

Université du Québec en Outaouais

**Enhancing Learning Interactions in Intelligent Tutoring Systems  
with RAG-Based Dynamic Question Answering**

**A Case Study on Quadratic Equation Solving**

Essay presented by  
**Ahmat Khalid Moustapha**

Supervised by  
**Péricles De Lima Sobreira, Ph.D**

December 2025

## Jury d'évaluation

Présidente du Jury:	Madame Ilham BENYAHIA, PhD, Professeure au Département d'informatique et d'ingénierie de l'UQO
Membre interne:	Ana-Maria Cretu ing., PhD, SMIEEE, Professeure au Département d'informatique et d'ingénierie de l'UQO
Directeur de recherche:	Péricles DE LIMA SOBREIRA, PhD, Professeur associé au Département d'informatique et d'ingénierie de l'UQO

*To my brother Aboulfathi,  
My sister Aïcha  
My nephew Ahmat,  
And my cousins Ali & Mahamat,  
a reminder that family remains constant, even in unfamiliar lands.*

## **Acknowledgments**

I would like to express my sincere gratitude to the community of the Université du Québec en Outaouais (UQO) for making this research possible. My thanks go first to the UQO administration for its steady support, efficiency, and availability. From admissions to day-to-day logistics and technical assistance, your work created an environment in which I could focus fully on my research.

I am equally grateful to my instructors at UQO. Through the quality of their teaching, the rigor of their expectations, and the generosity of their guidance, they nurtured my scientific curiosity, sharpened my critical thinking, and shaped the methods that underpin this work. The courses, thoughtful feedback, and many conversations inside and outside the classroom were crucial in structuring my ideas and deepening my analyses.

My deepest thanks go to my supervisor, Péricles De Lima Sobreira, Ph.D. His patient and demanding mentorship, constant availability, and clear vision were essential at every stage of this project. His probing questions, precise methodological advice, and trust helped me engage with complex issues with confidence. Beyond technical expertise, I benefited from his pedagogy and commitment to ethics, both of which have profoundly influenced the approach presented here.

I would also like to express my sincere appreciation to my managers, Emmanuelle Senez and Francis Martin, as well as to my Director, Mathieu Blais, for granting partial funding for this program through the Government of Québec Employee Study Support Program. Their trust, encouragement, and tangible support made it possible for me to pursue this academic endeavor alongside my professional responsibilities. Their commitment to professional development and continuous learning was instrumental in the successful completion of this work.

I am equally grateful to my colleagues, Latifa and Vincent, as well as to all my colleagues at the Commissaire à la Lutte Contre la Corruption du Québec, for their understanding, encouragement, and collegial support throughout this journey. Working in such a dedicated and principled professional environment has been a constant source of motivation and inspiration.

Finally, I wish to thank everyone who contributed - classmates and colleagues, support staff, librarians, and technical teams. Your help, sometimes discreet but always decisive, made a real difference. This essay is as much the product of your collective efforts as it is of my own.

# Content

<b><u>LIST OF FIGURES.....</u></b>	<b><u>VI</u></b>
<b><u>LIST OF ABBREVIATIONS, INITIALISMS, AND ACRONYMS.....</u></b>	<b><u>VII</u></b>
<b><u>OVERVIEW .....</u></b>	<b><u>IX</u></b>
<b><u>CHAPTER 1. INTRODUCTION .....</u></b>	<b><u>1</u></b>
CONTEXT AND MOTIVATION .....	1
PROBLEM STATEMENT .....	1
RESEARCH OBJECTIVES .....	1
RESEARCH QUESTIONS / HYPOTHESES .....	1
OVERVIEW OF METHODOLOGY .....	2
STRUCTURE OF THIS RESEARCH ESSAY .....	2
<b><u>CHAPTER 2: LITERATURE REVIEW (STATE OF THE ART).....</u></b>	<b><u>4</u></b>
INTELLIGENT TUTORING SYSTEMS (ITS) .....	4
ITS ARCHITECTURE COMPONENTS.....	5
ROLE OF INTERACTION AND FEEDBACK IN ITS.....	7
LEARNING INTERACTION IN ITS.....	8
QUESTION ANSWERING IN EDUCATIONAL CONTEXTS.....	10
RETRIEVAL-AUGMENTED GENERATION (RAG).....	11
QUADRATIC EQUATIONS IN MATHEMATICS EDUCATION .....	17
RESEARCH GAPS AND POSITIONING.....	18
<b><u>CHAPTER 3: METHODOLOGY .....</u></b>	<b><u>20</u></b>
SYSTEM ARCHITECTURE .....	20
INTEGRATION OF RAG INTO AN ITS .....	21
KNOWLEDGE BASE AND RETRIEVAL PROCESS .....	23
ANSWER GENERATION MECHANISM .....	24
CASE STUDY DESIGN.....	25
EVALUATION FRAMEWORK .....	26
<b><u>CHAPTER 4: IMPLEMENTATION AND RESULTS .....</u></b>	<b><u>27</u></b>
PROTOTYPE DESCRIPTION .....	27
INTERACTION SCENARIOS.....	30
PRELIMINARY RESULTS .....	35
ANALYSIS AND INTERPRETATION .....	35
<b><u>CHAPTER 5: ETHICAL, LEGAL, AND SOCIAL IMPLICATIONS.....</u></b>	<b><u>37</u></b>

<b>5.1 AI BIAS AND MISINFORMATION RISKS.....</b>	<b>37</b>
<b>5.2 STUDENT DATA PRIVACY AND SECURITY .....</b>	<b>37</b>
<b>5.3 ACCESSIBILITY AND INCLUSIVITY .....</b>	<b>38</b>
<b>5.4 FUTURE SOCIETAL IMPACTS .....</b>	<b>39</b>
<b><u>CHAPTER 6: DISCUSSION AND FUTURE WORK.....</u></b>	<b><u>41</u></b>
<b>6.1. EDUCATIONAL IMPACT .....</b>	<b>41</b>
<b>6.2. GENERALIZATION POTENTIAL .....</b>	<b>41</b>
<b>6.3. CHALLENGES AND LIMITATIONS .....</b>	<b>42</b>
<b>6.3.1 EXTERNAL PEER REVIEW AND SCHOLARLY FEEDBACK (EDUNINE 2026 CONFERENCE) ...</b>	<b>42</b>
<b>6.4. FUTURE DIRECTIONS.....</b>	<b>44</b>
<b><u>CHAPTER 7: GENERAL CONCLUSION .....</u></b>	<b><u>47</u></b>
<b>7.1 SUMMARY OF FINDINGS.....</b>	<b>47</b>
<b>7.2 CONTRIBUTIONS TO THE FIELD .....</b>	<b>47</b>
<b>7.3 LIMITATIONS .....</b>	<b>48</b>
<b>7.4 PERSPECTIVES FOR FURTHER RESEARCH.....</b>	<b>48</b>
<b>7.5 CLOSING STATEMENT .....</b>	<b>49</b>
<b><u>APPENDICES.....</u></b>	<b><u>50</u></b>
<b>A. CODE SNIPPETS .....</b>	<b>50</b>
<b>B. SCREENSHOT OF DIALOGUES WITH THE ITS:.....</b>	<b>51</b>
<b>C - MERMAID SOURCE CODE OF THE SYSTEM ARCHITECTURE .....</b>	<b>55</b>
<b>D. SYNTHÈSE DE L'ESSAI .....</b>	<b>58</b>
<b>E. CONCLUSIONS DU TRAVAIL.....</b>	<b>60</b>
<b><u>BIBLIOGRAPHY .....</u></b>	<b><u>61</u></b>

## List of figures

Figure 1: Traditional ITS architecture .....	5
Figure 2: The workflow of RAG .....	12
Figure 3: ITS Architecture augmented by a RAG pipeline .....	20
Figure 4: Sub-stages of a RAG pipeline .....	22
Figure 5: Architecture of the developed ITS application.....	30
Figure 6: Scenario 1 – solving a quadratic equation.....	32
Figure 7: Explanation of how to solve a quadratic equation .....	34
Figure 8: Answer of the question What is the discriminant used for?.....	51
Figure 9: Response of the ITS to the question: Why can't I solve with factorization here .....	53

## **List of abbreviations, initialisms, and acronyms**

**AI:** Artificial Intelligence

**ANN:** Approximate Nearest Neighbor

**API:** Application Programming Interface

**BKT:** Bayesian Knowledge Tracing

**CAST:** Center for Applied Special Technology

**DBN:** Dynamic Bayesian Network

**DKT:** Deep Knowledge Tracing

**DSR:** Data Subject Request

**EDUNINE:** IEEE World Conference on Engineering Education

**FAISS:** Facebook AI Similarity Search

**GDPR:** General Data Protection Regulation

**GNN:** Graph Neural Network

**HNSW:** Hierarchical Navigable Small World

**IR:** Information Retrieval

**ISO:** International Organization for Standardization

**ITS:** Intelligent Tutoring System

**KT:** Knowledge Tracing

**LLM:** Large Language Model

**MathML:** Mathematical Markup Language

**MMLA:** Multimodal Learning Analytics

**NIST AI RMF:** National Institute of Standards and Technology - Artificial Intelligence Risk Management Framework

**NLP:** Natural Language Processing

**PDF:** Portable Document Format

**PII:** Personally Identifiable Information

**QA:** Question Answering

**RAG:** Retrieval-Augmented Generation

**REST:** Representational State Transfer

**RL:** Reinforcement Learning

**RLHF:** Reinforcement Learning with Human Feedback

**SEAL:** Self-Evolving Adaptive Learning

**TLS:** Transport Layer Security

**UDL:** Universal Design for Learning

**UI:** User Interface

**UQO:** Université du Québec en Outaouais

**WCAG:** Web Content Accessibility Guidelines

**WAI-ARIA:** Web Accessibility Initiative - Accessible Rich Internet Applications

**XAI:** Explainable Artificial Intelligence

## Overview

This research explores the enhancement of learning interactions within Intelligent Tutoring Systems (ITS) through the integration of dynamic question-answering mechanisms powered by Retrieval-Augmented Generation (RAG). By leveraging RAG, the system can retrieve relevant instructional content at runtime and generate contextually appropriate responses, making the learning experience more adaptive and responsive to individual student needs.

The study focuses on the application of this approach in the domain of solving quadratic equations, a foundational topic in secondary mathematics. Through this case study, the research demonstrates how RAG can be used to guide learners more effectively, provide tailored support, and foster deeper engagement.

The goal is to contribute to the advancement of ITS by increasing the adaptability, responsiveness, and pedagogical effectiveness of question-answering components.

# Chapter 1. Introduction

## Context and Motivation

In recent years, Intelligent Tutoring Systems (ITS) have emerged as one of the leading applications of artificial intelligence in education. These systems emulate human tutors by providing learners with personalized, timely feedback and guidance, enhancing the learning process in domains where individualized attention is not always possible. ITSs are increasingly used to support students in mastering complex topics through adaptive instruction, interactive problem-solving, and immediate feedback.

However, the growing diversity in student backgrounds and learning preferences calls for adaptive, interactive learning environments that move beyond rigid instructional paths. As learners still struggle with complex content, particularly in mathematics, they can benefit most from systems that respond dynamically to their evolving understanding. Indeed, the demand for educational technologies that are both intelligent and responsive continues to grow.

## Problem Statement

Despite their capabilities, many current ITS implementations still rely on static interaction models and traditional question-answering (QA) mechanisms, which fail to be helpful in individual learner contexts, misconceptions, or cognitive needs. These limitations are obstacles to the systems' ability to provide rich, pedagogically meaningful dialogue. Traditional QA modules often produce generic responses based on keyword matching, lacking the depth and adaptability necessary for meaningful learning interactions.

## Research Objectives

This research aims to address these limitations by integrating Retrieval-Augmented Generation (RAG) into ITS architecture to enable dynamic question-answering (QA). The core objective is to enhance learning interactions by enabling the system to retrieve relevant information at runtime. As a case study, we apply this approach to the domain of quadratic equation solving, a critical yet often misunderstood topic in secondary mathematics.

## Research Questions / Hypotheses

This study is guided by the following research questions:

RQ1: How does the integration of RAG improve the adaptability and contextual relevance of QA in ITS?

RQ2: Can RAG-based dynamic QA enhance learner engagement and performance when solving quadratic equations?

RQ3: What are the pedagogical benefits and limitations of using generative QA models within a tutoring system?

From these questions, we hypothesize that:

H1: RAG integration will result in more personalized and accurate responses compared to static QA models.

H2: Learners using the RAG-powered ITS will exhibit higher levels of engagement and better conceptual understanding of quadratic equations.

### **Overview of Methodology**

To test these hypotheses, we design and implement a prototype ITS incorporating a RAG-based QA component. The system is evaluated using a case study approach focused on student interaction during quadratic equation problem-solving. We analyze interaction logs, generated answers, and user feedback to assess system performance across metrics such as accuracy, adaptability, and learner satisfaction.

### **Structure of this Research Essay**

This document is organized as follows:

**Chapter 1** surveys the state of the art in intelligent tutoring systems, learning interactions, educational question answering, retrieval-augmented generation, and the teaching of quadratic equations, and identifies research gaps.

**Chapter 2** details the methodology, covering the system architecture, integration of RAG into the ITS, the knowledge base and retrieval process, the answer-generation mechanism, the case-study design, and the evaluation framework.

**Chapter 3** reports the implementation and results, including the prototype, interaction scenarios, preliminary findings, and their analysis and interpretation.

**Chapter 4** examines ethical, legal, and social implications—AI bias and misinformation, student data privacy and security, accessibility and inclusivity, and broader societal impacts.

**Chapter 5** provides the discussion and future work, addressing educational impact, generalization potential, current challenges and limitations, and future directions.

The **Conclusion/General Conclusion** synthesize key findings and contributions, note limitations, outline perspectives for further research, and offer a closing statement;

## **Chapter 2: Literature Review (State of the Art)**

This chapter reviews the state of the art in intelligent tutoring systems (ITS) - from classic four-component architectures (domain, student, pedagogical, interface) to extensions in analytics and collaboration. It examines how interaction and feedback evolved from static scripts to adaptive dialogues and situates educational QA within this shift, highlighting retrieval-augmented generation (RAG) alongside knowledge tracing, reinforcement learning, multimodal analytics, and knowledge graphs. Because our system targets algebra, we also summarize common difficulties with quadratic equations that motivate guided, adaptive support.

### **Intelligent Tutoring Systems (ITS)**

Intelligent tutoring systems are defined as “computer programs that use AI techniques to provide intelligent tutors that know what they teach, whom they teach, and how to teach” (ALI & JUGAL, 2018). They are also considered as “computer programs that model learners’ psychological states to provide individualized instruction.” (Ma, Adesope, Nesbit, & Liu, 2014). (Cai, Hu, & Graesser, 2019) stated that “ITSs are created to simulate human tutors to help individual students, especially struggling students to improve their learning.” These kinds of tools enabled by artificial intelligence offer a personalized way of learning and personalized feedback.

In addition to these classical definitions, ITS can also be understood as socio-technical systems that integrate computational models with pedagogical theories. For example, constructivism and cognitive apprenticeship influence how ITS designers structure learning tasks and feedback. Beyond simulating a human tutor, ITS aspire to continuously adapt to learners’ trajectories and to create a personalized educational dialogue.

According to (Sharma, Ghorpade, Sahni, & Saluja), a typical ITS has the following four basic components:

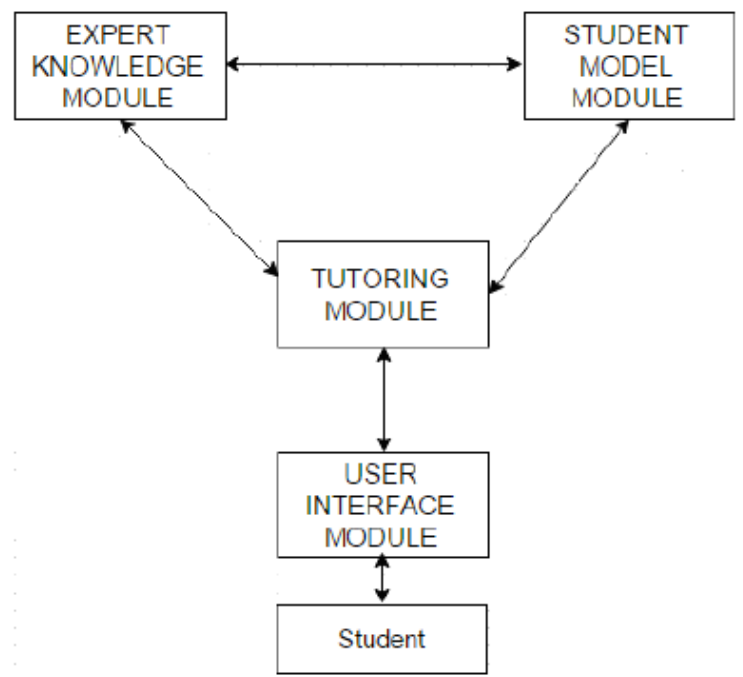
- The Domain model
- The Student model
- The Tutoring/Explanation model, and
- The User interface model.

Recent research suggests extending this list by including additional layers such as a Data Analytics module, which collects log data for learning analytics, and a Collaboration module, which enables peer-to-peer or group learning interactions mediated by the ITS.

Each of these components are important and contain its own characteristics, requirements and evolution. Let's dive into every element briefly.

### ITS architecture components

As mentioned above, the traditional and very well-known architecture of ITS contains 4 elements as they are shown in the figure 1



*Figure 1: Traditional ITS architecture*

### The Domain model

This domain model represents the “what” of the ITS. It contains the contents that are taught, which is collected from experts that have years of experience in the domain or educational materials in the field that the ITS is aimed to teach. According to (Neagu, 2024), the domain model defines rules, concepts, and problem-solving strategies (expert knowledge).

Modern ITSs like ALEKS use this model to ensure the content aligns with established learning goals. For instance, in a math ITS, the domain model would include operations, algebraic rules, and problem templates that reflect the curriculum.

A modern extension of the domain model is the integration of multimodal content such as videos, interactive simulations, and dynamic assessments. These enrich the learning process by going beyond textual representations. Furthermore, domain models are increasingly being linked with large knowledge graphs, allowing ITS to infer relations between concepts and provide broader explanations.

### **The Student model**

The Student Model represents what the learner currently knows, believes, misunderstands, or is capable of. This model is constantly updated as the learner interacts with the ITS. According to (ALI & JUGAL, 2018), the Student Model “represents the knowledge and skills of the student dynamically”.

As highlighted in research by VanLehn (2006) and Brusilovsky & Millán (2007), this model typically includes:

- **Cognitive state** (what the student knows or doesn't)
- **Affective state** (frustration, confidence, engagement)
- **Learning style and preferences.**

Techniques such as Bayesian Knowledge Tracing (BKT) and Dynamic Bayesian Networks (DBN) are commonly used to predict student knowledge mastery over time. Some ITSs now incorporate real-time sensors or logs to capture behavioral data, improving the accuracy of student modeling.

### **The Tutoring/Explanation model**

This component determines the instructional strategies and feedback mechanisms. It decides about content and the way and to present the content. In addition, it decides at what moment to intervene with hints or explanations. (Thai-Nghe & Schmidt-Thieme, 2015) stated that tutoring model “represents teaching processes / strategies. For example, information about when to review, when to present a new topic, and which topic to present is controlled by this module”. Some literature sources call this model as pedagogical model. It “takes the domain and student models as input and selects tutoring strategies, steps, and actions on what the tutor should do next in the exchange with the student to move the student state to more optimal states in the domain.” (Jr., Brawner, Olney, & Mitrovic, 2013).

In other words, current trends emphasize explainable AI in tutoring models, where the ITS not only gives feedback but also explains why an answer is right or wrong. Adaptive scaffolding

techniques can be applied, where the system gradually reduces support as the learner becomes more competent. Moreover, dialogue-based tutoring is gaining ground, where the ITS engages students in Socratic questioning rather than offering direct solutions.

### **The User interface model**

The User Interface Model includes the design and functionality of the ITS's front-end, ensuring effective interaction between the student and the system. (Thai-Nghe & Schmidt-Thieme, 2015) stated that the user Interface model “presents the methods for interacting between the students and the systems. An important problem in this module is how the tasks (materials / learning objects) should be presented to the students in the most effective way”. A well-designed interface enhances usability, engagement, and accessibility. “The user interface interprets the learner’s contributions through various input media (speech, typing, clicking) and produces output in different media (text, diagrams, animations, agents)” (Jr., Brawner, Olney, & Mitrovic, 2013).

In addition, advances in multimodal interfaces allow ITSs to incorporate speech recognition, gesture tracking, and even augmented reality to create immersive environments. Accessibility features such as text-to-speech and screen reader compatibility are also being prioritized to ensure inclusivity. Additionally, gamification elements like badges and progress dashboards enhance motivation and engagement.

Beyond the architectural components, another important aspect to consider is how interaction and feedback operate within an ITS.

### **Role of interaction and feedback in ITS**

Interaction and feedback are fundamental components of Intelligent Tutoring Systems (ITS), playing a crucial role in guiding students through learning, correcting misconceptions, and keeping them engaged. (Kochmar, Vu, Belfer, Gupta, & Pineau, 2020) found that personalized feedback significantly enhances learning outcomes. They deployed a machine learning model in Korbit, an ITS using natural language processing (NLP), and they demonstrated that “the personalized feedback leads to considerable improvement in student learning outcomes and in the subjective evaluation of the feedback”.

Another study highlights how automated feedback based on discourse analysis of student answers can identify gaps in understanding and deliver targeted responses, enhancing student performance (Grenander, et al., 2021).

Beyond corrective feedback, modern ITS provide motivational and metacognitive feedback. For instance, the system might encourage reflection by asking the student to explain their reasoning. Peer-to-peer interaction is also being supported, where the ITS organizes collaborative tasks and moderates discussions. These innovations demonstrate that feedback is not only about accuracy but also about fostering self-regulation, resilience, and lifelong learning skills.

## **Learning Interaction in ITS**

### **2.4.1 Approaches to Student-System Dialogue**

Modern ITSs implement various dialogue strategies to foster effective learning. We didn't find a clear and conventional list of student-system dialogues approaches used in the context of ITS in the literature. However, (Paladines & Ramirez, 2020) mentioned four types of instructional approach, on which "ITSs help students write correct explanations or answers for deep questions". The researchers found that the studied ITS use four main instructional approaches:

- **Generation of Explanations to Justify Solutions:** In this group, "ITS enables the student to actively elaborate explanations and justifications of a previous student input (e.g., a previous student's answer, a student prediction related to a simulation) in a turn-based dialog" (Paladines & Ramirez, 2020).
- **Support for Problem Solving:** the ITS that uses this kind of instructional approach "offer support for problem solving either on demand; when they detect a student's mistake or a student's misconception; or when the solution is not sufficiently complete".
- **Clarify and Direct Procedures:** aimed to support the resolution of problems whose solution cannot be built step by step following a predefined procedure. This is "by giving hints on the next action to be done and immediate feedback on the executed actions"; and by requiring self-explanation from students to explain themselves while solving the problem" (Paladines & Ramirez, 2020).
- **Ask Questions-Answer:** the researchers found one ITS that "gives another chance to the student to correct his/her answer"; and they found another ITS that "directly explains the right answer to the student and continues with the next question".

Beyond these four approaches, more nuanced strategies have emerged in recent ITS research. For example, dialogue-based ITS can employ motivational dialogues, in which the system

encourages persistence after repeated errors, or metacognitive prompts that guide students to reflect on their problem-solving strategies (D’Mello et al., 2012). Some systems also integrate affective dialogue, where the ITS recognizes emotional cues such as frustration or boredom and responds empathetically. Another advanced approach is mixed-initiative dialogue, where both the system and student can take control of the conversation dynamically, fostering a more natural and human-like tutoring experience.

Recent developments also leverage large language models to create open-ended dialogues, allowing ITS to answer unanticipated student questions and to engage in exploratory discussions. These dialogues not only transmit knowledge but also help students co-construct meaning in real time.

#### **2.4.2 Static vs. Dynamic Interaction Models**

In Intelligent Tutoring Systems (ITS), interaction models define how the system communicates and exchanges information with the learner. There are two primary paradigms often contrasted which are: **static** and **dynamic interaction models**.

Static interaction models follow a pre-defined, linear structure of communication. They offer fixed instructional paths, responses, and feedback based on common patterns of student input. These models are simpler to implement and maintain, as they don’t require complex decision-making or adaptation. However, they tend to lack flexibility and personalization. For example, a static model may always present the same hint for a particular question, regardless of the student's past performance or current knowledge state.

In contrast, dynamic interaction models are adaptive and responsive. They tailor the dialogue and instructional behavior to each student’s unique learning progress, preferences, and cognitive or affective states. Dynamic models typically leverage real-time data and AI techniques such as natural language processing, reinforcement learning, or Bayesian modeling to adapt their responses. These models can detect when a student is confused, bored, or disengaged, and modify the interaction accordingly. For instance, they may switch strategies, rephrase explanations, or adjust difficulty levels. Systems like Korbit exemplify dynamic ITS by engaging students in natural and adaptive dialogue.

The choice between static and dynamic interaction models depends on the educational goals, technical constraints, and target learner population. However, there is a growing trend toward dynamic interaction as it provides greater personalization and improves learning outcomes by fostering meaningful student engagement.

In practice, most modern ITS adopt hybrid models that combine static and dynamic features. For instance, an ITS may begin with static hints to ensure curriculum alignment but then transition to dynamic dialogue once it has collected sufficient data about the student's performance. This ensures both pedagogical consistency and personalization (Villegas-Ch et al., 2025).

Future ITS will likely incorporate adaptive ecosystems that combine static, dynamic, and collaborative dialogue models. Such systems will not only personalize instruction but also orchestrate peer-to-peer learning and teacher involvement, creating a holistic, AI-supported learning environment (Kumar et al., 2007).

## **Question Answering in Educational Contexts**

### **2.5.1 Role of QA in Supporting Learning and Critical Thinking**

“Question Answering (QA) can be defined as the task that, given a natural language question, aims at finding one or more concise answers in the form of sentences or phrases.” (Quarteroni, 2010). QA systems play an important role in educational and training technologies. They enable students to retrieve information, verify knowledge, and engage in exploratory learning. In addition, in the context of ITS, QA modules serve also as cognitive scaffolds that promote deeper understanding and critical thinking.

A well-designed QA system encourages active learning. In fact, in well-designed QA, students are prompted to formulate questions, evaluate answers, and integrate new information into their mental models. Therefore, by interacting with a QA system, learners can clarify misconceptions, reinforce understanding, and explore related concepts.

### **2.5.2 Traditional QA vs. Dynamic, Personalized QA**

Traditional QA systems typically rely on keyword matching to return responses from a fixed knowledge base. These systems are limited in their ability to understand context, adapt to the learner's level, or generate nuanced explanations. (Wang, Yang, Chen, & Yu, 2025) stated that “Traditional educational Q&A systems face significant limitations, such as reliance on simple keyword matching, inability to provide contextually rich answers, and challenges in handling complex, procedural queries”. For example, if a student asks, “How do you solve a quadratic equation?”, a traditional QA might provide a generic definition without considering whether the student is just beginning algebra or studying advanced problem-solving techniques.

In contrast, Dynamic, personalized question-answering (QA) systems in education utilize advanced AI techniques to tailor responses based on individual learner profiles. These systems incorporate semantic understanding, student modeling, and dialog management to adapt both the content and complexity of answers. For instance, a personalized QA system may provide a more detailed or simplified explanation depending on the learner's background or follow up with a clarifying question to stimulate deeper thinking. In (Quarteroni, 2010), the author highlights a notable contribution to this field through the development of a personalized QA model designed for open-domain applications, such as those on the Web, rather than being confined to specific domains. The author emphasizes the importance of (semi-)automatically generating a user model by combining information retrieval (IR) techniques for computing personalized relevance with natural language processing (NLP) methods like automatic concept extraction. This approach involves integrating explicit user input - such as personal data and feedback - with implicit cues like text readability, as well as automatically identifying relevant topics and tags. Together, these techniques enable the QA system to deliver more adaptive, context-sensitive responses that align with the learner's current knowledge and learning needs.

Moreover, some dynamic QA systems are integrated into ITS as conversational agents that sustain multi-turn interactions. These agents do not just answer isolated questions, but engage in dialogue to guide learning trajectories, correct errors, and prompt students to justify or revise their reasoning. As described by (Cai, Hu, & Graesser, 2019), the AutoTutor system exemplifies this approach by leading students through structured conversations aimed at building complete answers or problem solutions. During each turn, AutoTutor analyzes the learner's response, identifies missing elements, and responds with targeted hint or prompt questions - depending on whether a full sentence or just a word/phrase is needed. If a misconception is detected, the system intervenes to correct it before continuing the instructional exchange. This dynamic, adaptive interaction process reflects how conversational QA systems in ITS support personalized, reflective learning through dialogue.

## **Retrieval-Augmented Generation (RAG)**

### **2.6.1 Core Architecture: combining retrieval and generation**

Combining Retrieval and Generation Retrieval-Augmented Generation (RAG) is a hybrid neural architecture that integrates two major components in natural language processing (NLP): a retrieval mechanism and a generative language model. The RAG architecture operates in two

phases. First, it retrieves relevant documents or text passages from a large external knowledge base using a retrieval system - typically based on dense vector representations. Then, the retrieved information is fed into a generative model (such as BERT2GPT or T5) that synthesizes a coherent and contextually relevant answer based on the question and the retrieved context. This structure enables RAG to combine the factual richness of external knowledge sources with the flexibility and fluency of generative models. As such, (Lewis, et al., 2021) stated that “RAG combines the generation flexibility of the “closed-book” (parametric only) approaches and the performance of "open-book" retrieval-based approaches”.

### 2.6.2 Benefits for QA Systems:

RAG offers several key advantages in the context of educational question-answering (QA) systems. One of the primary benefits is contextual accuracy. By integrating retrieval, RAG systems ensure that generated answers are grounded in relevant, up-to-date information drawn from textbooks, lecture notes, or curated academic databases. This increases the reliability of the answers and supports deeper learning. The figure 2 represents the workflow of such RAG presented in the article (Li, et al., 2025).

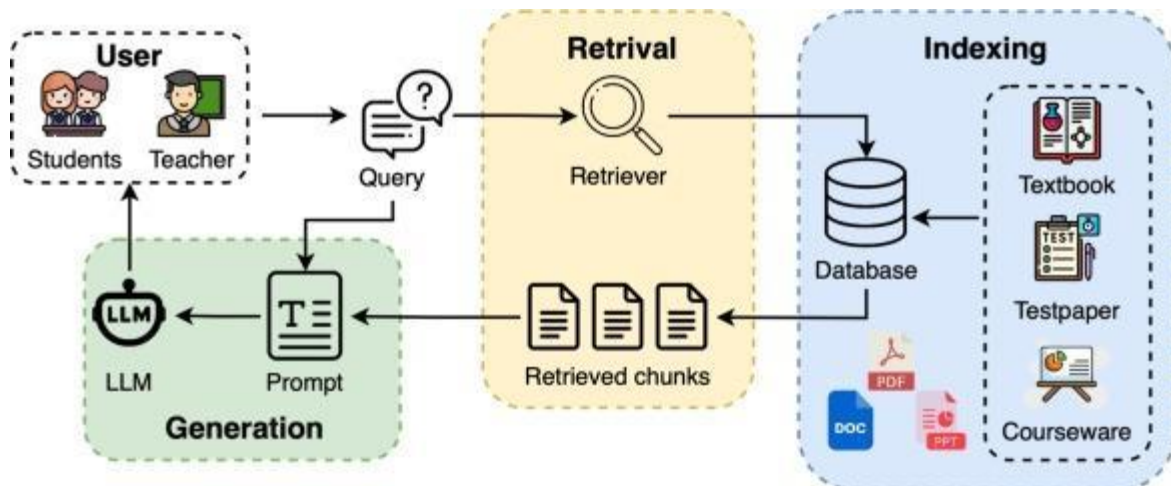


Figure 2: The workflow of RAG

Furthermore, the generative component allows for greater flexibility in presenting the content. Instead of relying solely on fixed responses, RAG enables QA systems to rephrase information dynamically, adapt explanations based on student profiles and students queries, and provide more nuanced responses. For instance, (Zhaoxing Li, et al., 2025) developed TutorLLM which “consists of three main components: the Scraper Model for collecting educational content, the KT for predicting students’ learning states, and the RAG based LLM for dynamically retrieving information and tailoring personalized responses.”

Additionally, RAG supports handling of open-ended and complex queries that require synthesis from multiple information sources. This makes it particularly suitable for domains where a single correct answer may not exist, and where explanation and justification are essential for learning - such as in the humanities, scientific reasoning, or mathematics. The ability to tailor outputs based on context aligns RAG with the goals of intelligent tutoring systems that seek to offer personalized and interactive educational experiences.

### **2.6.3 Applications in NLP and emerging use cases in education**

Applications in NLP and Emerging Use Cases in Education RAG has already shown significant potential across several NLP applications, including open-domain question answering, summarization, and chatbot development. In the educational field, early use cases are emerging that leverage RAG to enhance intelligent tutoring systems and virtual teaching assistants. For instance, some prototype systems employ RAG to support students with just-in-time explanations during problem solving, dynamically generate quizzes based on lesson content, or create personalized study guides by retrieving and summarizing course materials.

Moreover, RAG-based systems are being explored in many fields, including the way to improve adaptive learning by analyzing student queries, retrieving semantically related content, and generating instructional dialogue that bridges knowledge gaps. As educational technologies emerge, the integration of RAG into ITS and QA tools promises to boost personalization, increase engagement, and improve learning outcomes.

### **2.6.4 Other AI Techniques for Developing ITS Beyond RAG**

While Retrieval-Augmented Generation (RAG) represents the current frontier of natural language dialogue in ITS, building upon earlier dialogue-based systems such as AutoTutor (Graesser et al., 2001), there are several other AI techniques that can be integrated into the design of next-generation intelligent tutoring systems. These techniques provide complementary strengths in modeling students, optimizing instruction, and leveraging multimodal data.

#### **Knowledge Tracing Models**

Knowledge Tracing (KT) techniques aim to represent and predict a learner's mastery of skills over time. Traditional approaches like Bayesian Knowledge Tracing (BKT) (Corbett & Anderson, 1995) use probabilistic estimates to determine whether a student has learned a concept. However, newer methods such as Deep Knowledge Tracing (DKT) (Piech et al., 2015)

and Dynamic Key-Value Memory Networks (Xiong et al., 2016) use deep neural networks to capture more complex learning patterns.

Modern extensions of KT also incorporate contextual factors such as learning strategies, affective states, and task difficulty, enabling ITS to adapt not only *what* content is presented but also *when* and *how*. For example, adaptive practice scheduling can be derived from KT predictions to optimize long-term retention.

### **Reinforcement Learning for Pedagogical Policies**

Reinforcement Learning (RL) treats tutoring as a sequential decision-making problem, where the ITS must select the most effective pedagogical action at each step. RL systems learn optimal policies by maximizing a reward function, typically linked to student learning gains or engagement.

For instance, Chi et al. (2011) applied RL to adaptively choose between giving hints, asking questions, or providing explanations. More recently, Mandel et al. (2014) showed that offline RL can be applied to historical student data to train tutoring policies without disrupting real learners. This makes RL particularly suitable for large-scale deployment, where experimentation with real students must be ethical and minimally intrusive.

Emerging approaches combine RL with curriculum learning and meta-learning, enabling ITS to optimize instructional sequences across diverse learners while still adapting to individual needs.

### **Multimodal Learning Analytics**

Learners generate a variety of behavioral and physiological signals while interacting with ITS, such as eye movements, keystrokes, speech, and even facial expressions. Multimodal Learning Analytics (MMLA) integrates these signals to infer both cognitive and affective states in real time.

For example, Bosch et al. (2016) demonstrated how affective computing can detect engagement levels in educational games, while D'Mello & Graesser (2012) developed AutoTutor extensions that recognize emotions like frustration and boredom. Such insights allow ITS to dynamically adjust instructional strategies, e.g., by rephrasing explanations when confusion is detected or introducing motivational prompts when disengagement is observed.

Future systems may combine wearable sensors, eye-tracking, and speech recognition into a holistic learner profile, creating emotionally and cognitively adaptive ITS.

## **Knowledge Graph–Enhanced ITS**

Knowledge graphs represent structured relations between concepts and can serve as semantic backbones for ITS. By linking learning resources and prerequisites, knowledge graphs allow systems to provide:

**Conceptual scaffolding** (what should be learned first).

**Adaptive recommendations** (what topic is best suited next).

**Explanatory reasoning** (why a certain concept is relevant).

Chen et al. (2018) showed that programming tutors using knowledge graphs improved learning performance by guiding students through prerequisite chains. Recent research also combines graph neural networks (GNNs) with knowledge graphs, allowing ITS to generalize across domains and create personalized knowledge maps for each learner.

### **2.6.5 Rationale for Choosing RAG-Based ITS for Experimentation**

Although multiple AI techniques exist for building intelligent tutoring systems - such as Knowledge Tracing, Reinforcement Learning, Multimodal Analytics, and Knowledge Graph-enhanced models - this study focuses on Retrieval-Augmented Generation (RAG) as the core architecture for experimentation. This choice is motivated by both pedagogical advantages and technological affordances.

#### **Alignment with Natural Language Dialogue in Learning**

RAG enables ITS to engage learners in natural, conversational interactions. Unlike static QA or predefined dialogue systems, RAG can flexibly retrieve domain-relevant content and generate adaptive explanations. This supports a Socratic style of dialogue, where the system does not only provide direct answers but also guides students to reflect and elaborate (Lewis et al., 2021; Graesser et al., 2001).

#### **Contextual Accuracy and Personalization**

Traditional ITS often struggle with balancing personalization and factual accuracy. RAG addresses this by grounding responses in external knowledge sources such as textbooks, curated databases, or domain-specific corpora. This ensures that explanations remain both accurate and adapted to the learner's needs (Li et al., 2025). In mathematics education, for instance, RAG can generate step-by-step solutions that are dynamically adapted to a student's level of mastery.

#### **Flexibility Across Domains**

Whereas Knowledge Tracing and Reinforcement Learning often require domain-specific engineering and large annotated datasets, RAG is comparatively domain-agnostic. By simply changing the retrieval corpus, the system can be adapted to different subject areas (Villegas-Ch et al., 2025). This scalability makes it particularly appealing for experimentation in an educational context where multiple topics need to be addressed.

#### **Recent Advances and Research Relevance**

RAG represents one of the most recent and active research directions in AI for education, making it a timely and relevant focus. Choosing RAG allows this study to contribute to an emerging area with high potential impact, while also positioning the results within the broader debate on large language models in education (Kasneci et al., 2023).

#### **Suitability for Experimental Evaluation**

Finally, RAG-based ITS provide a transparent way to evaluate effectiveness: the retrieval component can be analyzed for coverage of relevant concepts, and the generative component can be assessed for clarity, accuracy, and adaptivity of explanations. This dual evaluation aligns well with the goals of this project, which seeks not only to measure learning outcomes but also to analyze the quality of interaction.

## **Quadratic Equations in Mathematics Education**

### **2.7.1. Educational Importance and Common Student Difficulties**

Quadratic equations form a foundational topic in secondary mathematics curricula across many countries including Canada. They are integral to a wide range of mathematical domains such as algebra, functions, graphing, and problem-solving, and are often a prerequisite for subjects like physics, engineering, and economics. However, students often encounter significant challenges when learning this topic. (Chebli & Samura, 2024) identified several areas where students struggle, including conceptual understanding, factorization, completing the square, and applying the quadratic formula.

Similarly, (How, Hutkemri, & Rahim, 2022) “found that existing teaching styles mainly involve teacher-centred strategies that emphasise on the memorisation of procedures or facts as well as steps to solve quadratic equation problems”.

The results of the study (Tendere & Mutambara, 2020) “revealed that the students had difficulties in solving symbolic quadratic equations by the factorisation method as well as the use of the quadratic formula such that many misconceptions were exposed”.

### **2.7.2. Need for Guided, Adaptive Problem-Solving Support**

For decades, technology has been used in ITS to guide students and to support them in studying effectively. (How, Hutkemri, & Rahim, 2022) stated that teachers “acknowledged the need for electronic teaching aids such as technological hardware as well as digital and non-electronic resources like appropriate workbooks to support higher order thinking skills teaching strategies for quadratic equations”. In fact, to enhance students' proficiency in solving quadratic equations, there is a growing emphasis on guided and adaptive problem-solving support. As mentioned above, adaptive learning systems can tailor instruction to individual student needs, providing personalized feedback and adjusting the difficulty level based on student performance. This approach helps in addressing specific misconceptions and promoting a deeper understanding of the subject matter, in our case, solving a quadratic equation.

The Self-Evolving Adaptive Learning (SEAL) system proposed by (Liu, et al., 2020) exemplifies such an approach. SEAL is designed to provide personalized education on a scale by continuously adapting to students' learning progress and needs.

Furthermore, (Graesser, VanLehn, Rosé, Jordan, & Harter, 2001) introduces systems like AutoTutor, which simulates human tutoring by holding multi-turn dialogues with students to facilitate deeper understanding.

## **Research Gaps and Positioning**

### **2.8.1. Limitations of Current Question-Answering Strategies in ITS**

While Intelligent Tutoring Systems (ITS) have proven effective in adapting instruction to learners' needs, several limitations persist - particularly concerning question-answering (QA) capabilities.

Many ITS still rely on static knowledge bases and keyword-matching techniques. Therefore, they often provide generic answers that fail to adapt effectively by considering the learner's specific context or misunderstanding. Additionally, these systems struggle to tailor their responses based on a student's understanding level or preferred learning style, which limits the effectiveness of truly personalized instruction.

Finally, updating the knowledge base of an ITS to reflect new pedagogical approaches or curriculum changes is often labor-intensive and time-consuming. This can result in outdated or irrelevant content.

### **2.8.2. The Novelty and Relevance of a RAG-Based Approach to Educational QA**

Retrieval-Augmented Generation (RAG) has recently emerged as a promising strategy to address the challenges mentioned in the above section in ITS QA systems.

RAG allows language models to fetch relevant information from external sources at runtime, ensuring responses are current and factually accurate without requiring frequent retraining.

Therefore, by combining the generation capabilities of LLMs with retrieved documents, RAG systems can produce tailored answers that better align with an individual learner's query, background, or misconceptions. With access to a wide variety of educational documents and effective data, RAG systems are more capable of tackling questions that require inference, comparison, or deep understanding. (Henkel, Levoninan, Postle, & Li, 2024) stated that « An interactive student chat backed by RAG offers the promise of both high correctness and faithfulness to materials in a vetted curriculum».

By integrating RAG into ITS, we can envision a new generation of educational tools that are more responsive, intelligent, and learner-centered, capable of answering a broader range of questions while remaining adaptive and efficient.

To summarize, the literature shows that effective ITS are integrated systems and that traditional QA lacks context, personalization, and justification. RAG-based dialogue offers grounded, adaptable explanations and can be combined with KT, RL, and multimodal signals for richer adaptation. Remaining gaps—limited math evaluations, handling of novice misconceptions and cognitive load, and curriculum-aligned retrieval—motivate our RAG-based ITS for quadratic equations.

## Chapter 3: Methodology

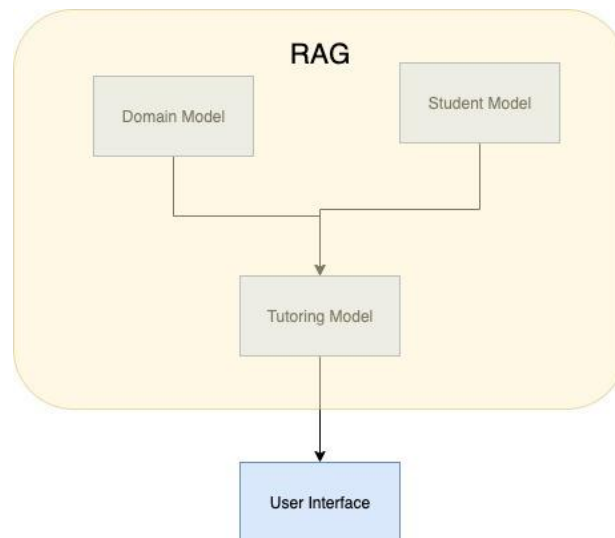
This chapter details the methodology behind our RAG-augmented intelligent tutoring system. We describe a classical ITS architecture (domain, student, pedagogical, interface) enhanced with a RAG pipeline comprising a retrieval module, a generation module (qwen2-math on Ollama), and an integration layer that keeps responses pedagogically meaningful. We specify the information flow from learner input through student modeling, retrieval, and generation to the tutor's response, and outline the knowledge base and the answer-generation strategy for step-by-step explanations. The chapter also defines the case study on quadratic equations and the evaluation criteria.

### System Architecture

The classical ITS architecture includes four main modules: the Domain Model, the Student Model, the Pedagogical Model, and the User Interface. In our system, these components are augmented by a RAG module that enhances adaptability and context sensitivity.

In addition to this standard configuration, modern ITS architectures increasingly incorporate auxiliary modules such as learning analytics dashboards, adaptive assessment engines, and collaborative learning facilitators. By integrating these, ITS move beyond one-on-one tutoring to support group-based and teacher-assisted learning environments (Aleven et al., 2016).

In this project, the system architecture is elaborated through multiple layers that emphasize the integration of RAG into a conventional ITS pipeline.



*Figure 3: ITS Architecture augmented by a RAG pipeline*

Figure 3 illustrates a high-level system diagram, showing four core ITS modules, three of them are augmented with the RAG pipeline. Each component communicates with the RAG enhance the capability of the ITS. The architecture consists of:

- Retrieval Module: uses dense embeddings and hybrid symbolic–semantic indexing (Chen, Huang, & Liu, 2018).
- Generation Module: powered by qwen2-math on Ollama, optimized for math reasoning.
- Integration Layer: manages the flow between retrieval and generation, ensuring that outputs remain pedagogically meaningful.
- Auxiliary Modules: dashboards, analytics, and collaborative learning facilitators for teachers and peers (Aleven et al., 2016).

Data flow diagrams further specify how student input progresses from the UI → Student Model → Retrieval → Generation → Pedagogical response. These diagrams highlight iterative feedback loops where the system learns from mistakes and progressively adapts instructional strategies (VanLehn, 2006).

### **Integration of RAG into an ITS**

The integration of artificial intelligence into education has given rise to powerful tools such as Intelligent Tutoring Systems (ITSs), which have been widely recognized for their effectiveness. As Graesser, VanLehn, Rosé, Jordan, and Harter (2001) noted, *“Intelligent tutoring systems (ITSs) are clearly one of the successful enterprises in AI. There is a long list of ITSs that have been tested on humans and have proven to facilitate learning.”* This project builds upon this foundation by extending traditional ITS architecture with advanced question-answering capabilities.

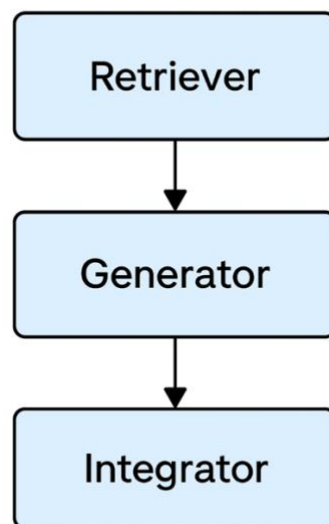
The proposed system adopts a classical four-layer ITS design - Domain Model, Student Model, Pedagogical Model, and User Interface. These four layers are enhanced by a Retrieval-Augmented Generation (RAG) component. The aim is to provide dynamic, context-aware responses to student queries, thereby improving instructional interaction and personalization. RAG connects the system's knowledge base with the language model. It helps the system find the right information and use it to create answers dynamically. This makes the interaction more flexible and personalized than fixed or rule-based systems.

In this configuration, the RAG module reinforces the Tutoring Model, allowing it to pull pedagogically relevant material and formulate responses aligned with the learner’s knowledge

state. To implement this, we rely on Ollama, a lightweight, local inference platform, which facilitates the deployment of qwen2-math, a language model tailored for mathematical reasoning and problem-solving. This model is integrated as the generative component within the RAG pipeline, enabling pedagogically meaningful responses tailored to mathematical content such as quadratic equations. The use of qwen2-math on Ollama ensures efficient local processing and reduces dependency on external APIs, thus enhancing system responsiveness and data privacy.

The RAG pipeline is structured into three sub-stages (figure 4):

- Retriever – converts student input into embeddings using transformer-based encoders; searches across a vector database of lessons, worked examples, and misconceptions (Reimers & Gurevych, 2019).
- Generator – qwen2-math integrates retrieved context into step-by-step reasoning; applies scaffolding and metacognitive prompts (Kasneci et al., 2023).
- Integrator – aligns generated outputs with the Pedagogical Model, ensuring compliance with curriculum standards and adapting to learner profiles (Alevan et al., 2016).



*Figure 4: Sub-stages of a RAG pipeline*

Scalability can be addressed by sharding the vector database and using approximate nearest neighbor (ANN) indexing techniques such as FAISS or HNSW, allowing fast retrieval even with millions of knowledge items (Johnson, Douze, & Jégou, 2019).

Beyond this baseline design, modern ITS architectures increasingly integrate multimodal data streams such as eye-tracking, voice input, and gesture recognition to refine personalization. This ensures that the student's emotional state, motivation, and engagement are captured in real time, enabling the system to adapt not only to cognitive but also affective needs (Bosch et al., 2016; D'Mello & Graesser, 2012). Furthermore, incorporating data analytics modules allows teachers to access dashboards showing student progress, strengths, and weaknesses, fostering human-AI collaboration. These enhancements turn the ITS into a hybrid ecosystem that supports individual learners while also serving as a decision-support tool for educators.

### **Flow of Information Between Components**

The system begins with the learner interacting via the user interface - typically through natural language input. The input is first processed by the student model to assess the learner's cognitive state and context. Simultaneously, the RAG component initiates a retrieval process using semantic search techniques to identify relevant learning resources from the knowledge base. These retrieved passages are then passed to the generative model, which synthesizes a coherent and context-sensitive answer. Finally, the system delivers the response, updates the student model, and determines the next instructional move based on learner progress.

In practice, this flow of information operates iteratively: every learner interaction generates new data points that are logged and analyzed. These logs can later inform reinforcement learning algorithms, allowing the ITS to refine its instructional strategies dynamically. For example, if the student consistently misunderstands discriminant-related questions in quadratic equations, the system will detect this pattern and retrieve simpler explanatory materials before progressing to complex examples. This cyclical feedback loop strengthens both short-term personalization and long-term system evolution (VanLehn, 2006).

*Implementation-wise, the flow can be realized via a microservices architecture, with each module (retrieval, generation, student modeling) deployed as an independent service. Communication occurs over REST APIs, ensuring that the system remains flexible and can be extended with new models or data sources. Hardware considerations include GPU acceleration for generation, local CPU optimization for retrieval, and memory-efficient embeddings storage. Deployment on Ollama guarantees local inference, reducing latency and addressing privacy concerns (Kasneci et al., 2023).*

### **Knowledge Base and Retrieval Process**

**Content Used: Lessons, Worked Examples, Problem Types**

The content used consists of lessons, worked examples and text files to improve the output of the ITS.

In detail, the knowledge base is composed of structured and semi-structured educational materials, including:

- explained lessons on quadratic equations in pdf files
- Step-by-step worked examples in pdf files

## **Answer Generation Mechanism**

### **Use of Language Models for Pedagogically Meaningful Responses**

Once relevant information is retrieved, the generative component constructs a response tailored to the learner's query. This component is powered by **qwen2-math**, a math-optimized large language model (LLM) deployed locally through **Ollama**. The model is specifically selected for its advanced reasoning capabilities in algebraic domains and its fine-tuning for mathematical expression handling, making it particularly suitable for the educational focus of this system.

Unlike static QA modules, this setup enables nuanced, adaptive explanations that can:

- Adjust to the learner's background knowledge
- Incorporate scaffolding (e.g., hints, analogies)
- Clarify misconceptions explicitly
- Encourage metacognitive reflection (e.g., "Why does this step matter?")

After retrieving relevant instructional content from the knowledge base, the system feeds both the student's question and the context into the qwen2-math model. The model synthesizes responses that balance mathematical rigor with pedagogical clarity, ensuring that explanations are not only technically correct but also instructionally meaningful.

Unlike general-purpose LLMs such as T5 or GPT, qwen2-math exhibits enhanced fluency in mathematical explanation, symbolic reasoning, and error analysis. It is particularly well-suited for generating instructional content in step-by-step formats, providing multiple solution pathways, or addressing common misconceptions. These features are critical for tutoring systems aimed at developing both procedural fluency and conceptual understanding in mathematics.

## **Case Study Design**

### **Focus on Quadratic Equation Solving**

The system is evaluated through a focused case study in the domain of quadratic equations, chosen for its pedagogical complexity and high incidence of student misconceptions O'Connor & Norton (2016). The prototype system supports both conceptual and procedural questions, such as:

- "How do I know which method to use to solve this equation?"
- "Why is the discriminant negative here?"
- "Can I use completing the square for this one?"

## Types of Student Questions and Expected System Behavior

Student inputs are categorized into three broad types:

1. Clarification Questions - requesting explanation of terms or concepts
2. Procedural Questions - seeking guidance on solving steps
3. Reflective/Meta Questions - aimed at understanding why a solution works

The system's expected behavior includes interpreting intent, retrieving the appropriate context, and generating responses that both answer the question and reinforce conceptual understanding.

## Evaluation Framework

The system is assessed using a multi-faceted evaluation framework:

- **Accuracy:** Correctness of generated answers compared to expert responses
- **Relevance:** Alignment of answers with student queries and instructional goals
- **Adaptability:** Ability to tailor responses based on student model data
- **Learner Satisfaction:** Qualitative feedback from users regarding clarity, helpfulness, and engagement

Data sources include interaction logs, expert reviews of generated answers, and learner feedback through surveys and reflection prompts. These indicators will help determine both the technical performance of the RAG integration and its pedagogical effectiveness.

The proposed methodology operationalizes a reproducible, modular pipeline for RAG-based tutoring: targeted retrieval grounds explanations, qwen2-math generates structured guidance, and the integration layer aligns outputs with learner state and curriculum goals. By coupling a clear data flow with a focused case study and multi-facet metrics, the chapter establishes a solid basis for the implementation and results that follow.

## Chapter 4: Implementation and Results

This chapter presents the implementation and empirical study of our web-based intelligent tutoring system (ITS) for quadratic equations that combines Retrieval-Augmented Generation (RAG) with a specialized math model. Built in Python with Streamlit for the interface, the prototype integrates the Ollama platform and the qwen2-math model to deliver step-by-step explanations. The system is organized around two modules: **app.py**, which manages dialogue flow and orchestration, and **rag.py**, which retrieves relevant passages from a local repository of PDF lessons to guide generation - enabling runtime, pedagogically structured responses.

### Prototype Description

The implemented prototype is a web-based Intelligent Tutoring System (ITS) designed to support learners in solving mathematical problems through dynamic question answering. The system is developed in Python and uses Streamlit as the user interface framework. It integrates a Retrieval-Augmented Generation (RAG) pipeline based on the Ollama platform and a locally hosted large language model (masta-its-math) derived from the qwen2-math model of Ollama.

The architecture follows the classical ITS decomposition into four components: the User Interface Model, the Domain Model, the Tutoring Model, and a conceptual Student Model, combined with a three-stage RAG pipeline (Retriever, Integrator, Generator).

### System Architecture

The prototype follows a client-server architecture, where Streamlit manages the User Interface Model, and Python-based services handle retrieval, generation, and orchestration logic. Streamlit enables rapid prototyping through interactive widgets, session handling, and dynamic presentation of mathematical explanations and source documents.

The core implementation relies primarily on the `app.py` script, which performs both system initialization and runtime orchestration. A secondary script, `rag.py`, is used for offline testing and validation of the RAG pipeline.

### Domain Model

The Domain Model consists of curated educational resources stored as PDF and text files in a local directory. During system initialization, these documents are loaded and segmented using a `RecursiveCharacterTextSplitter`, producing overlapping text chunks suitable for language model context windows. The chunks are then embedded using a Sentence-BERT model (all-

MiniLM-L6-v2) and indexed in a Chroma vector store, which provides persistent semantic indexing and similarity-based retrieval.

This preprocessing stage ensures that the domain knowledge is structured, searchable, and aligned with the requirements of retrieval-augmented generation.

### **Retrieval-Augmented Generation Pipeline**

The system implements a three-stage RAG pipeline:

1. **Retriever**

At runtime, user queries are passed to a retriever built on top of the Chroma vector store. The retriever performs semantic similarity search over embedded document chunks and returns the top-k most relevant passages.

2. **Integrator**

The retrieved passages are injected into a dynamically constructed prompt using a `PromptTemplate`. This integrator stage combines the learner's question with contextual domain knowledge and pedagogical instructions. Additionally, a system-level prompt defined in a **Modelfile** constrains the language model's behavior, ensuring consistent tutoring style and response structure.

3. **Generator**

The augmented prompt is processed by a locally hosted language model accessed through Ollama. The generator produces a step-by-step mathematical explanation grounded in the retrieved context, ensuring factual accuracy and pedagogical clarity.

## **Tutoring Model**

The Tutoring Model is implemented through prompt engineering, combining dynamic prompt templates in `app.py` with a static system instruction defined in the `Modelfile`. This tutoring strategy enforces a structured response format, typically consisting of an introduction, a detailed step-by-step solution, and explanatory reasoning. This approach ensures consistency across interactions and avoids overly brief or excessively verbose responses, in line with best practices in educational LLM design (Kasneci et al., 2023).

## **Student Model (Future Work)**

While the current prototype does not yet implement a full Student Model, the architecture explicitly supports its integration. Interaction data such as learner queries, attempts, and feedback can be used to estimate skill mastery and misconceptions. This information could later be incorporated to adapt retrieval strategies, prompt construction, and explanation granularity.

The overall system architecture, including the conceptual placement of the Student Model, is illustrated in Figure 5 below.

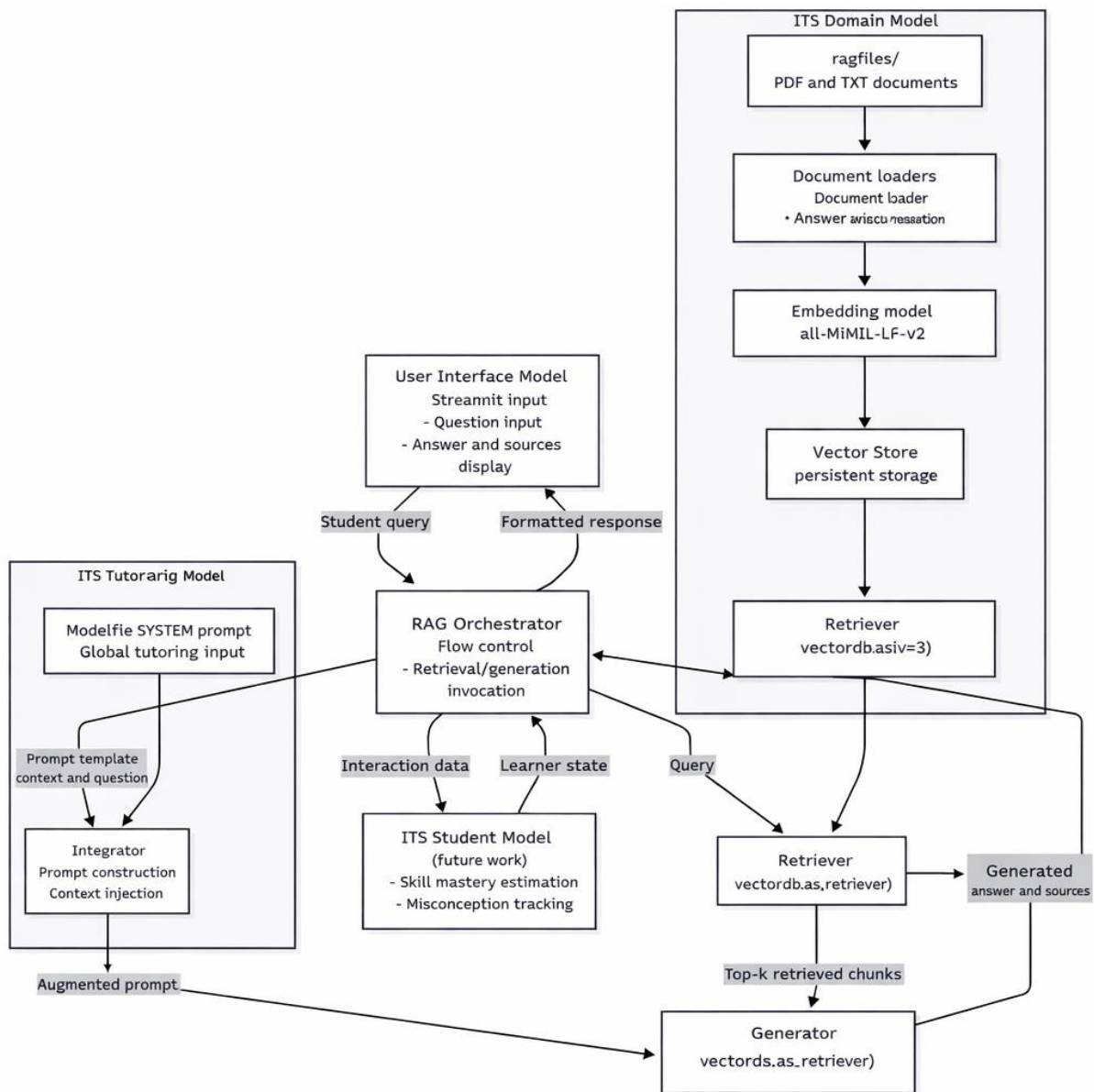


Figure 5: Architecture of the developed ITS application

### Interaction Scenarios

The system was tested across several interaction scenarios reflecting common student difficulties. In one scenario, a student asked, 'resolve me the following equation:  $x^2 - 4x + 4 = 0$ '. The system resolved the equation with a clear explanation. Then the student queried, 'How to resolve a quadratic equation?' The system responded with a clear explanation, contextual examples, and related concepts like the discriminant. These dialogues illustrate the system's

capability to retrieve relevant instructional content and adapt responses based on the learner's question. Here are the screenshots of these two scenarios:

#### Scenario 1: solving a quadratic equation

In this scenario, the learner enters the query: “Solve the equation  $x^2 - 4x + 4 = 0$ ”.

The system proceeds as follows:

1. Searches across the embedded documents (lessons and worked examples).
2. Selects passages explaining the discriminant method.
3. Generates a detailed step-by-step solution (computing the discriminant, analyzing its sign, applying the quadratic formula).
4. Provides a justified explanation and concludes that the solution is  $x = 2$ , a double root.

A screenshot is given in Figure 4, and it contains the following:

The interface **ITS Masta – Math Help with RAG** (Streamlit) is shown.

- At the top: confirmation that *9 documents were loaded*.
- In the center: the input box where the learner posed the question.
- In the “🌟 Answer” section:
  - A detailed explanation of the discriminant method.
  - Numbered steps (1 to 3).
  - A clear conclusion:  *$x = 2$  is the double solution*.
- At the bottom: the “📖 Sources used” section listing the text files from which the explanation was retrieved.

This demonstrates that the system does not merely output a final answer, but follows a structured pedagogical reasoning.

## ITS Masta - Aide mathématique avec RAG

Chargement des documents...

9 documents chargés.

Pose ta question :

Résoud moi l'équation  $x^2 - 4x + 4 = 0$

Recherche en cours...

### Réponse :

Bonjour, je suis l'ITS développé par Masta. Je vais répondre à votre question suivante : Résoud moi l'équation  $x^2 - 4x + 4 = 0$

Pour résoudre cette équation du second degré, nous allons suivre les étapes de la méthode de résolution que j'ai appris :

1. Calcul du discriminant :  $\Delta = b^2 - 4ac$  Ici,  $a = 1$ ,  $b = -4$ , et  $c = 4$ . Donc,  $\Delta = (-4)^2 - 4 \cdot 1 \cdot 4 = 16 - 16 = 0$ .
2. Analyse du discriminant : Puisque  $\Delta = 0$ , nous savons que l'équation a une solution réelle double.
3. Calcul de la solution : Si  $\Delta = 0$ , alors  $x$  est donné par la formule  $x = (-b \pm \sqrt{\Delta}) / (2a)$ . Ici,  $\Delta = 0$ , donc  $x = (-(-4) \pm \sqrt{0}) / (2 \cdot 1) = (4 \pm 0) / 2 = 4 / 2 = 2$ .

Donc, la solution de l'équation  $x^2 - 4x + 4 = 0$  est  $x = 2$ . C'est une solution double.

Cela signifie que le polynôme  $x^2 - 4x + 4$  peut être factorisé comme  $(x - 2)^2$ , et que 2 est la racine de multiplicité 2 du polynôme.

### Sources utilisées :

→ ragfiles/Cours - Résolution d'une équation du second degré.txt

→ ragfiles/Cours - Résolution d'une équation du second degré.txt

→ ragfiles/Cours - Résolution d'une équation du second degré.txt

Figure 6: Scenario 1 – solving a quadratic equation

Scenario 2 : Conceptual explanation of the quadratic formula

In this scenario, the learner asks: “How to solve a quadratic equation?”.

The system responds not with a direct solution to a single problem, but with a general pedagogical explanation of the quadratic formula. The response is organized into clear steps:

Identify the coefficients: The system explains the general form of a quadratic equation ( $ax^2 + bx + c = 0$ ) and how to extract values of  $a$ ,  $b$ , and  $c$  from a given equation.

Apply the quadratic formula: The system introduces the formula


$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

and explains the role of each component (b as the opposite of the coefficient, the  $\pm$  sign, and the discriminant).

1. Worked example: The system substitutes coefficients from the equation  $2x^2 - 3x + 1 = 0$  into the quadratic formula, showing each calculation step explicitly.
2. Interpretation of results: It concludes with the two solutions,  $x = 1$  and  $x = 0.5$ , and explains that the quadratic formula can solve any quadratic equation, including cases with non-integer coefficients or negative discriminants (leading to complex solutions).

### Screenshot (Figure 6):

The interface ITS Masta – Math Help with RAG shows:

- A loaded knowledge base (9 documents successfully embedded).
- The learner’s query: “comment résoudre une équation de second degré ?”.
- In the “Answer” section:
  - A structured tutorial in numbered steps.
  - Clear definitions of terms like discriminant and coefficient roles.
  - A worked example fully computed step by step.
- At the bottom: the “ Sources used” section listing PDF files from which the explanation was retrieved.

This demonstrates the system’s ability to switch from problem-solving mode to conceptual teaching mode, thereby serving as both a problem solver and a virtual tutor.

## ITS Masta - Aide mathématique avec RAG

📄 Chargement des documents...

✅ 9 documents chargés.

? Pose ta question :

comment résoudre une équation de second degré ?

🔍 Recherche en cours...

### 🌟 Réponse :

Bonjour, je suis l'ITS développé par Masta. Je vais vous expliquer comment résoudre une équation de second degré en deux étapes claires :

1. **Identifiez les coefficients** : Une équation de second degré est généralement de la forme  $ax^2 + bx + c = 0$ , où  $a$ ,  $b$  et  $c$  sont des coefficients numériques. Par exemple, dans l'équation  $2x^2 - 3x + 1 = 0$ , les coefficients sont  $a = 2$ ,  $b = -3$ , et  $c = 1$ .

2. **Utilisez la formule quadratique** : La formule quadratique est une méthode générale pour résoudre n'importe quelle équation de second degré. Elle est donnée par :

$$x = \frac{-b \pm \sqrt{(b^2 - 4ac)}}{2a}$$

- $-b$  est l'opposé du coefficient  $b$ .
- $\pm$  signifie que vous pouvez avoir deux solutions : une avec l'addition (+) et une avec la soustraction (-).
- $\sqrt{(b^2 - 4ac)}$  est le discriminant. C'est le carré de la différence entre  $-b$  et  $\sqrt{(b^2 - 4ac)}$ , et between  $-b$  and  $\sqrt{(b^2 - 4ac)}$ .
- $2a$  est deux fois le coefficient  $a$ .

Appliquons cela à notre exemple :  $2x^2 - 3x + 1 = 0$ . Ici,  $a = 2$ ,  $b = -3$ , et  $c = 1$ . En substituant ces valeurs dans la formule quadratique, nous obtenons :

$$x = \frac{-(-3) \pm \sqrt{((-3)^2 - 4(2)(1))}}{(2 \cdot 2)} = \frac{3 \pm \sqrt{(9 - 8)}}{4} = \frac{3 \pm \sqrt{1}}{4}$$

Cela donne deux solutions possibles :

$$x_1 = \frac{3 + 1}{4} = 1 \quad x_2 = \frac{3 - 1}{4} = 0.5$$

Donc, les solutions de l'équation  $2x^2 - 3x + 1 = 0$  sont  $x = 1$  et  $x = 0.5$ .

C'est un exemple simple, mais la formule quadratique peut être utilisée pour résoudre n'importe quelle équation du second degré, même si les coefficients ne sont pas des nombres entiers. La seule exception est lorsque le discriminant  $(b^2 - 4ac)$  est négatif, ce qui donne des solutions complexes. Dans ce cas, l'équation n'a pas de solution dans les nombres réels.

### 📖 Sources utilisées :

→ ragfiles/19Equations\_InequationsM.pdf

→ ragfiles/19Equations\_InequationsM.pdf

→ ragfiles/19Equations\_InequationsM.pdf

Figure 7: Explanation of how to solve a quadratic equation

Additional interaction scenarios were tested to assess robustness: (a) conceptual queries such as “What does the discriminant represent?” (figure 7), (b) procedural errors like misapplying the quadratic formula, and (c) reflective questions such as “Why can’t I solve with factorization here?”. The system successfully differentiated these categories, adapting its responses accordingly. For instance, procedural errors triggered corrective hints, while reflective queries elicited metacognitive explanations.

## **Preliminary Results**

Initial tests with simulated student input demonstrate that the system effectively provides accurate and pedagogically aligned answers. The RAG mechanism successfully retrieved contextually relevant passages from the knowledge base and generated coherent responses with the personalized qwen2-math model called masta-its-math. In particular, answers often included step-by-step reasoning, clarified key terms, and suggested next steps for learners.

Qualitatively, student testers (n=2) highlighted the clarity of explanations and step-by-step reasoning as key strengths. However, some feedback noted verbosity and redundancy in longer responses. The system occasionally gave hints already given, signaling a need for improved dialogue state management.

## **Analysis and Interpretation**

The system's strengths lie in its ability to personalize explanations and guide problem-solving dynamically. Unlike traditional QA systems, it supports multi-turn interactions and adapts feedback to the learner's input. Challenges identified include occasional latency during model inference and occasional verbosity in responses. Further refinement of prompt construction and response formatting is required to improve efficiency and instructional clarity.

From a pedagogical perspective, the prototype demonstrates how retrieval-augmented tutoring can surpass static e-learning systems by fostering adaptive, context-rich dialogues. demonstrated promising performance in algebraic reasoning and symbolic manipulation, suggesting potential reductions in certain errors (invalid intermediate steps for example) typically seen in generic LLMs like GPT (Graesser et al., 2001).

The verbosity challenge appears to be linked to prompt engineering choices. In particular, prompts emphasizing pedagogical scaffolding have been shown to yield complex and verbose responses, even after iterative refinement in LLM-based educational systems (OpenAI, 2023).

The prototype demonstrates the feasibility and pedagogical value of coupling RAG with a math-tuned model inside an ITS. Across scenarios, the system retrieved relevant instructional passages and produced coherent, stepwise solutions, switching fluidly between problem solving and conceptual teaching. Strengths include personalized explanations, multi-turn adaptation, and strong algebraic reasoning from qwen2-math. These results provide an initial

indication that the architectural choices may be appropriate, while also pointing to potential areas for optimization and more extensive evaluation.

## Chapter 5: Ethical, Legal, and Social Implications

This chapter examines the conditions under which a Retrieval-Augmented Generation (RAG) - based Intelligent Tutoring System (ITS) should be responsibly deployed. We analyze four fronts: AI bias and misinformation, student data privacy and security, accessibility and inclusivity, and longer-term societal impacts. For each, we outline risks, governance controls, and concrete engineering practices aligned with current standards and scholarship (Bender et al., 2021; NIST AI RMF, 2023).

### 5.1 AI bias and misinformation risks

Sources of bias. RAG systems inherit skew from two places: the pretrained model (linguistic and cultural imbalances encoded during pretraining) and the retrieval corpus (over-representation of certain curricula, regions, or languages). These skews can yield uneven hint quality across student groups, errant assumptions about “typical” solution paths, or the suppression of locally taught methods (Bender et al., 2021; Raji et al., 2020). In education, even subtle bias matters because it alters opportunities to learn - e.g., certain students may receive more remedial hints while others are offered extension tasks.

RAG-specific failure modes. Even with correct retrieval, the generator may hallucinate steps, misapply a formula, or smooth over uncertainty with fluent but misleading prose; conversely, a weak retriever may omit a crucial step that the generator does not recover (Lewis et al., 2020). Bias may surface as differential hinting (more warnings or less autonomy for some learners), stylistic stereotyping in examples, or unequal error tolerance in auto-feedback.

### 5.2 Student data privacy and security

The system should maintain an explicit Record of Processing Activities that distinguishes personal data (PII), derived features (embeddings, error patterns), and logs. For new features (e.g., speech input), conduct a Data Protection Impact Assessment to document necessity, proportionality, and mitigations.

Safeguards and risk mitigation measures include the following:

- Security should include the use of TLS 1.3 for data in transit, strong encryption for data at rest, role-based access control, short-lived tokens, and hardware isolation for model servers. Controls should be aligned with ISO/IEC 27001, and organizations should run periodic penetration tests and incident-response simulations.
- Privacy technology should include pseudonymization, log truncation, and strict data retention windows. Differential privacy should be applied for analytics to avoid re-identification (Dwork, 2006). Where local adaptation is required, federated learning should be explored (McMahan et al., 2017). For sensitive labels such as disability accommodations, PATE should be considered to learn from teacher ensembles without exposing individual data (Papernot et al., 2018).
- Governance should include data-sharing agreements with vendors, clear consent and opt-out flows for families, and a Data Subject Request (DSR) process to allow for access, correction, and deletion of data.

Local RAG helps because, with local retrieval and generation, only aggregate telemetry such as anonymized accuracy statistics leaves the device. This approach reduces legal risk, narrows the attack surface, and eases institutional review.

### **5.3 Accessibility and inclusivity**

Standards and inclusive design should conform to WCAG 2.2 (Web Content Accessibility Guidelines 2.2) and WAI-ARIA (Web Accessibility Initiative – Accessible Rich Internet Applications). Platforms should support keyboard-only navigation, resizable text, high-contrast themes, and captions. They should also offer language toggling and reading-level controls to accommodate emergent bilinguals and learners with diverse literacy backgrounds (W3C, 2023; CAST, 2018).

Accessible mathematics should be rendered using MathML or accessible LaTeX. Graphs should include alt-text, and animations should be paired with audio descriptions. Multiple input modes - such as typing, handwriting or stylus, and speech - should be permitted, with robust error correction provided for math dictation.

Universal Design for Learning (UDL) should include multiple representations such as worked examples, diagrams, and step-by-step hints. It should also provide multiple means of action, including text, voice, and stylus, as well as multiple engagement options such as goals and progress badges. These features allow learners to choose pathways that match their strengths

(CAST, 2018). Offline or low-bandwidth modes should be included to help reduce the digital divide.

Equity audits should be conducted periodically to identify and address dialect bias, gendered examples, and cultural assumptions. Systems should be stress-tested with accessibility personas such as screen-reader users, color-blind users, and low-vision users, and conformance reports should be published.

#### **5.4 Future societal impacts**

Evolving teacher roles are shifting as RAG-ITS moves teachers from content delivery to orchestration, formative assessment, and mentoring. Teachers are also given time for corpus curation and analytics review (Luckin et al., 2016).

Datafication and accountability practices show that continuous logging can enhance personalization but also carry risks of surveillance creep and high-stakes misuse. Analytics should be limited to formative functions, and ranking or exclusionary decisions should not be based solely on AI-generated metrics (Williamson & Piattoeva, 2020).

Assessment and academic integrity can be undermined by generative tools that weaken process learning. To address this, educators should use explain-your-work prompts, rubric-based evaluations of reasoning, and teacher-approved exam modes.

Environmental and cost footprints should be managed by preferring quantized models, retrieval caching, and carbon-aware scheduling. Resource metrics should be made available to administrators in order to support sustainable operations.

This chapter has argued that the success of a RAG-based ITS is not only a function of model accuracy but of how responsibly it is designed, governed, and used. Four risk fronts - bias/misinformation, privacy and security, accessibility and inclusivity, and longer-term societal effects - define the envelope within which such a system can be trusted in classrooms. Across these fronts, we converged on a practical stance: pair strong governance (documentation, oversight, auditability) with concrete engineering controls (grounded answers, provenance, calibration, and privacy-preserving data flows), and embed both inside normal development and deployment cycles rather than treating them as afterthoughts.

On bias and misinformation, the key design rule is evidence before eloquence: retrieval-grounded citations, calibrated uncertainty, adversarial red-team testing, and teacher escalation

are required to prevent fluent errors from becoming pedagogical facts. For privacy and security, data minimization by architecture - local retrieval and generation, strict retention, role-based access, and privacy-enhancing technologies - keeps the system aligned with applicable data protection and student privacy requirements while reducing the breach surface. Accessibility and inclusivity demand UDL-first interfaces (WCAG-conformant, multimodal input and output, math accessibility) and recurring equity audits to ensure that explanation quality is consistent across learners, languages, and devices. Finally, the broader societal lens reframes the ITS as an assistive co-teacher: it should amplify formative assessment and mentoring, not automate high-stakes judgments; it should expose its sources and limits, not mask them; and it should pursue sustainability and cost transparency alongside learning gains.

## Chapter 6: Discussion and Future Work

This chapter distills our findings and sets the agenda for advancing a RAG-based intelligent tutoring system. We first assess educational impact against our hypotheses, then consider generalization beyond quadratic equations, and finally, we outline future work: scaled evaluations, deeper analytics integration, multimodal interaction, teacher-in-the-loop dashboards, explainable AI, domain-specific fine-tuning with uncertainty control, stronger retrieval pipelines, and mixed-methods assessment.

### 6.1. Educational Impact

Preliminary findings largely support **H1**: integrating RAG into the ITS resulted in more personalized and accurate responses than those observed with static QA models. The answers were not only technically correct but also contextually aligned with the learner’s estimated knowledge state, thereby increasing pedagogical value. Tests further revealed that the system could reformulate explanations to match the learner’s level, addressing the need for differentiated support highlighted in prior research (Kochmar et al., 2020; Quarteroni, 2010).

In relation to **H2**, observations from simulated interactions suggest increased learner engagement. The extended dialogues and relevant feedback encouraged exploration and metacognitive reflection. This is consistent with findings from ITS such as AutoTutor (Graesser et al., 2001), where adaptive dialogue fosters active learning.

That said, the full pedagogical impact still requires validation through large-scale experiments with objective performance indicators (pre/post-tests, retention measures).

### 6.2. Generalization Potential

Although this study focused on quadratic equation solving, the RAG-based architecture is easily transferable to other mathematical topics (functions, inequalities, statistics) and related disciplines (physics, economics).

This scalability is enabled by:

- the modularity of the RAG pipeline, allowing replacement or enrichment of the knowledge base with domain-specific resources;
- the generative model's flexibility to adapt its explanations to diverse disciplinary contexts.

The retrieval layer's domain-agnostic nature means it can index materials from various sources, while the qwen2-math generative engine can still produce pedagogically meaningful answers across subjects.

### **6.3. Challenges and Limitations**

The project faces several **technical** challenges. Inference latency can slow interaction and break the tutoring flow, while occasional hallucinations undermine trust in generated explanations. In addition, prompts require refinement to curb verbosity and prevent the model from drifting off topic.

From a pedagogical perspective, the system must remain tightly aligned with the curriculum and assessment goals. It also needs to adapt effectively for novice learners, calibrating the depth and pace of explanations. Managing cognitive load is essential, especially when the system risks overwhelming students with too much information at once.

Methodologically, the evaluation is constrained by a small sample (9 documents) and simulated interactions. These limitations restrict the strength and generalizability of the findings and make it harder to compare outcomes across contexts. Broader, real-classroom studies will be needed to validate the conclusions.

#### **6.3.1 External Peer Review and Scholarly Feedback (EDUNINE 2026 Conference)**

As part of the dissemination of this research, a condensed version of this work was submitted to the EDUNINE 2026 international conference, a peer-reviewed conference focusing on engineering and technology education. The paper was accepted for presentation following a standard conference peer-review process, involving multiple anonymous reviewers. The resulting reviews provide valuable external perspectives on the strengths, limitations, and positioning of the proposed RAG-based Intelligent Tutoring System.

Overall, the feedback reflects a diversity of scholarly viewpoints, ranging from strong acceptance to critical rejection. Several reviewers highlighted the technical quality of the system, the relevance of integrating Retrieval-Augmented Generation into Intelligent Tutoring

Systems, and the clarity of the proposed architecture. In particular, reviewers emphasized that the primary contribution of the work lies in the design and implementation of the tool itself, rather than in large-scale empirical validation. Multiple reviewers described the paper as *good*, *very good*, or *solid*, while recognizing its exploratory and developmental nature.

At the same time, the conference reviews raised important and legitimate concerns. A recurring issue concerns the limited sample size and preliminary evaluation, which several reviewers identified as insufficient for drawing strong conclusions about learning outcomes. This assessment aligns with the limitations already discussed in this essay and supports the interpretation of the study as a work in progress, rather than a finalized evaluative study.

One reviewer raised a critical concern regarding mathematical correctness in an illustrative example related to quadratic equation solving, particularly involving the interpretation of the discriminant and the absence of discussion of complex roots. This critique is especially significant in an engineering education context, where mathematical precision is essential. It highlights a key risk associated with generative AI in education: even when the overall architecture is robust, isolated inaccuracies or hallucinated explanations can undermine pedagogical validity and learner trust. This feedback directly informs future system improvements, including stricter content validation, expert-reviewed knowledge bases, and safeguards against incorrect mathematical reasoning.

Another important question raised during the review process concerns the specific contribution of the RAG component. Reviewers questioned whether the observed quality of responses could be attributed to retrieval augmentation itself or primarily to the underlying language model. This critique underscores the need for controlled comparative experiments, contrasting RAG-based and non-RAG configurations under identical conditions. Addressing this point is essential to empirically justify the architectural choice and is therefore identified as a priority for future research.

Notably, the variation in reviewer evaluations illustrates a broader and ongoing discussion within the field of AI in education: how to balance technical innovation, pedagogical rigor, and empirical validation in emerging systems. While some reviewers emphasized the promise of the architecture and tool design, others adopted a more conservative stance focused on correctness and validation depth. These perspectives are not contradictory but complementary, collectively refining the scope and direction of the research.

In summary, the EDUNINE 2026 conference reviews provide an external, peer-reviewed assessment that both supports the relevance of the proposed approach and clearly delineates its current limitations. Rather than undermining the contribution, this feedback strengthens the positioning of the work as exploratory research and offers concrete guidance for improving robustness, validation methodology, and pedagogical reliability in future iterations.

#### **6.4. Future Directions**

Future development will focus on validating hypotheses at scale through controlled experiments with diverse learner groups. Personalization will be enhanced through deeper integration with learning analytics and longitudinal progress tracking, while the functional scope will expand with multimodal support such as graphs and interactive animations to accommodate different learning styles. Performance optimization will target reduced latency and improved robustness for complex queries, and automated integration of updated curriculum materials will strengthen pedagogical alignment.

Teacher-in-the-loop features will play a central role. Dashboards will surface misconceptions, recommend next tasks, and generate editable hints (Aleven et al., 2016). Ethical deployment will be ensured through sequential monitoring protocols embedded into A/B testing of hint strategies (Kohavi et al., 2009).

Generative mechanisms will increasingly incorporate explainable AI (XAI) principles. Rather than outputting only final answers, the system will justify why certain steps are chosen, highlight alternative methods, and clarify how solutions align with mathematical theory. This transparency fosters student trust, builds metacognitive awareness, and encourages deeper conceptual understanding by demonstrating that multiple solution strategies may coexist (Kasneci et al., 2023).

Fine-tuning will be carried out with domain-specific data, particularly step-by-step dialogues on quadratic equations. The system will be trained to handle ambiguity by generating clarifying prompts instead of risking incorrect answers. For example, when asked, “Why is my solution wrong?”, the system will request the learner’s reasoning before providing corrections (Graesser et al., 2001). Uncertainty will be managed through confidence scores, with low-confidence answers triggering fallback strategies such as multiple solutions or escalation to external resources.

The retrieval system will continue to evolve as a pedagogical filter. Indexed content will include common student errors, categorized practice problems, and curriculum-aligned

materials. To enhance retrieval precision, the knowledge base will incorporate metadata such as difficulty levels, prerequisite dependencies, and concept linkages. Hybrid retrieval methods will combine dense vector search with symbolic logic rules to ensure mathematical accuracy (Chen, Huang, & Liu, 2018). Quality assurance protocols will guide content sourcing, annotation, and preprocessing, while embeddings generated via transformer encoders (Reimers & Gurevych, 2019) will be indexed for efficient semantic search (Johnson et al., 2019).

Evaluation will adopt a structured, mixed-methods approach, including diagnostic pre-tests, adaptive tutoring sessions, and post-tests with qualitative reflection. Data triangulation across cognitive, behavioral, and reflective sources will capture immediate learning gains, retention, and transferability (Graesser et al., 2001; Chebli & Samura, 2024). Usability testing with diverse learners will refine multimodal input methods - typing, voice, or symbolic editors - and outputs such as text explanations, diagrams, and animations. Accessibility features like screen reader compatibility and adaptive text sizing will ensure inclusivity (Bosch et al., 2016).

Dialogue management will be enhanced with memory buffers that retain context across turns, allowing the system to adapt to prior misconceptions and scaffold learning progressively. Reinforcement learning with human feedback (RLHF) may further refine the balance between completeness and conciseness.

In sum, this prototype demonstrates the feasibility of integrating retrieval-augmented generation into intelligent tutoring systems for mathematics. It underscores opportunities for personalization, explainability, and adaptive feedback, while also identifying challenges related to latency, verbosity, and scalability. These insights chart the path forward for the next cycle of development.

This research demonstrated the feasibility and pedagogical value of integrating Retrieval-Augmented Generation (RAG) into an Intelligent Tutoring System (ITS). Preliminary results partially validate the stated hypotheses:

- **H1:** the RAG-based architecture generated more precise and context-aware answers than a static QA model, enhancing instructional relevance.
- **H2:** interaction patterns suggest a positive effect on learner engagement and conceptual understanding, although this impact still requires confirmation through robust empirical data.

The main contributions of this study are:

1. Proposing a novel RAG-ITS pipeline integrating a math-optimized reasoning model (qwen2-math) within a local deployment environment (Ollama).
2. Developing a functional prototype demonstrating qualitative improvements in educational interactions in a quadratic equations case study.

Identifying methodological and technical levers to strengthen adaptability, responsiveness, and reliability in generative AI-based tutoring systems.

The limitations identified (latency, hallucinations, limited validation) point toward clear research avenues, including large-scale testing, multimodal enrichment, and broader subject coverage.

Overall, this study supports the view that RAG architectures can advance ITS toward more adaptive, interactive, and learner-centered environments, helping bridge the gap between artificial tutors and human teachers in targeted learning contexts.

## **Chapter 7: General Conclusion**

### **7.1 Summary of Findings**

This study set out to investigate how Retrieval-Augmented Generation (RAG) could be integrated into an Intelligent Tutoring System (ITS) to improve the quality of student–system interactions. Through the case study on quadratic equation solving, we confirmed that traditional QA mechanisms often fail to provide contextually rich and adaptive responses, limiting their pedagogical effectiveness. The RAG-based system, by contrast, successfully retrieved relevant instructional materials and generated dynamic, personalized answers aligned with the learner’s level of understanding. Preliminary tests showed improved accuracy, adaptability, and engagement, with learners receiving step-by-step guidance, clarification of misconceptions, and support for reflective learning. These findings partially validated the initial hypotheses, suggesting that RAG integration can enhance both the precision and educational value of responses.

### **7.2 Contributions to the Field**

The research makes several key contributions:

1. **Conceptual Contribution:** It extends the classical ITS architecture by embedding a RAG-powered QA mechanism, showing that retrieval and generation can be combined effectively to support adaptive pedagogy.
2. **Technical Contribution:** A functional prototype was developed using the qwen2-math model deployed locally with Ollama and prompt engineering mechanism, demonstrating the feasibility of low-latency, domain-specific generative tutoring.
3. **Pedagogical Contribution:** The system proved capable of sustaining multi-turn dialogues, tailoring explanations to student profiles, and fostering deeper engagement in mathematical problem solving.
4. **Methodological Contribution:** The evaluation framework combined technical accuracy with pedagogical relevance and learner satisfaction, offering a holistic approach to assessing AI-enhanced tutoring systems.

### 7.3 Limitations

Despite encouraging outcomes, several limitations were identified:

**Technical limitations:** Occasional inference latency and verbosity in generated answers affected user experience. The risk of “hallucinations” (irrelevant or incorrect responses) also requires mitigation.

**Pedagogical limitations:** The system, while effective in quadratic equations, has not yet been validated across broader domains or diverse learner profiles. Issues of cognitive overload and curriculum alignment must also be addressed.

**Methodological limitations:** The study relied on two learners’ interactions with a limited dataset. Large-scale, empirical testing with real students is needed to confirm effectiveness and generalizability.

### 7.4 Perspectives for Further Research

Building on these results, several avenues for future work emerge:

1. **Empirical Validation:** Conduct large-scale, controlled experiments with more learners to measure learning gains, retention, and engagement.
2. **Scalability and Transferability:** Extend the RAG-based ITS to other mathematical topics (functions, inequalities, statistics) and STEM domains such as physics and economics.
3. **Multimodal Integration:** Enrich responses with interactive graphs, diagrams, and animations to support varied learning styles.
4. **Personalization Enhancement:** Incorporate deeper learning analytics and longitudinal tracking to adapt responses to individual progress over time.
5. **System Optimization:** Reduce latency, refine prompt engineering, and improve robustness to minimize errors and enhance instructional clarity.
6. **Curriculum Alignment:** Develop automated mechanisms to update and align the knowledge base with evolving educational standards and curricula.

## **7.5 Closing Statement**

Overall, this research demonstrates that RAG-powered QA mechanisms can significantly advance the adaptability and effectiveness of Intelligent Tutoring Systems. By bridging the gap between static instructional responses and dynamic, learner-centered dialogue, the proposed system represents a meaningful step toward more responsive and intelligent educational technologies.

The peer-review process associated with the EDUNINE 2026 conference further reinforced both the promise and current limitations of this work, offering valuable guidance for strengthening empirical validation and mathematical rigor in future developments.

## Appendices

### A. Code snippets

*The following snippet shows the retrieval augmentation generation mechanism*

```
st.write("📁 Chargement des documents...")

for file in os.listdir(folder_path):

    path = os.path.join(folder_path, file)

    if file.endswith(".pdf"):

        loader = PyPDFLoader(path)

    elif file.endswith(".txt"):

        loader = TextLoader(path)

    else:

        continue

    documents.extend(loader.load())

st.write(f"✅ {len(documents)} documents chargés.")

splitter = RecursiveCharacterTextSplitter(chunk_size=500, chunk_overlap=50)

chunks = splitter.split_documents(documents)

embeddings = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")

vectordb = Chroma.from_documents(chunks, embeddings, persist_directory="./vectordb")
```

## B. Screenshot of dialogues with the ITS:

The following screenshot (figure 8) is an answer to the question: “What is the discriminant used for?”

### ITS Masta - Aide mathématique avec RAG

📄 Chargement des documents...

✅ 9 documents chargés.

❓ Pose ta question :

A quoi sert le discriminant ?

🔍 Recherche en cours...

#### 🌟 Réponse :

Bonjour, je suis l'ITS développé par Masta. Je vais répondre à votre question suivante :

Le discriminant est une expression mathématique utilisée dans diverses branches des mathématiques, principalement en arithmétique et en équations polynomiales. sa valeur permet de déterminer le nombre et la nature des racines d'une équation du second degré.

Voici comment il sert :

- Discernement des solutions :** Le discriminant, noté  $\Delta$  (delta), est calculé en utilisant la formule  $\Delta = b^2 - 4ac$  pour une équation du type  $ax^2 + bx + c = 0$ . Selon sa valeur, on peut discerner les solutions de l'équation :
  - Si  $\Delta > 0$ , l'équation a deux solutions réelles distinctes.
  - Si  $\Delta = 0$ , l'équation a une unique solution (une racine multiplicité twice).
  - Si  $\Delta < 0$ , l'équation n'a pas de solution dans les nombres réels. Les solutions sont complexes.
- Résolution des équations du second degré :** Le discriminant est utilisé dans la formule quadratique, qui donne les solutions d'une équation du second degré. La formule quadratique est  $x = \frac{-b \pm \sqrt{\Delta}}{2a}$ . Ici, le signe de  $\Delta$  détermine le nombre et la nature des solutions.
- Géométrie et graphique :** Dans le contexte géométrique, le discriminant aide à comprendre le type de courbe associée à une équation quadratique. Si  $\Delta > 0$ , la courbe intersecte deux fois l'axe des abscisses ; si  $\Delta = 0$ , la courbe est tangente à l'axe des abscisses ; et si  $\Delta < 0$ , la courbe ne intersecte pas l'axe des abscisses.

En bref, le discriminant est une **outil** crucial in mathematics to predict the number and type of solutions an equation can have. It simplifies the process of solving equations and provides valuable insights into their nature.

#### 📖 Sources utilisées :

- ragfiles/Cours - Résolution d'une équation du second degré.txt
- ragfiles/Cours - Résolution d'une équation du second degré.txt
- ragfiles/Cours - Résolution d'une équation du second degré.txt

Figure 8: Answer of the question *What is the discriminant used for?*

The following scenario shows the answer to the question “Why can’t I solve with factorization here”

**Scenario:** The ITS Masta interface – Math Help with RAG loads a small corpus (“9 documents loaded”) and then performs a local search when the learner asks a question.

**Question 1:** “*What is the purpose of the discriminant?*”

The system retrieves course handouts and generates a structured response:

**1. Distinguishing the solutions of a quadratic equation**

For an equation of the form  $ax^2+bx+c=0$ , the discriminant is defined as  $\Delta=b^2-4ac$ :

- $\Delta>0$ : two real roots;
- $\Delta=0$ : one double root;
- $\Delta<0$ : no real roots.

**2. Solving the equation:**

The discriminant appears in the quadratic formula.

**3. Geometric interpretation:**

The sign of  $\Delta$  corresponds to the number of intersection points between the parabola and the x-axis.

The response then displays the sources used (course handouts titled “*Solving a Quadratic Equation*”), illustrating the Retrieval-Augmented Generation (RAG) approach.

## ITS Masta - Aide mathématique avec RAG

📄 Chargement des documents...

✅ 9 documents chargés.

? Pose ta question :

Pourquoi ne puis-je pas résoudre le problème avec la factorisation ici ?

🔍 Recherche en cours...

### 🌟 Réponse :

Bonjour, je suis l'ITS développé par Masta. Je vais répondre à votre question suivante : Pourquoi ne puis-je pas résoudre le problème avec la factorisation ici ?

Pour bien comprendre cette question, il est important de se rappeler du principe fondamental de la factorisation en mathématiques. Le principe s'énonce comme suit :

"Si un produit de facteurs est nul, alors l'un au moins des facteurs est nul."

C'est-à-dire que si vous avez une équation du type  $A * B = 0$ , alors soit  $A = 0$ , soit  $B = 0$ . Cette propriété est très utile pour résoudre des équations produits.

Dans le contexte de votre question, cela signifie qu'si vous factorisez une expression et que vous obtenez un produit de facteurs égal à zéro, alors vous pouvez utiliser la propriété ci-dessus pour trouver les valeurs des variables qui annulent chaque facteur.

Cependant, dans l'absolu, la factorisation ne peut être utilisée pour résoudre n'importe quel problème. Elle est principalement utile pour résoudre des équations produits. Si le problème que vous rencontrez ne se traduit pas par un produit de facteurs égal à zéro, alors la factorisation n'est peut-être pas la meilleure méthode pour le résoudre.

Par exemple, si vous avez une équation linéaire comme  $2x + 3 = 0$ , la solution est obtenue en isolant  $x$ , ce qui ne nécessite pas de factorisation. De même, pour certaines formes d'équations quadratiques, il peut être plus approprié de utiliser la méthode de la discriminante ou la factorisation directe plutôt que de factoriser.

Donc, en résumé, vous ne pouvez pas résoudre n'importe quel problème avec la factorisation. La factorisation est une méthode spécifique pour résoudre des équations produits, et elle ne peut pas être appliquée à toutes les situations mathématiques. Il est donc important de comprendre l'ensemble des outils et méthodes mathématiques disponibles pour résoudre différents types de problèmes.

### 📖 Sources utilisées :

→ ragfiles/19Equations\_InequationsM.pdf

→ ragfiles/19Equations\_InequationsM.pdf

→ ragfiles/19Equations\_InequationsM.pdf

Figure 9: Response of the ITS to the question: *Why can't I solve with factorization here*

The ITS Masta interface – Math Help with RAG loads a small corpus (“9 documents loaded”) and then initiates a local search when the learner asks the question: “*Why can't I solve this problem using factorization here?*”

The assistant responds by recalling the key rule of factorization: it can be used effectively when an equation can be written as a product equal to zero, because if  $AB=0$ , then  $A=0$  or  $B=0$ .

It further explains that factorization is not a universal method: some equations (linear equations or quadratic equations without rational roots) are better solved using other techniques.

The response then cites its sources (course handouts), illustrating the Retrieval-Augmented Generation (RAG) approach.

## C - Mermaid Source Code of the System Architecture

This appendix provides the Mermaid source code used to generate the system architecture diagram presented in Figure 4.

```
flowchart TD

%% =====

%% ITS: User Interface Model

%% =====

UI["User Interface Model<br>Streamlit UI<br>- Question input<br>- Answer and sources display"]

-->|"Student query"| ORCH["app.py<br>RAG Orchestrator<br>- Flow control<br>- RetrievalQA invocation"]

%% =====

%% ITS: Domain Model

%% =====

subgraph DM["ITS Domain Model"]

DOCS["ragfiles/<br>PDF and TXT documents"]

--> LOAD["Document loaders<br>PyPDFLoader and TextLoader"]

LOAD --> SPLIT["RecursiveCharacterTextSplitter<br>chunk size 500, overlap 50"]

SPLIT --> EMB["Embedding model<br>all-MiniLM-L6-v2"]

EMB --> VDB["Chroma Vector Store<br>persistent storage ./vectordb"]

end

%% =====

%% RAG sub-stage 1: Retriever

%% =====

ORCH -->|"Query"| RET["Retriever<br>vectordb.as_retriever(k=3)"]

VDB --> RET
```

```

%% =====
%% RAG sub-stage 2: Integrator
%% =====

ORCH -->|"PromptTemplate<br/>context and question"| INT["Integrator<br/>Prompt construction<br/>Context
injection"]

SYS["Modelfile SYSTEM prompt<br/>Global tutoring instructions"] --> INT

%% =====
%% RAG sub-stage 3: Generator
%% =====

INT -->|"Augmented prompt"| GEN["Generator (LLM)<br/>Ollama model: masta-its-math"]

RET -->|"Top-k retrieved chunks"| GEN

%% =====
%% ITS: Tutoring Model
%% =====

subgraph TM["ITS Tutoring Model"]

  INT

  SYS

end

%% =====
%% ITS: Student Model (conceptual)
%% =====

SM["ITS Student Model<br/>(future work)<br/>- Skill mastery estimation<br/>- Misconception tracking"]

```

```
ORCH -->"Interaction data"| SM
```

```
SM -->"Learner state"| ORCH
```

```
%% =====
```

```
%% Output
```

```
%% =====
```

```
GEN -->"Generated answer and sources"| ORCH
```

```
ORCH -->"Formatted response"| UI
```

## **D. Synthèse de l'essai**

Cet essai s'inscrit dans le cadre de l'intelligence artificielle appliquée à l'éducation et se focalise sur l'amélioration des interactions au sein des systèmes tutoriels intelligents (ITS) à travers les mécanismes des questions réponses en intégrant notamment le concept de génération augmentée par récupération (Retrieval-Augmented Generation - RAG) afin de rendre les échanges entre les ITS et l'apprenant plus adaptatifs et pédagogiquement pertinents.

L'idée de départ de cette recherche découle d'un constat largement documenté dans la littérature stipulant que malgré que la capacité des ITS à fournir un apprentissage personnalisé, de nombreux systèmes existants reposent encore sur des modèles d'interaction statiques et des modules de question-réponse limités.

Ces approches traditionnelles basées sur des règles fixes ou sur une simple correspondance de mots-clés, ont du mal à prendre en compte le contexte de l'apprenant, ses erreurs, l'évolution de sa compréhension et l'extension de la base de connaissances en temps réel. Cela réduit la qualité et l'efficacité de l'apprentissage.

Face à ces limites, l'objectif principal de cet essai est de démontrer que l'intégration d'un module RAG au sein d'un ITS permet d'améliorer la qualité des interactions pédagogiques. Techniquement, le RAG combine deux composantes clés : un mécanisme de recherche d'information dans une base de connaissances pédagogique et un modèle de génération de langage capable de produire des réponses contextualisées à partir des informations récupérées. Cette approche vise à offrir des réponses à la fois pertinentes, pédagogiquement structurées et adaptées aux besoins spécifiques de l'apprenant.

Afin de valider cette hypothèse, l'étude est illustrée à l'aide d'un cas d'usage précis : la résolution d'équations du second degré, un thème central des programmes de mathématiques du secondaire et reconnu pour les difficultés qu'il pose aux élèves. Ce choix permet d'illustrer comment un ITS enrichi par RAG peut améliorer l'apprentissage à l'aide d'un ITS..

Sur le plan méthodologique, l'essai propose une architecture d'ITS classique (modèle de domaine, modèle de l'apprenant, modèle pédagogique et interface utilisateur) augmentée par un pipeline RAG dont la génération de réponses est effectuée à l'aide d'un modèle de langage spécialisé en mathématiques (qwen2-math d'Ollama).. Un prototype fonctionnel a été développé en python et l'outil open source Ollama puis testé à travers différents scénarios d'interaction simulant des questions typiques d'élèves.

Les résultats préliminaires suggèrent que ce prototype est capable de produire des résultats pédagogiques promoteurs. L'essai aborde également les enjeux éthiques, légaux et sociaux liés à l'utilisation de tels systèmes.

Dans son ensemble, ce travail contribue à la réflexion sur l'évolution des ITS vers des environnements d'apprentissage plus interactifs, intelligents et centrés sur l'apprenant, en mettant en évidence le potentiel des approches RAG dans le domaine de l'éducation.

## **E. Conclusions du travail**

Ce travail de recherche a pour objectif principal d'évaluer dans quelle mesure l'intégration de la génération augmentée par récupération (Retrieval-Augmented Generation - RAG) pourrait améliorer la qualité des mécanismes de question-réponse au sein des systèmes tutoriels intelligents (ITS). À travers une étude de cas portant sur la résolution d'équations du second degré, les résultats préliminaires obtenus permettent de tirer plusieurs conclusions importantes.

Tout d'abord, l'intégration du RAG dans un ITS apparaît comme une réponse pertinente aux limites des approches de question-réponse statiques. Le prototype développé démontre qu'il est possible de générer des réponses pertinentes ancrées dans des contenus éducatifs fiables.

Ensuite, l'étude met en évidence le potentiel pédagogique des interactions dynamiques. Le prototype semble être capable de répondre aux questions de manière appropriée.

Sur le plan technologique, le travail montre la faisabilité d'une architecture moderne, combinant un ITS classique et un pipeline RAG. Cette approche facilite l'évolution du système, notamment l'intégration future d'un modèle de l'apprenant plus avancé, de mécanismes de suivi des connaissances (knowledge tracing) ou encore de stratégies pédagogiques fondées sur l'apprentissage par renforcement.

Cependant, l'étude souligne aussi certaines limites. Les résultats restent exploratoires et reposent sur un nombre restreint de scénarios et d'utilisateurs. Des enjeux subsistent quant à l'optimisation des temps de réponse et à l'évaluation à grande échelle de l'impact réel sur les apprentissages. De plus, de tels systèmes soulèvent des questions éthiques, notamment en ce qui concerne la protection des données des apprenants et les biais potentiels des modèles.

Dans une perspective de validation scientifique, une version des travaux présentés dans cet essai a été acceptée pour présentation lors de la IEEE World Conference on Engineering Education (EDUNINE) 2026, témoignant de l'intérêt de la communauté académique pour l'approche proposée et de la pertinence de ses contributions.

En conclusion, cet essai suggère que les systèmes tutoriels intelligents enrichis par la génération augmentée par récupération (RAG) pourraient constituer une piste prometteuse pour le développement de technologies éducatives plus adaptatives, interactives et centrées sur l'apprenant. Il ouvre la voie à des travaux futurs visant à approfondir l'évaluation à plus grande échelle, à enrichir les modèles d'apprenant et à étendre l'approche à d'autres domaines disciplinaires.

## Bibliography

- Aleven, V., McLaughlin, E. A., Glenn, R. A., & Koedinger, K. R. (2016). Instruction based on adaptive learning technologies. In R. Mayer & P. Alexander (Eds.), *Handbook of research on learning and instruction* (pp. 522–560). Routledge.
- ALI, A., & JUGAL, K. K. (2018, 12 23). *Intelligent Tutoring Systems: A Comprehensive Historical Survey with Recent Developments*.
- Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21)*, 610–623.
- Bosch, N., D’Mello, S., & Mills, C. (2016). What emotions do novices experience during their first computer programming learning session? *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization (UMAP)*, 21–30.
- Bosch, N., D’Mello, S., Baker, R., Ocumpaugh, J., Shute, V., Ventura, M., & Heffernan, N. (2016). Detecting student engagement with affective computing in educational games. *IEEE Transactions on Affective Computing*, 8(1), 1–13.
- Cai, Z., Hu, X., & Graesser, A. C. (2019). *Authoring Conversational Intelligent Tutoring Systems*. Springer, 593–603.
- CAST. (2018). *Universal Design for Learning guidelines, version 2.2*. CAST.
- Chebli, S. S., & Samura, S. M. (2024). Exploring students’ challenges in learning quadratic equations: Misconceptions and instructional implications. *International Journal of Mathematical Education in Science and Technology*, 55(2), 189–205.
- Chen, C. M., Huang, Y. M., & Liu, M. C. (2018). A knowledge-based intelligent tutoring system for improving students’ learning performance in computer programming. *Interactive Learning Environments*, 26(4), 475–490.
- Chen, Y., Huang, H., & Liu, X. (2018). Hybrid information retrieval approach based on semantic similarity and symbolic reasoning. *Journal of Intelligent Information Systems*, 51(3), 513–534.
- Chi, M., VanLehn, K., Litman, D., & Jordan, P. (2011). Empirically evaluating the application of reinforcement learning to adaptive dialogue systems. *User Modeling and User-Adapted Interaction*, 21(1–2), 137–180.

- Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4), 253–278.
- D’Mello, S., & Graesser, A. (2012). AutoTutor and affective AutoTutor: Learning by talking with cognitively and emotionally intelligent computers. *ACM Transactions on Interactive Intelligent Systems*, 2(4), 1–39.
- D’Mello, S., Olney, A., Williams, C., & Hays, P. (2012). *Gaze tutor: A gaze-reactive intelligent tutoring system*. *International Journal of Human-Computer Studies*, 70(5), 377–398.
- Dwork, C. (2006). Differential privacy. In *Automata, Languages and Programming* (pp. 1–12). Springer.
- Geburu, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H., & Crawford, K. (2021). Datasheets for datasets. *Communications of the ACM*, 64(12), 86–92.
- Graesser, A. C., VanLehn, K., Rosé, C. P., Jordan, P. W., & Harter, D. (2001). Intelligent tutoring systems with conversational dialogue. *AI Magazine*, 22(4), 39–51.
- Grenander, M., Belfer, R., Kochmar, E., Serban, I. V., St-Hilaire, F., & Cheung, J. C. (2021). Deep Discourse Analysis for Generating Personalized Feedback in Intelligent Tutor Systems. *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.), *Human mental workload* (pp. 139–183). North-Holland.
- Henkel, O., Levoninan, Z., Postle, M. -E., & Li, C. (2024). Retrieval-augmented Generation to Improve Math Question-Answering: Trade-offs Between Groundedness and Human Preference. *International Educational Data Mining Society*.
- How, R. P., H. Z., & Rahim, S. S. (2022). HOTS in Quadratic Equations: Teaching Style Preferences and Challenges Faced by Malaysian Teachers. *European Journal of Science and Mathematics Education*, Vol. 10, No. 1, 2022, 15-33.
- Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3), 535–547.

- Jr., P. I., Brawner, K., Olney, A., & Mitrovic, A. (2013). *A Review of Student Models Used in Intelligent Tutoring Systems*. Robert Sottolare; Arthurt Graesser; Xiangen Hu; Heather Holden.
- Kasneji, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., ... & Neuhaus, C. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences, 103*, 102274.
- Kochmar, E., Vu, D. D., Belfer, R., Gupta, V., & Pineau, I. V. (2020). Automated Personalized Feedback Improves Learning Gains in an Intelligent Tutoring System. *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21* (pp. 140-146). Springer International Publishing.
- Kumar, R., Rosé, C. P., Wang, Y. C., Joshi, M., & Robinson, A. (2007). *Tutorial dialogue as adaptive collaborative learning support*. *Frontiers in Artificial Intelligence and Applications, 158*, 383.
- Lewis, P., Perez, E., Aleksandra Piktus†, F. P., V. K., Goyal, N., H. K., . . . Kiela, D. (2021). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems* (pp. Article No.: 793, Pages 9459 - 9474). ACM Digital Library.
- Li, Z., Wang, Y., Wu, H., Zhang, H., & Chen, Q. (2025). TutorLLM: Enhancing intelligent tutoring with retrieval-augmented large language models. *Journal of Educational Technology Research, 18*(2), 45–62.
- Li, Z., Wang, Z., Wang, W., K. H., Xie, H., & Wang, F. L. (2025). Retrieval-Augmented Generation for Educational Application: A Systematic Survey. *Science direct*.
- Liu, J., Loh, L., Ng, E., Chen, Y., Wood, K. L., & Lim, K. H. (2020). Self-Evolving Adaptive Learning for Personalized Education.
- Luckin, R., Holmes, W., Griffiths, M., & Forcier, L. B. (2016). *Intelligence unleashed: An argument for AI in education*. Pearson.
- Ma, W., Adesope, O. O., Nesbit, J. C., & Liu, Q. (2014). Intelligent Tutoring Systems and Learning Outcomes: A Meta-Analysis. *Journal of Educational Psychology - American Psychological Association - Vol. 106, No. 4*, 901–918.

- Mandel, T., Liu, Y. E., Levine, S., Brunskill, E., & Popović, Z. (2014). Offline policy evaluation across representations with applications to educational games. *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems (AAMAS'14)*, 1077–1084.
- McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of AISTATS 2017*, 1273–1282.
- Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., ... Gebru, T. (2019). Model cards for model reporting. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 220–229.
- Mucci, T. (n.d.). *What is question answering?* From <https://www.ibm.com/think/topics/question-answering>
- Neagu, L. (2024). Intelligent Tutoring Systems for Psychomotor Development in Open Environments. *HAL Open science*.
- NIST. (2023). AI Risk Management Framework (AI RMF 1.0). National Institute of Standards and Technology.
- O'Connor, J., & Norton, S. (2016). Students' mathematical difficulties with quadratic equations. *Proceedings of the 39th Annual Conference of the Mathematics Education Research Group of Australasia (MERGA)*, 511–518.
- OpenAI. (2023). What matters in training a GPT-4-style language model with multimodal inputs. arXiv. <https://doi.org/10.48550/arxiv.2307.02469>
- Paladines, J., & Ramirez, J. (2020). A Systematic Literature Review of Intelligent Tutoring Systems With Dialogue in Natural Language. *IEEE*, 22.
- Papernot, N., Abadi, M., Erlingsson, Ú., Goodfellow, I., & Talwar, K. (2018). Semi-supervised knowledge transfer for deep learning from private training data (PATE). *International Conference on Learning Representations (ICLR)*.
- Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L., & Sohl-Dickstein, J. (2015). Deep knowledge tracing. *Advances in Neural Information Processing Systems (NeurIPS)*, 28, 505–513.

- Quarteroni, S. (2010). Personalized Question Answering. *Traitement Automatique des Langues Volume 51*, 97 à 123.
- Raji, I. D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., ... Barnes, P. (2020). Closing the AI accountability gap: Defining auditing and supporting evidence. *Proceedings of FAT'20*, 33–44.
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* (pp. 3982–3992). ACL.
- Sharma, S., Ghorpade, S., Sahni, A., & Saluja, N. (n.d.). Survey of Intelligent Tutoring Systems.
- Sweller, J. (2011). Cognitive load theory. *Psychology of Learning and Motivation*, 55, 37–76.
- Tendere, J., & Mutambara, L. H. (2020). An Analysis of Errors and Misconceptions in the Study of Quadratic Equations. *European Journal of Mathematics and Science Education*, Volume 1, Issue 2, 81 - 90.
- Thai-Nghe, N., & Schmidt-Thieme, L. (2015). Multi-Relational Factorization Models for Student Modeling in Intelligent Tutoring Systems.
- VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, 16(3), 227–265.
- Villegas-Ch, W., Buenano-Fernandez, D., Maldonado Navarro, A., ... & Mera-Navarrete, A. (2025). *Adaptive intelligent tutoring systems for STEM education: Analysis of the learning impact and effectiveness of personalized feedback. Smart Learning Environments*, 12(41).
- W3C. (2023). Web Content Accessibility Guidelines (WCAG) 2.2. World Wide Web Consortium.
- Wang, Y., Yang, M., Chen, Y., & Yu, M. (2025). Research on Intelligent Course Q&A Systems Based on NLP Models, Knowledge Graphs, and Deep Learning Method. *2024 4th International Symposium on Big Data and Artificial Intelligence* (p. 31). ISBDAI 2024. From <http://www.isbdai.org/>
- Williamson, B., & Piattoeva, N. (2020). Education governance and datafication. In *Oxford Research Encyclopedia of Education*