various subjects and settings.

Funding

This research is partially funded by the Ministry of Science and Innovation through the AvisSA project grant number (PID2020-118345RB-I00), the Departament de Recerca i Universitats de la Generalitat de Catalunya through the 2021 SGR 01412 research groups award, and the Universidad del País Vasco/Euskal Herriko Unibertsitatea through the contract GIU21/037 under the program "Convocatoria para la Concesión de Ayudas a los Grupos de Investigación en la Universidad del País Vasco/Euskal Herriko Unibertsitatea (2021)".

CRediT authorship contribution statement

Marc Alier: Writing – review & editing, Writing – original draft, Visualization, Supervision, Software, Project administration, Formal analysis, Conceptualization. **Juanan Pereira:** Writing – review & editing, Writing – original draft, Software, Data curation, Conceptualization. **Francisco José García-Peñalvo:** Writing – review & editing, Supervision, Methodology, Data curation, Conceptualization. **Maria Jose Casañ:** Writing – review & editing, Validation, Data curation. **Jose Cabré:** Writing – review & editing, Resources, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data for fine tuning are academic resources

References

[1] OpenAI, "GPT-4 Technical Report," *arXiv,* art. arXiv:2303.08774v4, 2023. doi: 10.48550/arXiv.2303.08774.
[2] F.J. García-Peñalvo, A. Vázquez-Ingelmo, What do we mean by GenAI? A systematic mapping of the evolution, trends, and techniques involved in Generative AI, Int. J. Interact. Multimedia and Artificial Intellig. 8 (4) (2023) 7–16, https://doi.org/10.9781/ijimai.2023.07.006.
[3] M. Alier-Forment, F. Llorens-Largo, Cabalga el Cometa, in: EP-31 Las Alucinaciones de ChatGPT con Faraón Llorens, España, 2023.
[4] W.X. Zhao et al., "A Survey of Large Language Models," *arXiv,* art. arXiv: 2303.18223v13, 2023. doi: 10.48550/arXiv.2303.18223.
[5] A. Vaswani et al., "Attention is all you need," *arXiv,* art. arXiv:1706.03762v7, 2023. doi: 10.48550/arXiv.1706.03762.
[6] W.-L. Chiang et al., "Chatbot Arena: an Open Platform for Evaluating LLMs by Human Preference," *arXiv,* art. arXiv:2403.04132v1, 2024. doi: 10.48550/arXiv.2403.04132.
[7] H. Levesque, E. Davis, L. Morgenstern, The winograd schema challenge, in: Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning, USA, Association for the Advancement of Artificial Intelligence, 2012, pp. 552–561.
[8] F.J. García-Peñalvo, F. Llorens-Largo, J. Vidal, The new reality of education in the face of advances in generative artificial intelligence, RIED: Revista Iberoamericana de Educación a Distancia 27 (1) (2024) 9–39, https://doi.org/10.5944/ried.27.1.37716.
[9] F.J. García-Peñalvo, Generative artificial intelligence and education: an analysis from multiple perspectives, Education in the Knowledge Society 25 (2024), https://doi.org/10.14201/eks.31942 art. e31942.
[10] A. Bahrini et al., "ChatGPT: applications, Opportunities, and Threats," *arXiv,* art. arXiv:2304.09103v1, 2023. doi: 10.48550/arXiv.2304.09103.
[11] J. Crawford, M. Cowling, K.A. Allen, Leadership is needed for ethical ChatGPT: character, assessment, and learning using artificial intelligence (AI), J. Univ Teach. Learn. Practice 20 (3) (2023), https://doi.org/10.53761/1.20.3.02.
[12] L. Henrickson, A. Meroño-Peñuela, Prompting meaning: a hermeneutic approach to optimising prompt engineering with ChatGPT, AI Soc (2023), https://doi.org/10.1007/s00146-023-01752-8. In Press.
[13] A. Nazir, Z. Wang, A comprehensive survey of ChatGPT: advancements, applications, prospects, and challenges, Meta-Radiology 1 (2) (2023), https://doi.org/10.1016/j.metrad.2023.100022 art. 100022.
[14] H.H. Thorp, ChatGPT is fun, but not an author, Science 379 (6630) (2023) 313, https://doi.org/10.1126/science.adg7879.
[15] Y.K. Dwivedi, et al., So what if ChatGPT wrote it?" Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy, Int. J. Inf. Manage. 71 (2023) 102642, https://doi.org/10.1016/j.ijinfomgt.2023.102642 art.
[16] A. Iskender, Holy or Unholy? Interview with Open AI's ChatGPT, European J. Tourism Res. 34 (2023), https://doi.org/10.54055/ejtr.v34i.3169 art. 3414.
[17] E.P.H. Choi, J.J. Lee, M.H. Ho, J.Y.Y. Kwok, K.Y.W. Lok, Chatting or cheating? The impacts of ChatGPT and other artificial intelligence language models on nurse education, Nurse Educ. Today 125 (2023), https://doi.org/10.1016/j.nedt.2023.105796 art. 105796.
[18] D. Gašević, G. Siemens, S. Sadiq, Empowering learners for the age of artificial intelligence, Computers and Education: Artificial Intelligence 4 (2023), https://doi.org/10.1016/j.caeai.2023.100130 art. 100130.
[19] D.R.E. Cotton, P.A. Cotton, J.R. Shipway, Chatting and cheating: ensuring academic integrity in the era of ChatGPT, Innovat. Education Teach. Int. 61 (2) (2024) 228–239, https://doi.org/10.1080/14703297.2023.2190148.
[20] H. Lee, The rise of ChatGPT: exploring its potential in medical education, Anat Sci Educ 17 (5) (2024) 926–931, https://doi.org/10.1002/ase.2270.
[21] F. Llorens-Largo, F.J. García-Peñalvo, La inteligencia artificial en el gobierno universitario, in: Universídad, 2023. Available from: https://bit.ly/46SSxbG.
[22] W.M. Lim, A. Gunasekara, J.L. Pallant, J.I. Pallant, E. Pechenkina, Generative AI and the future of education: ragnarök or reformation? A paradoxical perspective from management educators, Int. J. Manag. Education 21 (2) (2023) 100790, https://doi.org/10.1016/j.ijme.2023.100790 art.
[23] T. Wang, Y. Zhang, S. Qi, R. Zhao, Z. Xia and J. Weng, "Security and Privacy on Generative Data in AIGC: a Survey," *arXiv,* art. arXiv:2309.09435v2, 2023. doi: 10.48550/arXiv.2309.09435.
[24] M. Alier, M.J. Casañ Guerrero, D. Amo, C. Severance, D. Fonseca, Privacy and E-Learning: a Pending Task, Sustainability 13 (16) (2021), https://doi.org/10.3390/su13169206 art. 9206.
[25] S. Triberti, R. Di Fuccio, C. Scuotto, E. Marsico, P. Limone, Better than my professor?" How to develop artificial intelligence tools for higher education, Frontiers in Artificial Intelligence 7 (2024), https://doi.org/10.3389/frai.2024.1329605 art. 1329605.
[26] OpenAI. (2024). *OpenAI Platform*. Available from: https://d66z.short.gy/BiocOD.
[27] P. Lewis, et al., Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, in: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, 2020, pp. 9459–9474. Eds.
[28] Y. Gao et al., "Retrieval-Augmented Generation for Large Language Models: a Survey," *arXiv,* art. arXiv:2312.10997v5, 2024. doi: 10.48550/arXiv.2312.10997.
[29] P. Sahoo, A.K. Singh, S. Saha, V. Jain, S. Mondal and A. Chadha, "A Systematic Survey of Prompt Engineering in Large Language Models: techniques and Applications," *arXiv,* art. arXiv:2402.07927v1, 2024. doi: 10.48550/arXiv.2402.07927.
[30] L. Ouyang et al., "Training language models to follow instructions with human feedback," *arXiv,* art. arXiv:2203.02155v1, 2022. doi: 10.48550/arXiv.2203.02155.
[31] P. Christiano, J. Leike, T.B. Brown, M. Martic, S. Legg and D. Amodei, "Deep reinforcement learning from human preferences," *arXiv,* art. arXiv:1706.03741v4, 2023. doi: 10.48550/arXiv.1706.03741.
[32] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, Y. Zhang, A survey on large language model (LLM) security and privacy: the Good, The Bad, and The Ugly, High-Confidence Computing 4 (2) (2024), https://doi.org/10.1016/j.hcc.2024.100211 art. 100211.
[33] F.W. Liu and C. Hu, "Exploring Vulnerabilities and Protections in Large Language Models: a Survey," *arXiv,* art. arXiv:2406.00240v1, 2024. doi: 10.48550/arXiv.2406.00240.
[34] M. Alier, F.J. García-Peñalvo, J.D. Camba, Generative Artificial Intelligence in Education: from Deceptive to Disruptive, International Journal of Interactive Multimedia and Artificial Intelligence 8 (5) (2024) 5–14, https://doi.org/10.9781/ijimai.2024.02.011.
[35] M. Alier, M.J. Casañ and D. Amo, "Smart Learning Applications: leveraging LLMs for Contextualized and Ethical Educational Technology," in *Proceedings TEEM2023: Eleventh International Conference on Technological Ecosystems for Enhancing Multiculturality (Bragança, Portugal, October 25–27, 2023)*, J. A. de Carvalho Gonçalves, J. L. Sousa de Magalhães Lima, J. P. Coelho, F. J. García-Peñalvo and A. García-Holgado, Eds. Lecture Notes in Educational Technology, pp. 190–199, Singapore: Springer Nature Singapore, 2024. doi: 10.1007/978-981-97-1814-6_18.
[36] I. Pesovski, R. Santos, R. Henriques, V. Trajkovik, Generative AI for customizable learning experiences, Sustainability 16 (7) (2024), https://doi.org/10.3390/su16073034. Available from:.
[37] O. Noroozi, S. Soleimani, M. Farrokhnia, S.K. Banihashem, Generative AI in education: pedagogical, theoretical, and methodological perspectives, Int. J. Technol. Edu. 7 (3) (2024) 373–385, https://doi.org/10.46328/ijte.845.
[38] J. Su, W. Yang, Unlocking the power of ChatGPT: a framework for applying generative AI in education, ECNU Review of Education 6 (3) (2023) 355–366, https://doi.org/10.1177/20965311231168423.
[39] D.C. Arghir, Implementation of learning management systems with generative artificial intelligence functions in the post-pandemic environment, Information

Technologies and Learning Tools 100 (2) (2024) 217–232, https://doi.org/10.33407/itlt.v100i2.5518.
[40] Z. Dong, T. Tang, L. Li and W.X. Zhao, "A Survey on Long Text Modeling with Transformers," *arXiv,* art. arXiv:2302.14502v1, 2023. doi: 10.48550/arXiv.2302.14502.
[41] M.J. Casañ, M. Alier, A. Llorens, A collaborative learning activity to analyze the sustainability of an innovation using PESTLE, Sustainability 13 (16) (2021), https://doi.org/10.3390/su13168756 art. 8756.
[42] Gemini Team Google, "Gemini 1.5: unlocking multimodal understanding across millions of tokens of context," *arXiv,* art. arXiv:2403.05530v2, 2024. doi: 10.48550/arXiv.2403.05530.
[43] S. Yin, S. Bai, J. Huang, X. Chen and Y. Zhang, "A Survey on Multimodal Large Language Models," *arXiv,* art. arXiv:2306.13549v1, 2023. doi: 10.48550/arXiv.2306.13549.
[44] M. Alier, M.J. Casañ, J. Piguillem, R. Lapuente, Creating wiki communities in blended learning environment and the creation of the Moodle New Wiki, Int. J. Social and Humanistic Comput. 1 (3) (2010) 300–313, https://doi.org/10.1504/IJSHC.2010.03269.
[45] M.A. Forment, et al., Didactical patterns for the usage of wikis in educational and learning activities, Int. J. Eng. Educ. 28 (6) (2012) 1347–1352.