

Enabling Assessments with Artificial Intelligence

Master Thesis

written by
Dmitriy An

supervised and evaluated by

Prof. Dr. Andrew Paice (Advisor)
Prof. Dr. Petra Müller-Csernetzky (Co-Advisor)
Dr. Christopher Ganz (Expert)

on the

Lucerne University of Applied Sciences and Arts – Master of Science in Engineering
Field of study: Data Science

Submission date
HS25, 15. January 2026

Abstract

English

Assessing student proficiency is a resource-intensive challenge in higher education. To address the lack of deep assessment capabilities in Edisconet's platform, this thesis investigates the optimal Large Language Model (LLM) pipeline for generating open-ended exercises from unstructured course materials. Using a mixed-methods approach, four pipelines were developed comparing context preparation (Sliding Window vs. Concept Extraction) and cognitive frameworks (Bloom's Revised Taxonomy vs. Webb's Depth of Knowledge). A survey of 21 module coordinators at HSLU evaluated the output quality. Results indicate that Concept Extraction combined with Webb's Depth of Knowledge yields the highest satisfaction, particularly for technical subjects. While effective as a cost-efficient (\$0.05/set) drafting engine, findings confirm that a review by educators is essential for validity, establishing the tool as a productivity aid rather than an autonomous examiner.

Key Terms: Automated Assessment, Large Language Models, Question Generation, Webb's Depth of Knowledge, Bloom's Taxonomy, Higher Education, Proficiency Estimation

Deutsch

Die Kompetenzbeurteilung von Studierenden ist eine ressourcenintensive Herausforderung. Um die Edisconet-Plattform um tiefgehende Bewertungsmöglichkeiten zu erweitern, untersucht diese Arbeit die optimale Large Language Model (LLM)-Pipeline zur Generierung offener Übungsaufgaben aus unstrukturierten Materialien. Mittels eines Mixed-Methods-Ansatzes wurden vier Pipelines entwickelt, die Kontextaufbereitung (Sliding Window vs. Concept Extraction) und kognitive Rahmenwerke (Bloom's Revised vs. Webb's Depth of Knowledge) vergleichen. Eine Umfrage unter 21 Modulverantwortlichen an der HSLU evaluierte die Qualität. Die Ergebnisse zeigen, dass Concept Extraction kombiniert mit Webb's DOK die höchste Zufriedenheit erzielt, besonders in technischen Fächern. Während sich das System als kosteneffiziente „Drafting Engine“ (0,05 \$/Set) bewährt, bleibt ein Review von Dozenten für die Validität unerlässlich. Das Tool fungiert somit als Produktivitätshilfe, nicht als autonomer Prüfer.

I. Table of Contents

Declaration of Independence and Honesty	ii
Independence	ii
Honesty	ii
Abstract	iii
I. Table of Contents	i
II. List of Figures	iii
III. List of Tables	iv
IV. Glossary	v
1. Introduction	1
1.1 Context	1
1.2 Problem	1
1.3 State-of-the-Art and Knowledge Gap	2
1.4 Objectives and Research Question	2
1.5 Methodology	3
1.6 Structure of the Thesis	4
2. Literature Review	5
2.1 Method of Literature Analysis	5
2.2 Cognitive Frameworks	7
2.3 Existing Tools and Comparison	13
2.4 Course Material Modeling	17
2.5 Exercise Generation	26
2.6 Assessment Design	31
2.7 Didactics with Exercises	33
2.8 Knowledge Gap and Research Objective	34
3. Methodology	38
3.1 Method Selection	38

3.2	Prototype Implementation.....	41
3.3	Survey	44
4.	Results.....	46
4.1	Prototype Results	46
4.2	Survey Results	50
4.3	Findings.....	57
5.	Discussion.....	59
5.1	Discussion of Findings.....	59
6.	Conclusion	63
6.1	Practical Implications.....	63
6.2	Theoretical Implications	63
6.3	Limitations and further Research.....	64
6.4	Recommendations.....	64
7.	Reflection on Project	65
7.1	Requirements	65
7.2	Risks.....	65
7.3	Project Management	65
8.	List of References	67
9.	Appendix.....	75
9.1	Project Plan and Milestones.....	1
9.2	Catalog of Requirements.....	2
9.3	Risk Management	4
9.4	Prompt Instructions.....	6
9.5	Gioia's Data Structure.....	22
9.6	OLS Results	27

II. List of Figures

Figure 1 <i>Methodology Sketch</i>	3
Figure 2 <i>Bloom's Original Taxonomy Hierarchy</i>	7
Figure 3 <i>Bloom's Revised Taxonomy Matrix</i>	8
Figure 4 <i>SOLO Taxonomy Hierarchy</i>	9
Figure 5 <i>Webb's Depth of Knowledge</i>	10
Figure 6 <i>Dreyfus' original 5-stage Model</i>	11
Figure 7 <i>Proposed Framework for Personalized E-Learning</i>	13
Figure 8 <i>Workflow of the MCQ Generation</i>	15
Figure 9 <i>Training process of Question Generation</i>	15
Figure 10 <i>Matrix about the type of Questions that ChatGPT can generate</i>	16
Figure 11 <i>Example Question Generation Process for the Text Content in one Topic</i>	18
Figure 12 <i>Block Algorithm to calculate the Lexical Score</i>	19
Figure 13 <i>Vocabulary Introduction Algorithm to calculate the Lexical Score</i>	19
Figure 14 <i>Chains Algorithm to calculate the Lexical Score</i>	20
Figure 15 <i>Dual two-layer bidirectional LSTM Architecture for Topic Segmentation</i>	21
Figure 16 <i>Macro Discourse Parser Framework with Example</i>	21
Figure 17 <i>Model Architecture of TM-BERT</i>	22
Figure 18 <i>Process of Macro Discourse Parser</i>	22
Figure 19 <i>Proposed Pseudo-Instruction for document Chunking Framework</i>	23
Figure 20 <i>Three-step process of LumberChunker</i>	23
Figure 21 <i>Savaal five-step Pipeline</i>	24
Figure 22 <i>ConQuer Framework</i>	25
Figure 23 <i>Construction Process of a Knowledge Graph</i>	25
Figure 24 <i>Workflow of SciCheck</i>	26
Figure 25 <i>Quality and Skill of Prompting Strategies by different LLMs</i>	27
Figure 26 <i>Self-validating Question Generation Flowchart based on Bloom</i>	28
Figure 27 <i>Question Generation Process using Discourse Cues</i>	29
Figure 28 <i>Overview of a RAG Framework using Webb's DOK</i>	30
Figure 29 <i>Preferences for ChatGPT-4 vs BloomLLM across Cognitive Levels</i>	30
Figure 30 <i>Overview of the TwinStar Architecture</i>	31
Figure 31 <i>Degree of Alignment in Taxonomy and Quality</i>	31
Figure 32 <i>Model to build Authentic Assessments</i>	32
Figure 33 <i>Pacing for Instructions</i>	33
Figure 34 <i>Core Critical Thinking Skills</i>	34
Figure 35 <i>Methodology Sketch</i>	38
Figure 36 <i>Implementation Flowchart</i>	42
Figure 37 <i>Recursive Chunking Flowchart</i>	43
Figure 38 <i>GPT-5 and GPT-5-mini Pricing</i>	44
Figure 39 <i>Time Distribution for Processing Steps</i>	46
Figure 40 <i>Cost Distribution for all API calls</i>	48
Figure 41 <i>Percentual Distribution of Tokens and their Costs</i>	49
Figure 42 <i>Bloom vs. Webb Survey Preferences</i>	51
Figure 43 <i>Sliding Window vs. Concept Extraction Survey Preferences</i>	52
Figure 44 <i>Pipeline Survey Preferences</i>	53
Figure 45 <i>Ratings about Extracted Concepts</i>	54
Figure 46 <i>Survey Ratings about Usability with Automated Evaluation</i>	54
Figure 47 <i>Survey Ratings about Usability with Manual Evaluation</i>	55

III. List of Tables

Table 1 <i>Searched Databases and Filters</i>	5
Table 2 <i>Search Queries and Article Counts</i>	6
Table 3 <i>Simplified Cognitive Rigor Matrix</i>	12
Table 4 <i>Cognitive Framework Comparison</i>	13
Table 5 <i>Question Type and Target Argument for Discourse Connectives</i>	28
Table 6 <i>Knowledge GAP-Table of Literature Review</i>	36
Table 7 <i>Morphological Box for Pipeline Prototypes</i>	39
Table 8 <i>Pipeline Combinations for Implementation</i>	41
Table 9 <i>Contacted Module Coordinators</i>	45
Table 10 <i>Revisited GAP-Table</i>	60

IV. Glossary

DL	Deep Learning
ML	Machine Learning
LLM	Large Language Model
LSTM	Long Short-Term Memory
RAG	Retrieval-Augmented Generation
MCQ	Multiple-Choice Question
NLP	Natural Language Processing
DOK	(Webb's) Depth of Knowledge

1. Introduction

This thesis addresses the growing administrative burden on educators and the lack of personalized assessment tools in digital learning environments. As educational platforms transition from simple content delivery to active proficiency estimation, the need for scalable, didactically sound exercise generation becomes paramount. This chapter establishes the foundation for a technical solution developed in partnership with Edisconet.

The following sections detail the context of the research, the specific limitations of current Large Language Model (LLM) implementations in pedagogy, and the primary objectives of the study. It concludes with a description of the mixed-methods research design and a structural overview of the thesis.

1.1 Context

In education, assessing learners' actual understanding of course material remains a persistent challenge. Teachers spend, on average, only half of their time directly interacting with students. Out of a typical 50-hour workweek, around 10.5 hours are dedicated to preparation tasks, including developing content and exercises (U.S. Department of Education & Office of Educational Technology, 2023). Crafting tailored exercises for specific topics is time-consuming and often limits the quantity and diversity of available practice opportunities. Moreover, accurately estimating a student's proficiency usually requires multiple assignments and manual grading, adding further workload for educators.

From the learner's perspective, traditional exercise sets are often designed for the "average student," overlooking prior knowledge, individual learning speeds, and personal weaknesses. As a result, students may receive exercises that are too easy, causing disengagement, or too difficult, leading to frustration. In emerging or highly specialized subjects, the lack of available exercises further restricts opportunities for targeted practice and knowledge assessment.

Edisconet, the project partner, aims to address this gap. Currently, their platform can only certify course completion without providing insights into a learner's true comprehension level. As a single-access platform integrating multiple learning management systems, Edisconet links training activities to business performance indicators and supports personal development with AI-driven solutions. Developing a method to automatically generate exercises that estimate learners' proficiency would allow Edisconet to go beyond course attendance tracking and deliver meaningful evaluations of learning success, strengthening the connection between education and measurable impact.

Ultimately, providing a high-quality educational experience requires a vast and diverse repository of exercises that span the full range of difficulty levels for every topic taught. To achieve the level of proficiency estimation envisioned by Edisconet, the system must offer a dynamic assessment environment that adapts to the individual learner's needs. However, manually creating such a voluminous and didactically varied body of work is an unrealistically time-consuming task for educators, creating a fundamental barrier to scalable, personalized learning.

1.2 Problem

The core problem lies in the manual burden and technical difficulty for creating authentic, varied assessments that accurately map to different proficiency levels. While Large Language Models (LLMs) present a promising avenue for overcoming the limitation of static exercise design, the effective integration of these models into a reliable educational pipeline is not straightforward.

Simply prompting a model to “generate questions” often results in content that is linguistically fluent but not didactically sound. Existing automated solutions frequently struggle with unstructured course materials, such as PDFs or slides, failing to preserve the necessary context or hallucinating information not present in the source text. Furthermore, there is a significant disconnect between technical text processing and didactic validity. A system is required that does not merely extract sentences but understands the cognitive depth required to test a novice versus an expert. The challenge is to engineer an automated pipeline that can ingest raw university-level course material and output exercises that are not only contextually relevant but also calibrated to varying levels of difficulty, such as those defined by Bloom’s Taxonomy or Webb’s Depth of Knowledge, without requiring extensive manual oversight from already overburdened educators.

1.3 State-of-the-Art and Knowledge Gap

The current body of literature offers various solutions for automated question generation, yet significant gaps remain regarding their application in higher education. Traditional Natural Language Processing (NLP) methods have focused heavily on generating Multiple-Choice Questions (MCQs). While widely researched, these approaches often rely on rigid rule-based systems or older statistical models that produce context-poor questions targeting only lower-order cognitive skills like rote memorization.

With the advent of Generative AI, recent studies have begun to explore the use of LLMs for educational content creation. Early attempts utilizing models like GPT-3.5 demonstrated potential but faced limitations regarding input structure. The pipelines often required structured data formats like XML or HTML to function correctly, which is rarely available in standard university lecture notes. Furthermore, while cognitive frameworks like Bloom’s Taxonomy are frequently cited, there is a lack of empirical research comparing how different frameworks perform when interpreted by modern state-of-the-art LLMs.

A critical knowledge gap exists in the intersection of unstructured document processing and high-level didactic alignment. There is currently no established “best practice” pipeline that combines effective context preparation, such as Sliding Window or Concept Extraction, with advanced reasoning models to generate open-ended, proficiency-estimating exercises from standard course PDFs. This thesis addresses that gap by moving beyond simple MCQ generation to explore how modern pipelines can create authentic, open-ended assessments.

1.4 Objectives and Research Question

The primary goal of this project is to develop and evaluate a method that leverages Large Language Models to automatically generate exercises tailored to specific course content. This approach is intended to significantly reduce the manual effort of teachers, thereby allowing them to focus more on direct student interaction and feedback.

The method is designed to dynamically generate exercises with adjustable difficulty levels, providing a foundation for personalized learning experiences. For a platform like Edisconet, this adaptability is crucial for moving beyond simple attendance tracking toward the certification of a learner’s actual proficiency level. By automating the production of diverse assessment pairs, the solution seeks to enhance both teaching efficiency and the measurable impact of digital learning environments.

To achieve this, the research is guided by the following central question:

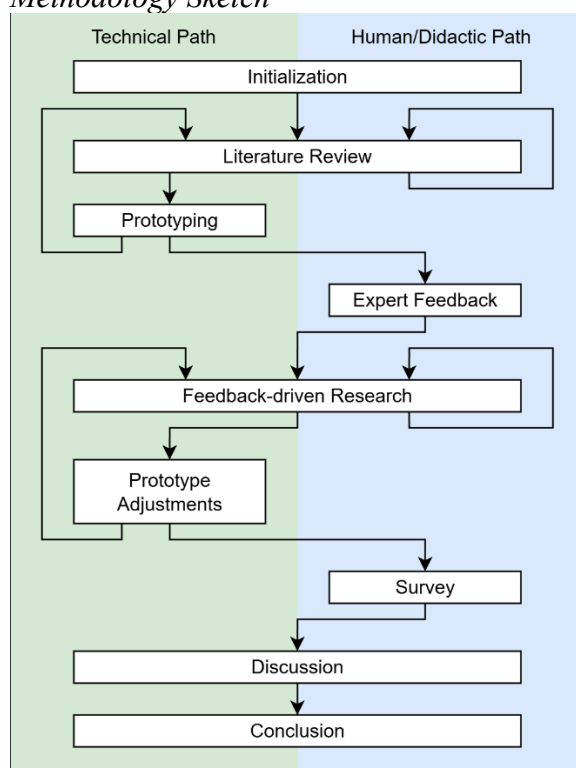
"What is the optimal LLM-based pipeline architecture for transforming unstructured university course materials into didactically sound, open-ended assessment pairs by empirically comparing context preparation methods and cognitive frameworks regarding their technical quality, cost-effectiveness, and expert-perceived utility."

To answer this, the study focuses on balancing breadth and depth of content coverage, ensuring didactic alignment through cognitive frameworks, and validating the practical feasibility and value for university lecturers.

1.5 Methodology

This project utilizes a mixed-methods approach to develop and evaluate an LLM-based system for automated exercise generation. As shown in Figure 1, the methodology integrates a Technical Path with a Human/Didactic Path to ensure both functional performance and educational quality.

Figure 1
Methodology Sketch



The process began with initialization, establishing the project scope and plan, requirements, and risk management strategies (Appendix 0, 9.2, and 9.3). This was followed by a systematic literature review to ground the development in current research regarding course material modeling, LLM frameworks, proficiency scales, and learning theories.

In the technical phase, four distinct pipelines were developed in Python as prototypes. These implementations compare two context-processing methods, Sliding Window and Concept Extraction, against two cognitive frameworks: Bloom's Revised Taxonomy and Webb's Depth of Knowledge. Following iterative refinements based on expert feedback from

project advisors, these prototypes were tested using authentic course documents from the Lucerne University of Applied Sciences and Arts.

The final validation phase involved a comprehensive survey of 21 module coordinators across seven institutes. These subject matter experts evaluated AI-generated questions specific to their curricula. This evaluation combined a quantitative win-rate mechanism to identify preferred pipelines with a qualitative analysis of lecturer feedback using Gioia's Data Structure. This dual-layered assessment ensures a rigorous evaluation of the method's technical feasibility and its practical didactic value.

1.6 Structure of the Thesis

The thesis is structured to guide the reader from the theoretical underpinnings to the practical application and evaluation of the proposed solution. Following this introduction, the Literature Review examines the current state of automated assessment, identifying the specific technological and didactical gaps that necessitate this research. The Methodology chapter details the construction of the four prototype pipelines, the specific algorithms used for text processing, and the design of the survey.

The Results chapter presents the quantitative data regarding generation time, cost analysis, and the win-rate preferences derived from the lecturer survey, alongside the qualitative themes extracted from expert feedback. The Discussion interprets these findings, contextualizing the trade-offs between cost, speed, and didactic quality, and answers the research question. Finally, the Conclusion summarizes the contributions, outlines the limitations, and provides actionable recommendations for the implementation of such systems in higher education.

2. Literature Review

This chapter establishes the theoretical and technical foundation required to develop an automated exercise-generation pipeline. While Chapter 1 introduced the problem of educator workload and the potential of AI, this review synthesizes existing research to identify where current solutions fail, specifically regarding didactic depth and the processing of unstructured university materials.

The primary objective of this review is to bridge the gap between pedagogical frameworks (how students learn) and technical architectures (how LLMs process information). By evaluating 58 core articles across five thematic areas, this chapter justifies the selection of specific cognitive taxonomies and document-processing methods used in the subsequent methodology.

2.1 Method of Literature Analysis

To establish a rigorous theoretical foundation, this study employed a systematic literature review supplemented by an explorative search. The review was structured around a central research question focused on identifying the optimal LLM-based pipeline for generating university-level proficiency assessments. This inquiry was divided into five thematic areas: existing automated tools, methods for modeling course material into discrete informational snippets, LLM-based exercise generation techniques, the components of effective higher education assessments, and didactical methods for maximizing learning outcomes.

The search strategy targeted peer-reviewed journals, conference proceedings, and highly cited preprints (with at least ten citations) published in English or German since 2018. Across these five themes, 25 distinct search queries were executed across the databases detailed in Table 1.

Table 1
Searched Databases and Filters

Database	Sub Questions	Publication Years	Citations	Document Type	Language
Web of Science	1, 2, 3, 4, 5	2018-2025	≥ 10	Article	English or German
IEEE Xplore	1, 2, 3	2018-2025	≥ 10	Conferences & Journals	-
Google Scholar	4, 5	2018-2025	≥ 10	-	-

Initial screening of titles and abstracts from the top results yielded 694 articles. After filtering for relevance, sorting by citation count, and removing duplicates, the pool was narrowed to 256 articles. From this set, the 20 most relevant papers per sub-question underwent full-text screening, resulting in the 58 core articles summarized in Table 2.

Table 2
Search Queries and Article Counts

Sub Question	Query	Exports	Filtered Exports	Chosen Articles
1	"large language model*" AND ("education" OR "proficiency") AND "assessment"	26		
	"AI-based exercise generation" OR "automatic question generation"	21		
	"proficiency estimation" AND ("LLM" OR "language model")	10	79	20
	"adaptive learning" AND ("university" OR "higher education") AND "AI"	82		
	"pipeline" AND ("exercise generation" OR "assessment") AND "LLM"	2		
2	"course material modeling" AND ("snippet" OR "fragment" OR "knowledge unit")	0		
	"knowledge graph" AND "education" AND "content modeling"	5		
	"semantic segmentation" AND "text" AND ("education" OR "course")	13	31	5
	"educational content representation" AND ("higher education" OR "university")	2		
	"topic modeling" AND "course materials" AND "education"	11		
3	"automatic question generation" AND ("LLM" OR "GPT" OR "BERT")	0		
	"prompt engineering" AND "exercise generation" and "education"	0		
	"LLM-based question generation" AND ("university" OR "higher education")	4	4	1
	"text-to-question" OR "text-to-exercise" AND "machine learning"	0		
	"assessment item generation" AND "AI" AND "Bloom* Taxonomy"	0		
4	"assessment design" AND "higher education"	75		
	"components of effective assessment" AND ("university" OR "higher education")	6		
	"assessment framework" AND ("validity" OR "reliability") AND ("higher education" OR "university")	89	82	19
	"AI-based assessment" AND ("higher education" OR "university")	26		
	"formative assessment" AND "summative assessment" AND ("higher education" OR "university")	86		
5	"didactic principles" AND "exercise" AND ("higher education" OR "university")	23		
	"instructional design" AND "exercises" AND ("university" OR "higher education")	69		
	"learning outcomes" AND "exercise-based learning" AND ("higher education" OR "university")	17	60	13
	"active learning" AND "exercises" AND ("higher education" OR "university")	105		
	"evidence-based teaching methods" AND "exercises" AND ("higher education" OR "university")	22		
Total		694	256	58

To address a lack of sufficient data for material modeling and generation techniques, an additional explorative review was conducted to ensure a minimum of ten relevant articles for each category. Finally, specific research into cognitive frameworks was integrated to provide a basis for defining difficulty levels within the generated exercises.

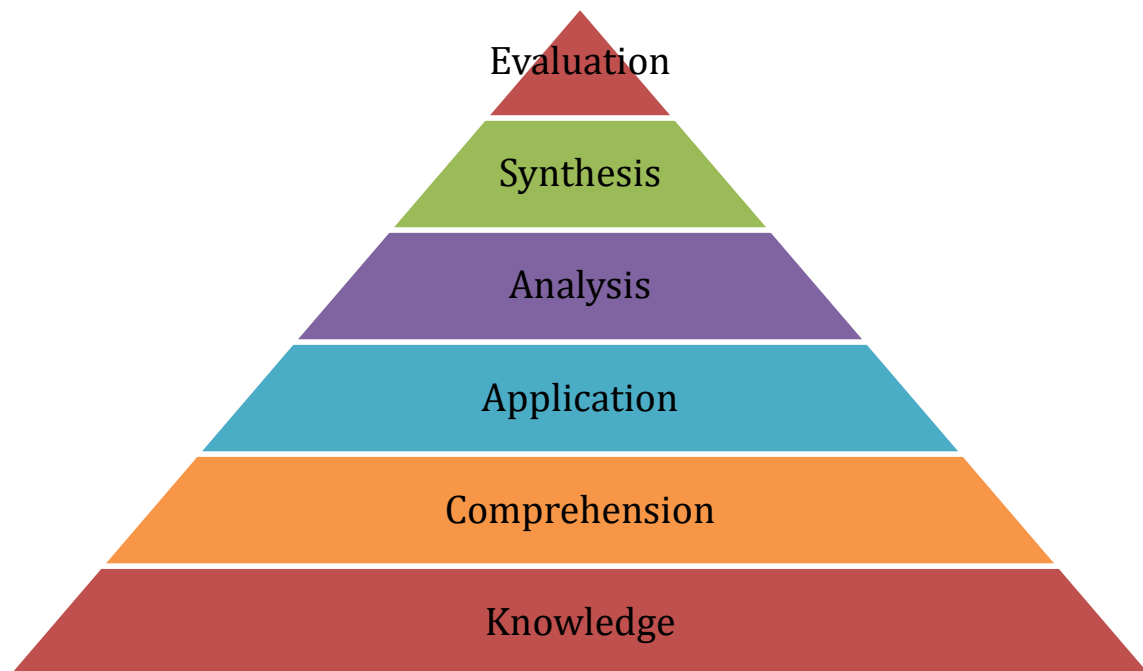
2.2 Cognitive Frameworks

2.2.1 Bloom's Original Taxonomy

The original taxonomy developed by Benjamin S. Bloom et al. (1956) provides a standardized framework for classifying educational outcomes, moving beyond vague concepts like “thinking” to facilitate precise communication among educators and researchers. As illustrated in Figure 2, the framework is structured as a cumulative hierarchy of six levels, where each stage necessitates mastery of the previous one to engage in increasing complex cognitive tasks.

Figure 2

Bloom's Original Taxonomy Hierarchy



The foundation, Knowledge, focuses on the psychological process of remembering and recalling specific information, structures, or methods. Building upon this is Comprehension, which represents the most basic form of understanding. Here, a student can grasp and use communication without necessarily connecting it to broader implications. Application advances this by requiring the use of abstractions, such as rules, procedures, or technical principles, within concrete and particular situations.

Higher-order cognitive skills begin with Analysis, the systematic breakdown of a communication into its constituent parts to clarify the relationships and organizational principles between ideas. This is followed by Synthesis, where students combine disparate elements to form a new, original pattern or structure. The hierarchy culminates in Evaluation, involving the qualitative or quantitative judgment of materials and methods against internal or external criteria (Benjamin S. Bloom et al., 1956).

Ultimately, this progression serves as a critical planning tool for educators, ensuring that curricula foster sophisticated cognitive skills by requiring a student to know, understand, apply, analyze, and synthesize information before they can effectively evaluate it (Anderson, 2009).

2.2.2 Bloom's Revised Taxonomy

The revised version of Bloom's Taxonomy by Anderson (2009) incorporates empirical research and modern cognitive learning theories to better reflect how students process information. One significant structural change is the reversal of the Synthesis and Evaluation levels. The authors argued that Create (formerly Synthesis) involves inductive thinking, which represents a more complex cognitive task than deduction. The terminology was also updated to be more intuitive and action-oriented: Knowledge became Remember, Comprehension was renamed Understand, and Synthesis became Create. Furthermore, the strict cumulative hierarchy of the original model was softened, acknowledging that while lower-level skills support higher-order thinking, absolute mastery of one level is not always a prerequisite for the next.

To address the overlap between types of knowledge and cognitive activities in the original framework, the revision introduces a two-dimensional matrix. This separates the Cognitive Process Dimension from the Knowledge Dimension, which spans from concrete to abstract categories: Factual, Conceptual, Procedural, and Metacognitive Knowledge. This dual-axis approach, illustrated in Figure 3, allows for a more granular classification of education objectives by mapping what a student learns against how they mentally process that information.

Figure 3

Bloom's Revised Taxonomy Matrix

The Knowledge Dimensions	The Cognitive Process Dimensions					
	Level 1 <i>Remember</i>	Level 2 <i>Understand</i>	Level 3 <i>Apply</i>	Level 4 <i>Analyze</i>	Level 5 <i>Evaluate</i>	Level 6 <i>Create</i>
A. Factual Knowledge						
B. Conceptual Knowledge						
C. Procedural Knowledge						
D. Metacognitive Knowledge						

Source: Anderson (2009)

2.2.3 SOLO Taxonomy

The Structure of Observed Learning Outcomes (SOLO) taxonomy, proposed by Biggs et al. (1982), adapts Piagetian developmental stages to evaluate the quality and depth of a learner's understanding. Unlike the process-based Bloom's Revised Taxonomy, SOLO focuses on observable learning outcomes to both assess student performance and design targeted assessment questions. The framework is defined by three primary components: Capacity, referring to the required working memory; Relating Operation, describing how cues and responses interrelate; and the balance between Consistency and Closure, which represents the tension between a learner's need for a definitive conclusion and the need to avoid logical contradictions.

The taxonomy, as shown in Figure 4, consists of five hierarchical levels of increasing complexity. At the Prestructural level, the learner misses the point or avoids the question entirely. Unistructural responses rely on a single relevant aspect of the data, leading to limited or dogmatic conclusions, while Multistructural responses incorporate several relevant aspects but fail to integrate them, often ignoring internal inconsistencies. Higher-level thinking begins at the Relational stage, where the learner integrates all evidence into a coherent system within the given context. Finally, at the Extended Abstract level, the learner generalized the information to new situations, testing hypotheses and keeping conclusions open to alternative possibilities. Between these levels, learners may exhibit Transitional Responses, characterized by the confusion or inconsistency that often precedes a shift to a more sophisticated cognitive stage.

Figure 4
SOLO Taxonomy Hierarchy

Developmental base stage with minimal age	SOLO description	1 Capacity	2 Relating operation	3 Consistency and closure	4 Response Structure
Formal Operations (16+ years)	Extended Abstract	<i>Maximal:</i> cue + relevant data + interrelations + hypotheses	Deduction and induction. Can generalize to situations not experienced	Inconsistencies resolved. No felt need to give closed decisions—conclusions held open, or qualified to allow logically possible alternatives. (R ₁ , R ₂ , or R ₃)	
Concrete Generalization (13-15 years)	Relational	<i>High:</i> cue + relevant data + interrelations	Induction. Can generalize within given or experienced context using related aspects	No inconsistency within the given system, but since closure is unique so inconsistencies may occur when he goes outside the system	
Middle Concrete (10-12 years)	Multistructural	<i>Medium:</i> cue + isolated relevant data	Can "generalize" only in terms of a few limited and independent aspects	Although has a feeling for consistency can be inconsistent because closes too soon on basis of isolated fixations on data, and so can come to different conclusions with same data	
Early Concrete (7-9 years)	Unistructural	<i>Low:</i> cue + one relevant datum	Can "generalize" only in terms of one aspect	No felt need for consistency, thus closes too quickly: jumps to conclusions on one aspect, and so can be very inconsistent	
Pre-operational (4-6 years)	Prestructural	<i>Minimal:</i> cue and response confused	Denial, tautology, transduction. Bound to specifics	No felt need for consistency. Closes without even seeing the problem	

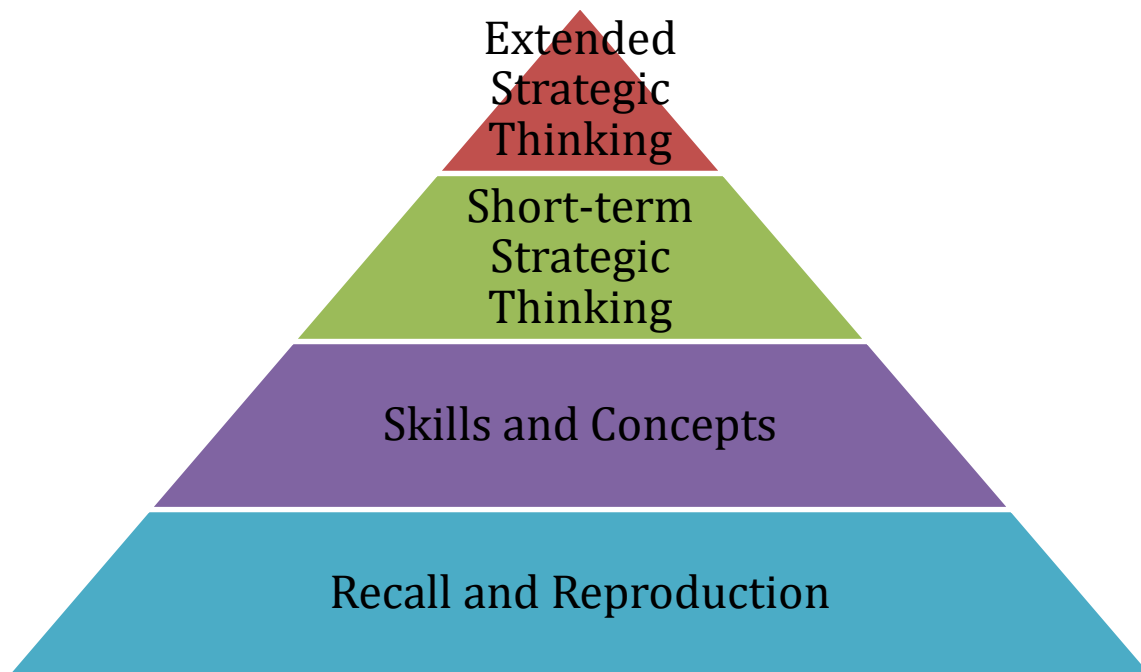
Source: Biggs et al. (1982)

2.2.4 Webb's Depth of Knowledge

Norman L. Webb (1997) originally developed this framework to analyze the alignment between educational standards and standardized assessments, later expanding it to include specific applications for core academic subjects (Norman L. Webb, 2002). The model operates on the principle that curricular elements can be categorized by the cognitive demands required to produce a valid response. Crucially, these levels focus on the complexity

of the cognitive processes demanded by a task rather than its inherent difficulty. As visualized in Figure 5, the model comprises four distinct levels: Recall and Reproduction, Skills and Concepts, Short-term Strategic Thinking, and Extended Thinking (Mississippi Department of Education, 2009).

Figure 5
Webb's Depth of Knowledge



The first level, Recall and Reproduction, involves basic tasks such as recalling facts, formulas, or procedures with minimal mental transformation of the target knowledge. Moving to the second level, Skills and Concepts, require mental processing that goes beyond simple recall; students must contrast, classify, or explain relationships and cause-and-effect patterns.

The third level, Short-term Strategic Thinking, demands the application of higher-order processes like analysis and evaluation to solve real-world problems with predictable outcomes. A defining marker of this level is the requirement for students to state their reasoning while coordinating knowledge from multiple subject areas. Finally, Extended Strategic Thinking involves the sustained use of complex thinking processes, such as synthesis and reflection, over longer periods. At this highest level, students conduct investigations to solve real-world problems with unpredictable outcomes, requiring constant assessment and adjustment of their strategic plans (Mississippi Department of Education, 2009).

2.2.5 Five-Stage Model of Adult Skill Acquisition

Effective skill training must be grounded in a model of skill acquisition to address the specific needs of learners at different developmental phases (Stuart E. Dreyfus & Hubert L. Dreyfus, 1980). This model posits that learners progress through distinct stages of mental processing as they gain experience. Initially, a Novice relies on context-free, non-situational features to perform tasks, recognizing patterns without prior experience. Progressing toward Competence requires actual exposure to real-world situations, where learners begin to identify recurrent meaningful patterns that are no longer context-free but situational.

As practice increases, a learner reaches Proficiency, where they view whole situations through the lens of long-term goals. At this stage, the performer uses memorized principles, or maxims, to determine appropriate actions based on the salience of different situational aspects. Expertise marks the transition to intuitive processing. The expert's vast repertoire of experience allows them to bypass analytical rules and immediately grasp the appropriate action for a specific situation. Finally, Mastery is characterized by moments of intense absorption where the expert ceases conscious monitoring, allowing mental energy to flow entirely into instantaneous, high-level performance. These original model characteristics are visualized in Figure 6.

Figure 6

Dreyfus' original 5-stage Model

Skill Level Mental Function	NOVICE	COMPETENT	PROFICIENT	EXPERT	MASTER
Recollection	Non-situational	Situational	Situational	Situational	Situational
Recognition	Decomposed	Decomposed	Holistic	Holistic	Holistic
Decision	Analytical	Analytical	Analytical	Intuitive	Intuitive
Awareness	Monitoring	Monitoring	Monitoring	Monitoring	Absorbed

Source: Stuart E. Dreyfus & Hubert L. Dreyfus (1980)

In 1986, the authors revised this framework to create a more distinct progression (Dreyfus et al., 1986). The revision introduced an Advanced Beginner stage at Level 2 to better bridge the gap between a novice's context-free rules and a competent learner's situational coping. This shifted Competence to Level 3, now defined by the ability to plan and troubleshoot. Additionally, the Master level was consolidated into the Expert stage at Level 5, identifying intuitive, absorbed performance as the hallmark of expertise rather than a separate category. This standardized the widely adopted hierarchy: Novice, Advanced Beginner, Competent, Proficient, and Expert (Dreyfus et al., 1986).

In the year 1986, the authors revised their original work, where they restructured the intermediate and final stages to create a more distinct progression. The original listed Competence as the second stage, occurring only after considerable experience. In the revision, a new stage called Advanced Beginner was inserted at Level 2 to bridge the gap between the context-free rules of the Novice and the complex situational coping of the Competent performer. Consequently, Competence shifted to Level 3, defining a learner who can now plan and troubleshoot based on recurrent meaningful component patterns.

The second major change was the removal of the Master level as a distinct fifth stage. The revised model consolidates it into the Expert stage at Level 5, viewing this intuitive, absorbed performance as the defining characteristic of an Expert rather than a separate Master category. This standardized the final and more widely adopted hierarchy to: Novice, Advanced Beginner, Competent, Proficient, and Expert (Dreyfus et al., 1986).

2.2.6 Cognitive Framework Comparison

The transition from Bloom's Original Taxonomy to the revised version represents a fundamental shift from classifying educational outcomes to modeling cognitive complexity. While the original framework focused on what a student achieves, the revision emphasizes

how the mind processes knowledge, moving from a product-oriented view to a process-oriented one.

A comparison with the SOLO Taxonomy reveals that while Bloom describes the degree to which knowledge is used, SOLO measures the degree to which it is integrated and structured. The Prestructural level of SOLO sits below Bloom's Remember level, as it indicates a total absence of recall. From the Unistructural level onward, SOLO effectively bypasses simple retrieval to focus on the depth and connectedness of understanding that occurs between Bloom's Understand and Evaluate stages.

Webb's Depth of Knowledge (DOK) aligns with the Bloom's Revised Taxonomy by addressing the cognitive process, yet they diverge in their core questions. Blooms classifies which process is engaged, whereas Webb categorizes the complexity or depth of that process. These frameworks can be mapped such that DOK Level 1 corresponds to Remembering, Level 2 to Understanding and Applying, Level 3 to Analyzing and Evaluating, and Level 4 to Creating. This relationship is further refined in Hess's Cognitive Rigor Matrix (simplified in Table 3), which demonstrates that certain higher-order tasks, like evaluation, require the specific strategic depths found only in Webb's third and fourth levels (Karin K. Hess et al., 2009).

Finally, the Five-Stage Model of Adult Skill Acquisition shares a qualitative and goal-oriented foundation with both the original Bloom and SOLO taxonomies. It is particularly close to SOLO in its focus on the depth of understanding rather than isolated cognitive acts. While a strict one-to-one mapping between Dreyfus's stages and the other frameworks is context-dependent, they all follow a similar developmental logic that rewards integration and intuition over rote execution. A comprehensive overview of these alignments is provided in the comparative summary in Table 4.

Table 3
Simplified Cognitive Rigor Matrix

Bloom	Webb Level 1	Webb Level 2	Webb Level 3	Webb Level 4
Remember	Recall facts or definitions	-	-	-
Understand	Define or locate steps	Explain relationships	Connect or generalize ideas	Relate concepts across domains
Apply	Follow rules or procedures	Solving routine problems	Solving non-routine problems	Design complex projects
Analyze	Identify parts or details	Categorize or organize data	Draw conclusions or cite evidence	Analyze multiple sources
Evaluate	-	-	Justify, critique, or verify	Evaluating complex themes
Create	Brainstorm ideas	Generate hypothesis	Develop original models	Synthesize or design systems

Table 4
Cognitive Framework Comparison

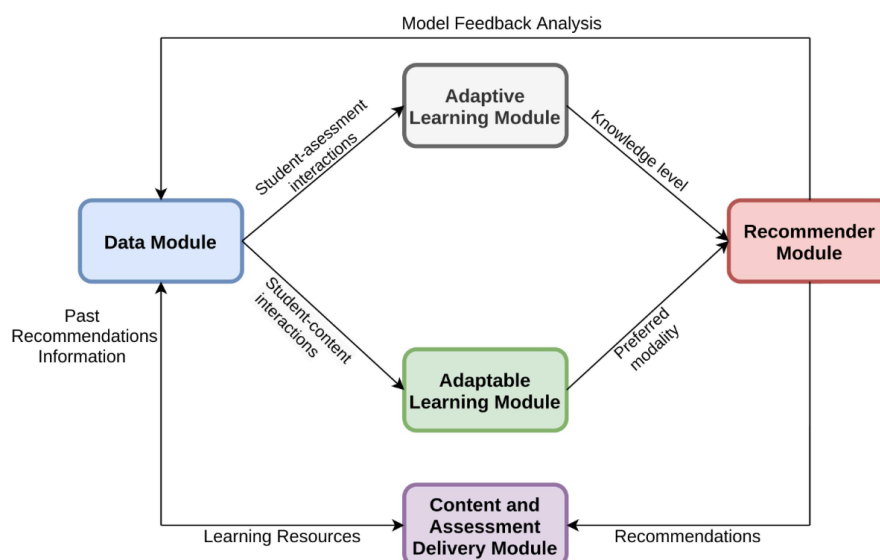
Framework	Original Bloom	Revised Bloom	SOLO	Webb's DOK	Original Dreyfus	Revised Dreyfus
Year	1956	2001	1982	1997/2002	1980	1986
Focus	Outcome	Action	Outcome	Demand	Competence	Competence
Cognition	Process	Process	Complexity	Complexity	Complexity	Complexity
Level 1	Knowledge	Remember	Prestructural	Recall & Reproduction	Novice	Novice
Level 2	Comprehension	Understand	Unistructural	Skills & Concepts	Competent	Advanced Beginner
Level 3	Application	Apply	Multistructural	Strategic Thinking	Proficient	Competent
Level 4	Analysis	Analyze	Relational	Extended Thinking	Expert	Proficient
Level 5	Synthesis	Evaluate	Extended Abstract	-	Master	Expert
Level 6	Evaluation	Create	-	-	-	-

2.3 Existing Tools and Comparison

2.3.1 Theoretical Frameworks for Personalized Assessment

Accurately estimating a learner's proficiency requires automated pipelines that move beyond simple correctness checks toward personalized and adaptive experiences. These methods foster immersive environments and enable the assessment of higher-order thinking skills, addressing the scalability challenges traditional instructional designs face regarding educator oversight. While large-scale AI applications are increasingly utilized for predictive analytics and adaptive learning paths (T. Wang et al., 2023), their success depends on a robust pedagogical foundation. A foundational model for personalized e-learning integrates five essential components into automated systems (Murtaza et al., 2022): a data module, adaptive and adaptable learning modules, a recommender module, and a delivery module, as illustrated in Figure 7.

Figure 7
Proposed Framework for Personalized E-Learning



Source: Murtaza et al. (2022)

To ensure didactical validity, the generation of exercises within an LLM-based pipeline must align with established learning theories. This requires prompts designed to address behaviorist task-based learning and cognitivist problem-solving, alongside constructivist principles that build on prior knowledge and humanist approaches that align with learner interests. Furthermore, the pipeline should map generated exercises to recognized instructional design models, such as Bloom's Taxonomy, Gagne's Nine Events of Instruction, or Merrill's Principles, to maintain educational rigor.

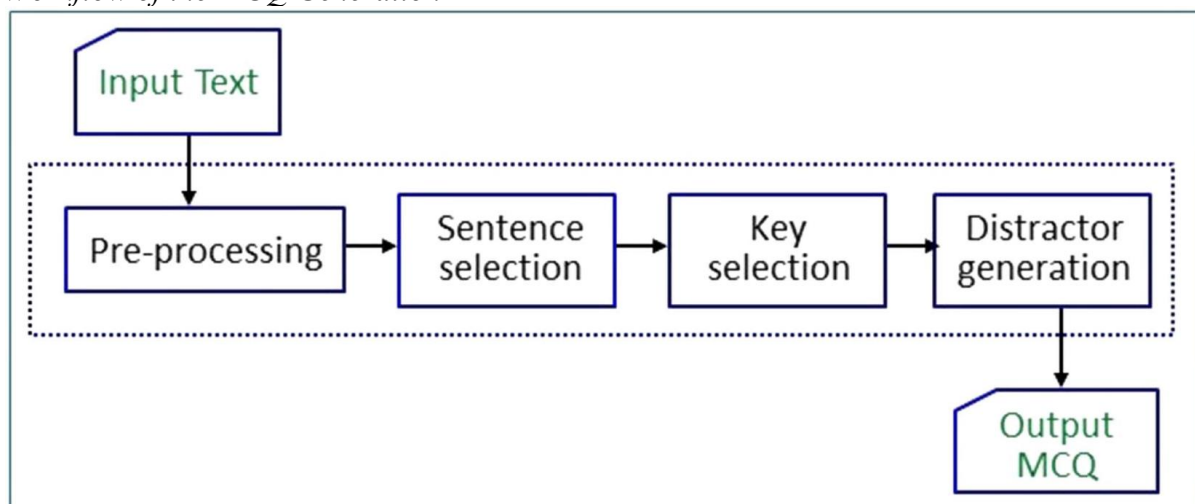
The system must further distinguish between adaptivity and adaptability to accurately support the learner. Adaptivity involves delivering content based on comprehension levels through methods like knowledge tracing or item response theory. Conversely, adaptability focuses on the delivery modality, such as choosing between video or text, to ensure environmental or personal factors do not hinder the student. Ultimately, these components rely on a continuous data collection pipeline. By retrieving assessment results and learning patterns, the system can utilize collaborative, content-based, or hybrid recommendation strategies, ensuring the LLM-based pipeline functions as part of a data-driven feedback loop rather than a static text generator (Murtaza et al., 2022).

2.3.2 Traditional NLP-based Generation Pipeline

Automated question generation systems have previously focused on MCQs through complex NLP workflows. Ch & Saha (2020) established a generic workflow beginning with extensive pre-processing, including text normalization, structural analysis, and coreference resolution to map pronouns to their respective nouns. This process involves simplifying sentences to remove unintended clues and utilizing lexical and syntactic analysis to identify "questionable facts". Once a fact is selected, the system reforms the sentence into an interrogative type and generates distractors based on the initial pre-processing data.

In a subsequent study, Ch & Saha (2023) refined this approach by incorporating automatic context-free grammar extractors and dependence parsers to simplify input text. This refined method, illustrated in Figure 8, uses Jaccard similarity and Word2Vec to identify the most plausible distractors. Although this system produces sensible questions 40 times faster than manual writing, it remains limited by the need for manually pre-defined key terms and reference sentences for each new text. Furthermore, while MCQs offer reliable and quick scoring, they are often criticized for testing only lower-order cognitive levels and being susceptible to student guessing.

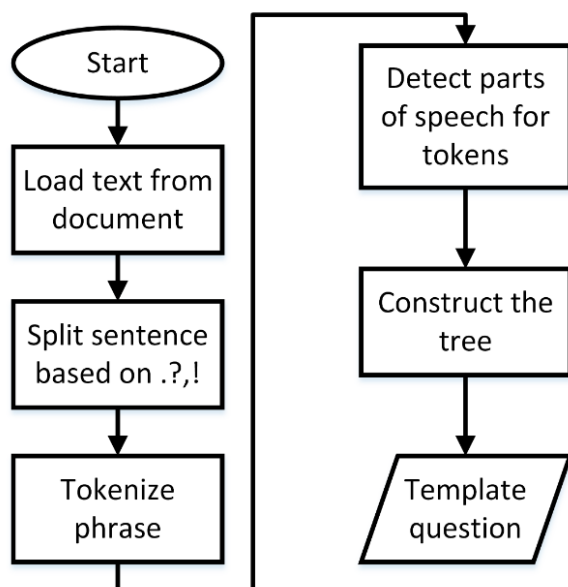
Figure 8
Workflow of the MCQ Generation



Source: Ch & Saha (2023)

To address these limitations, research has expanded to include diverse question types such as true/false, fill-in-the-blank, and Wh-questions. Killawala et al. (2018) introduced a pipeline that utilizes Long Short-Term Memory (LSTM) models to classify which question types are most suitable for a given text segment. While true/false questions share the cognitive limitations of MCQs, Wh-questions allow for a deeper assessment of a learner's comprehension. However, as shown in the training process in Figure 9, this deep learning approach requires vast amounts of domain-specific training data to achieve generalizability across different academic fields.

Figure 9
Training process of Question Generation



Source: Killawala et al. (2018)

2.3.3 Generative AI and LLM-based Approaches

Generative AI, and LLMs in particular, offer significant potential to enhance virtual tutoring systems, stimulate creativity, and drive educational innovation (Neumann et al.,

2023). These models have already been integrated into various learning and research environments through the use of chatbots (T. Wang et al., 2023).

Lee et al. (2024) demonstrated this potential by developing an automated pipeline for generating English reading exercises using OpenAI's GPT-3. The process involves selecting a passage of fewer than 250 words, determining a specific task type and question format, and feeding these alongside a corresponding prompt into the LLM. The resulting output is then reviewed for use in instructional settings.

Figure 10 illustrates the diverse matrix of question types and reasoning levels, ranging from literal identification to complex inferential summaries, that such a model can generate.

Figure 10

Matrix about the type of Questions that ChatGPT can generate

		Contents		Reasoning								
		literal(identify, check)		inferential								
		Concordance (non-literature)	Diagrams	Topic (non-literature)	Purpose (non-literature)	Infering content (non-literature)	Mood, atmosphere (literature)	Summary, main idea and claim (non-literature)	blank	irrelevant sentence	Inserting a sentence	Sequence
y-n/ alternative/ t-f	multiple-choice	A1		A2	A3	A4	A5	A6				
	open-ended	B1		B2	B3	B4	B5	B6				
wh-q	multiple-choice	C1		C2	C3	C4	C5	C6		C8	C9	C10
	open-ended	D1		D2	D3	D4	D5	D6				
cloze	multiple-choice			E2		E4	E5	E6				
	open-ended			F2		F4	F5	F6				

Source: Lee et al. (2024)

While this LLM-based method is considerably less complex than traditional NLP workflows, it remains partially manual, as it requires human intervention to select the text passages, formats, and question types for each iteration.

2.3.4 Existing Commercial Adaptive Platforms

Several established adaptive learning platforms utilize proprietary pipelines to estimate student proficiency and tailor educational content. A comparative study by Dutta et al. (2024) analyzed four leading systems, noting that while they offer sophisticated personalization, they often face limitations regarding cost and disciplinary scope.

Carnegie Learning employs real-time data analysis and knowledge spaces to map student understanding, allowing learners to progress only after mastering specific concepts. While the platform aims to close the gap between high- and low-performing students, evidence supporting this relative impact remains limited. Furthermore, its offerings are

primarily focused on mathematics and science. Similarly, DreamBox Learning uses cognitive data to build personalized pathways but is restricted to K-8 mathematics, though it successfully incorporates gamification to reduce student anxiety.

Smart Sparrow emphasizes active learning and spaced repetition to enhance knowledge retention, offering diverse multimedia content and accessibility features like text-to-speech. However, it requires significant financial investment and specific technical infrastructure. Finally, Knewton leverages AI to shift the focus from assignment completion to competency-based progress, though it remains largely centered on STEM subjects.

Ultimately, the high cost and narrow subject scope of these commercial platforms often make them unsuitable for the diverse requirements of higher education modules. Nevertheless, integrating their successful features, such as gamification, competency tracking, and spaced repetition, could significantly enhance the value of an automated question-generation pipeline within a broader e-learning ecosystem.

2.3.5 Challenges and Limitations of Automated Methods

The implementation of automated and personalized learning platforms faces significant hurdles, particularly in feature identification and the generation of adaptable content across various modalities. A core technical difficulty lies in knowledge tracing: mapping incorrect assessment responses to specific knowledge gaps is complex when multiple concepts are involved, just as correct responses do not always guarantee complete comprehension. Furthermore, continuous assessment systems must manage constantly shifting datasets of user logs and learning patterns, making it difficult to determine optimal intervals for data computation. Successfully eliciting and maintaining an accurate profile of learner preferences over time remains a persistent challenge for these systems (Murtaza et al., 2022).

Beyond these technical aspects, the broader use of AI in education introduces critical concerns regarding privacy, algorithmic bias, and the potential to widen existing educational inequalities (Neumann et al., 2023; T. Wang et al., 2023).

LLMs specifically present unique feasibility challenges. Local deployments require substantial computational resources and storage for both training and inference. There is also a significant need for robust credibility and quality assessments to ensure that both students and educators can trust the model's feedback. While LLMs can alleviate teacher workloads through personalized instructional assistance, they necessitate a shift in teacher professional development to ensure effective collaboration between human and machine (Gan et al., 2023). Ultimately, because LLM accuracy is never guaranteed, the final responsibility for verifying the quality and correctness of generated content remains with the instructor or the academic institution.

2.4 Course Material Modeling

2.4.1 Content Selection and Pre-Processing

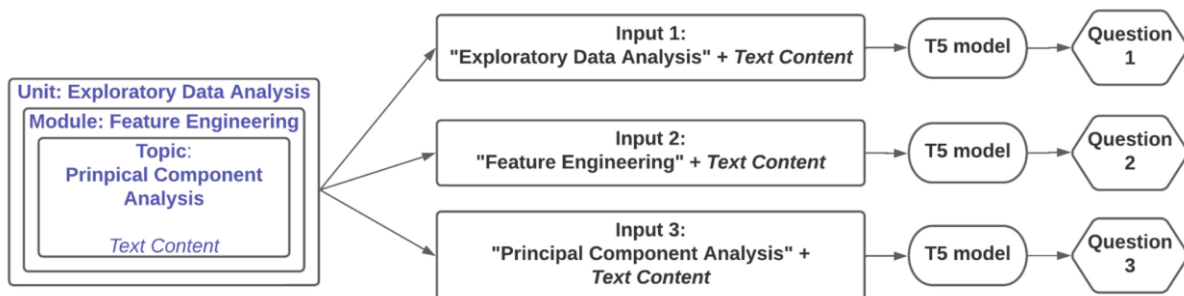
Before modeling can occur, raw course materials must be meticulously selected and cleaned. Educators often struggle with outdated materials, time constraints, and limited content knowledge when updating curricula. To address this, beginner-friendly reading materials for technical fields should be sourced from authentic, up-to-date publications like IEEE Spectrum or National Geographic. These materials must maintain an appropriate vocabulary difficulty, prepare students for future academic tasks, and provide autonomy by allowing both teachers and students to influence content selection (John & Devi, 2021).

Utilizing structured text formats, such as XML or HTML, simplifies pre-processing by preserving the original content hierarchy. Nguyen et al. (2022) demonstrated this by prepending headers to paragraphs to provide a T5 transformer model with necessary

conceptual context. As illustrated in Figure 11, this structured approach enables the model to generate specific questions mapped to various levels of the course architecture, from broad units to specific topics.

Figure 11

Example Question Generation Process for the Text Content in one Topic



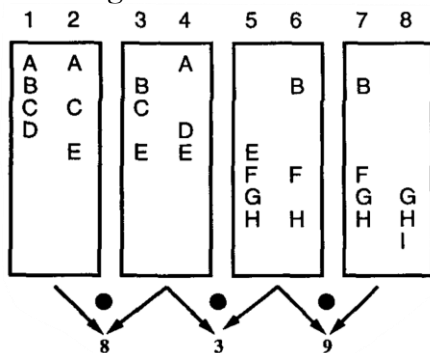
Source: Nguyen et al. (2022)

In their study, a GPT-3 model fine-tuned for binary classification was used to filter these outputs for "didactical soundness," defined as questions that accurately assess domain knowledge relevant to the course. However, significant discrepancies in quality remain; while the AI identified 151 of 203 generated questions as sound, human experts approved only 99 of those. With approximately 65% of generated questions proving practically useful, these findings underscore the ongoing necessity for human oversight in evaluating AI-generated exercises before they are deployed in a classroom setting.

2.4.2 Structural Segmentation

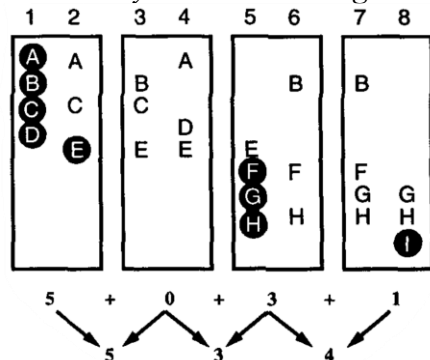
While paragraphs are ideally written as coherent, self-contained units with distinct topic and summary sentences, real-world texts often fail to meet these expectations (Marti A. Hearst, 1997). Consequently, effective content modeling requires algorithms capable of dividing text into multi-paragraph passages or subtopic segments. To address this, Hearst (1997) introduced "TextTiling," a framework comprising three distinct algorithms, Blocks, Vocabulary Introductions, and Lexical Chains, that identify topic shifts by assigning scores to sentence-sized units.

The Blocks algorithm identifies subtopic shifts by comparing the lexical similarity of adjacent text blocks. Each sentence unit is represented as a vector of term frequencies, and the inner product between adjacent blocks is calculated and normalized to produce a lexical score. As illustrated in Figure 12, a significant drop in this score, flanked by high similarity scores, suggests a vocabulary shift and a likely subtopic boundary. Notably, Hearst found that weighting words based on their frequency within the specific block proved more robust for these comparisons than traditional TF-IDF metrics.

Figure 12*Block Algorithm to calculate the Lexical Score*

Source: Marti A. Hearst (1997), Note: Dot product of vectors of word counts in neighboring blocks.

The Vocabulary Introductions algorithm utilizes type-token curves to track how many unique words appear for the first time within a moving 40-word window. As shown in Figure 13, sharp upturns in the frequency of new terms following deep valleys correlate closely with information flow and constituent boundaries. While this method is effective for tracking the introduction of new concepts, it sometimes lags by one or two sentences, as paragraph onsets are not always immediately signaled by new vocabulary.

Figure 13*Vocabulary Introduction Algorithm to calculate the Lexical Score*

Source: Marti A. Hearst (1997), Note: Number of words that occur for the first time at sentence gaps.

Finally, the Lexical Chains algorithm measures the extent of subtopics based on the repetition of terms or their morphological variants. Instead of relying on a single chain, this approach, illustrated in Figure 14, analyzes multiple active chains simultaneously across a text. Segment boundaries are identified at sentence gaps where chains begin, end, or return after a hiatus. Although both the Blocks and Vocabulary Introduction algorithms significantly outperform randomized baselines, their performance remains below that of human judges, highlighting the inherent limitations of rule-based lexical analysis for complex topic segmentation.

Figure 14

Chains Algorithm to calculate the Lexical Score

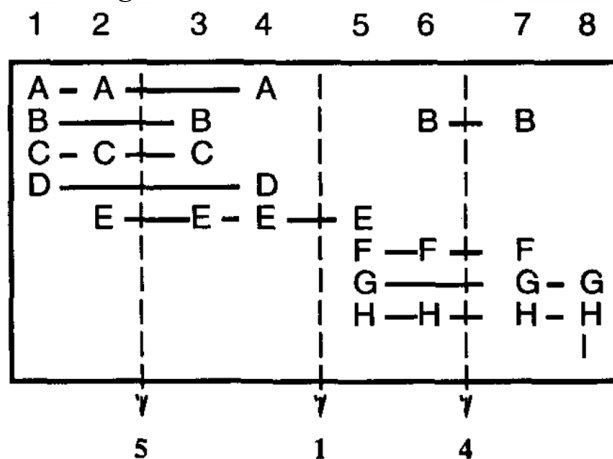
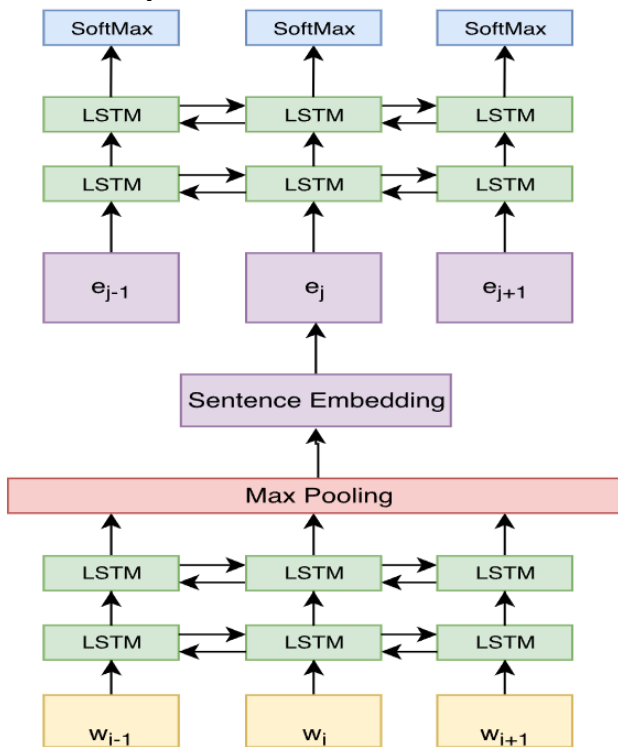


Figure 15

Dual two-layer bidirectional LSTM Architecture for Topic Segmentation

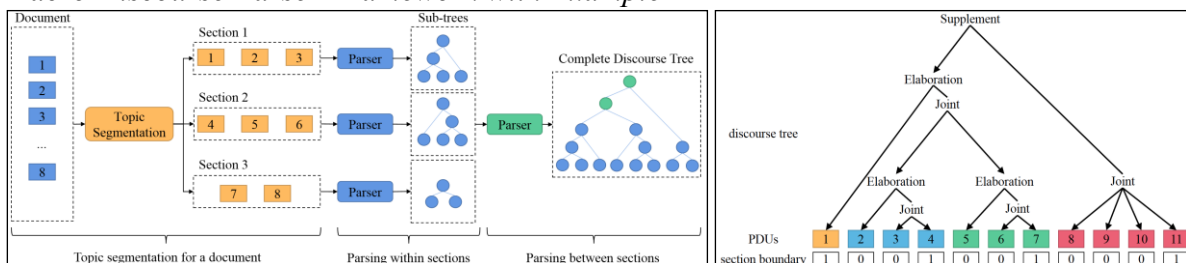


Source: Koshorek et al. (2018), Note: Model contains a sentence embedding sub-network, followed by a segmentation prediction sub-network to predict cut-off probabilities.

Moving beyond linear text splitting, Jiang et al. (2021) introduced a discourse parsing framework that constructs a macro-structure tree using implicit inter-paragraph boundaries derived from topic segmentation. As illustrated in Figure 16, this pipeline systematically parses a document by segmenting topics and then analyzing the discourse relationships within and between those sections.

Figure 16

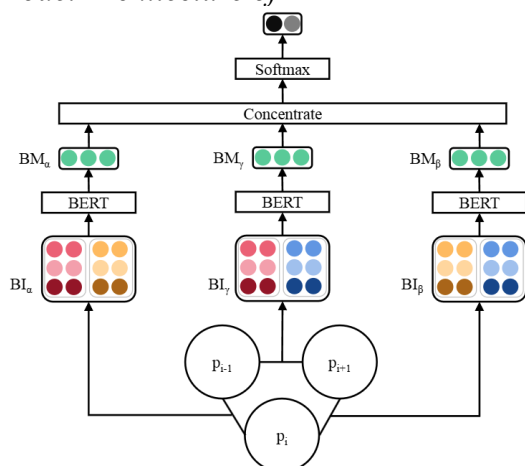
Macro Discourse Parser Framework with Example



Source: Jiang et al. (2021)

The core of this approach is a triple semantic matching model based on BERT, known as TM-BERT. This model utilizes a sliding window mechanism covering three consecutive paragraphs to predict whether a specific paragraph serves as a topic boundary. As shown in Figure 17, the architecture encodes word positions, segment identifiers, and relative paragraph positions for all three possible pairs within the window. These inputs are processed through BERT to generate semantic matching outputs, which a decoding layer then concentrates to calculate boundary probabilities via a softmax function.

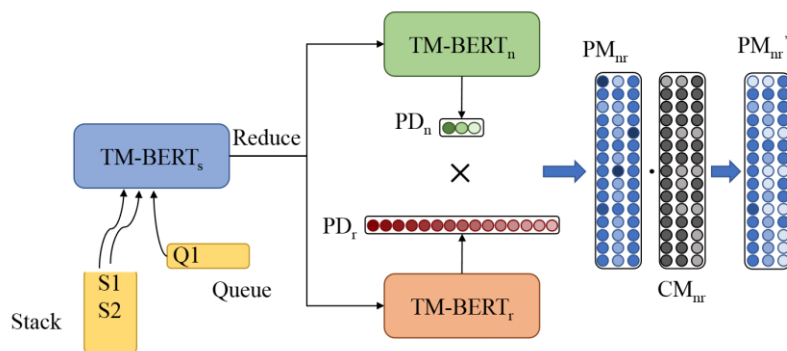
Figure 17
Model Architecture of TM-BERT



Source: Jiang et al. (2021)

To build the final discourse structure tree, the framework employs a shift-reduce algorithm. At each step, a structure classifier TM-BERT_s determines whether to generate a new span. Once a span is created, a nuclearity classifier TM-BERT_n and a relation classifier TM-BERT_r predict the respective labels for that span, as visualized in Figure 18.

Figure 18
Process of Macro Discourse Parser

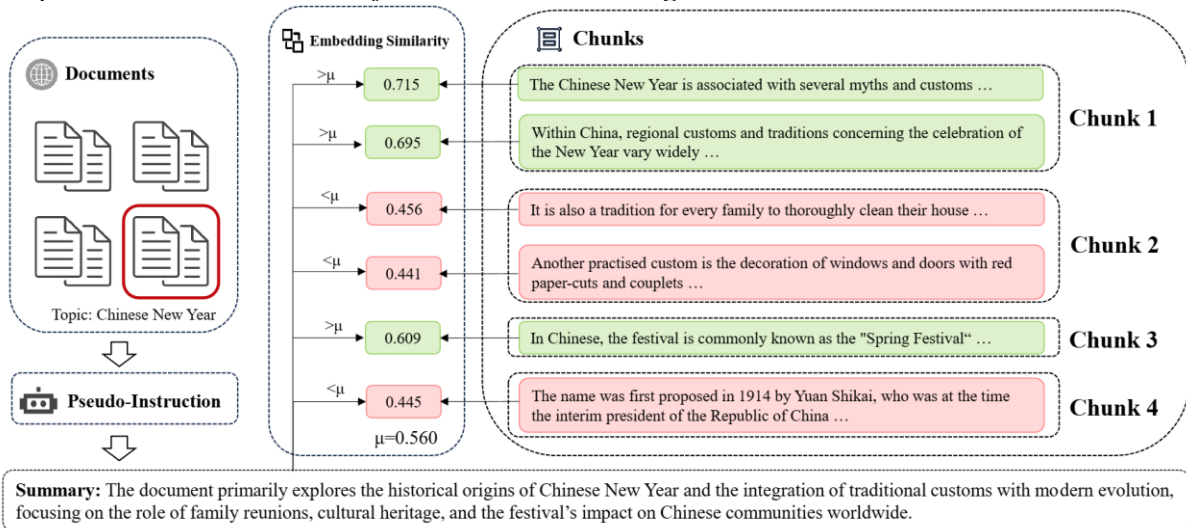


Source: Jiang et al. (2021)

While this method outperforms existing models on most performance metrics, it operates under the assumption that topics do not shift within a single paragraph. Adapting this architecture to a sentence-level analysis could potentially enhance the generation of self-contained topic snippets.

Recent research has explored using semantic similarity and LLMs to refine document chunking. Z. Wang et al. (2025) introduced the Pseudo-Instruction for document Chunking (PIC) framework, which generates a concise summary of the document to serve as a thematic anchor. As shown in Figure 19, sentences and the generated summary are embedded, specifically using the bge-large-en-v1.5 model, to calculate cosine similarity scores. Sentences are then grouped into chunks based on whether their similarity to the summary is above or below the document average. While this method aligns chunks with key themes, its performance gains over simpler fixed-size windows are minimal. Furthermore, the reliance on a static average threshold can lead to suboptimal splits when adjacent sentences fluctuate near the mean similarity.

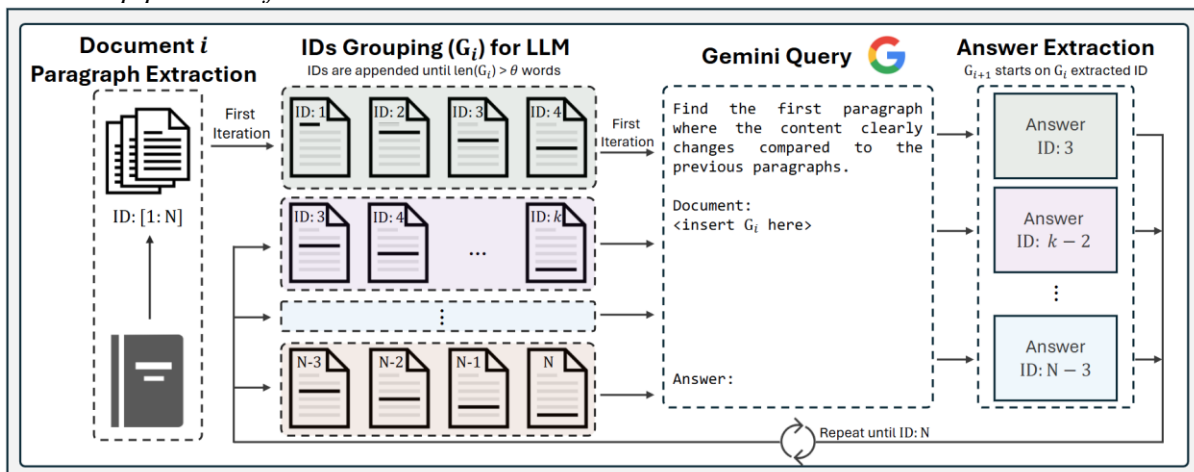
Figure 19
Proposed Pseudo-Instruction for document Chunking Framework



Source: Z. Wang et al. (2025)

An alternative by Duarte et al. (2024), purely LLM-based method known as LumberChunker utilizes iterative prompting to identify thematic shifts. In this process, illustrated in Figure 20, the document is divided into paragraphs that are sequentially grouped until they reach a specific token threshold, e.g., 550 tokens for the Gemini-1.0-pro model. The LLM is then prompted to identify the exact paragraph where the content clearly changes compared to previous entries.

Figure 20
Three-step process of LumberChunker



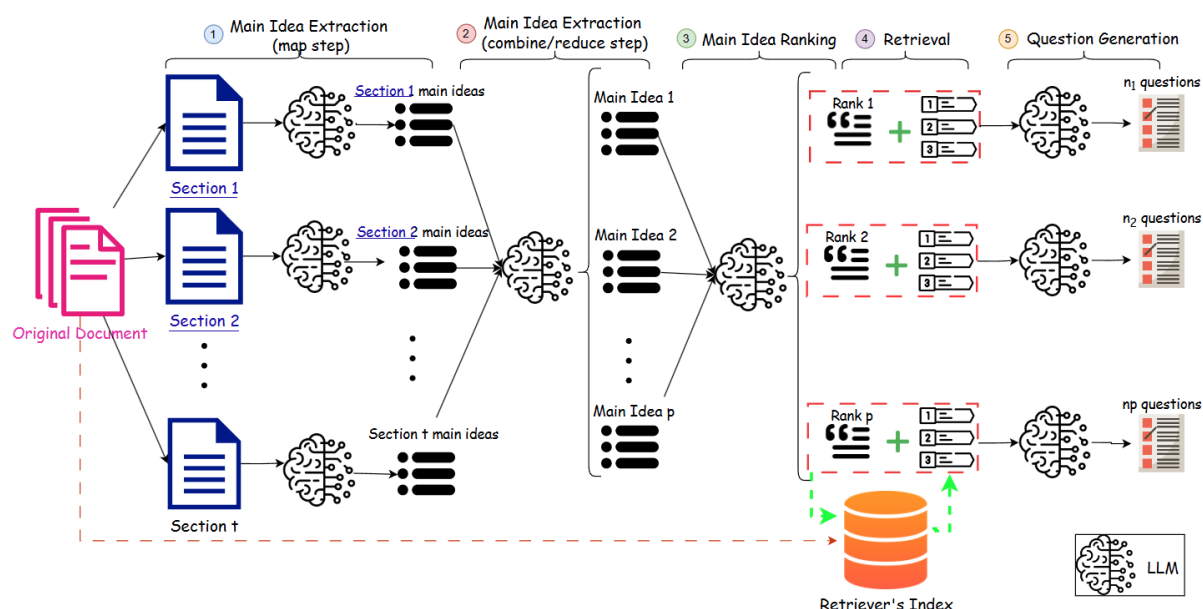
Source: Duarte et al. (2024)

This method avoids the reasoning errors associated with excessive context while ensuring segments are large enough to be meaningful. Although LumberChunker significantly outperforms traditional and embedding-based methods, the necessity for multiple, iterative LLM calls makes it considerably more expensive when processing large volumes of text. While originally tested on paragraphs, this logic might remain applicable for sentence-level segmentation to create highly specific topic snippets.

2.4.3 Semantic Modeling

Because academic documents often revisit core themes multiple times, linear segmentation can fail to capture the full context of a topic. Retrieval-Augmented Generation (RAG) address this by identifying and aggregating relevant segments from across an entire document. Noorbakhsh et al. (2025) introduced Savaal, a scalable, domain-independent pipeline designed to generate high-quality multiple-choice questions. As illustrated in Figure 21, the system uses an LLM to extract and rank key concepts, retrieves the three most relevant segments for each via the ColBERT embedding model, and then prompts an LLM to generate questions based on this targeted context. Randomizing the answer choices further mitigates potential LLM bias.

Figure 21
Savaal five-step Pipeline

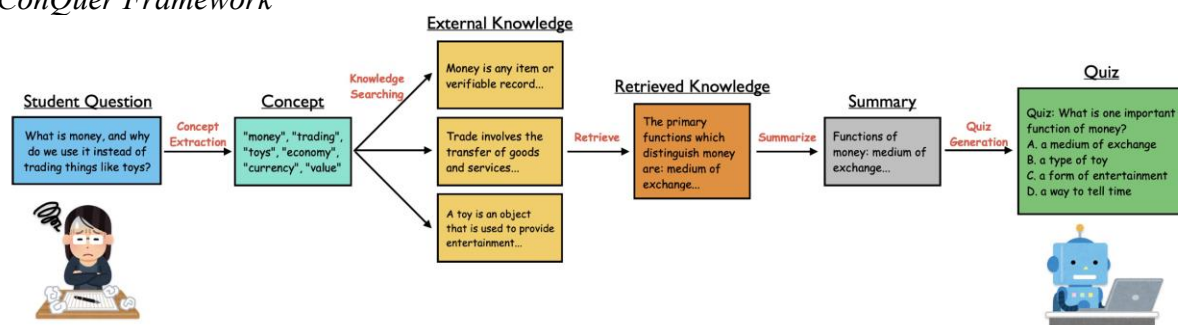


Source: Noorbakhsh et al. (2025)

This targeted approach is more effective than using a full document as context, as LLMs often allocate attention unevenly across long texts and struggle with dependencies between distant sections. By focusing on specific chunks, the pipeline avoids the "lost in the middle" phenomenon and the vagueness associated with simple corpus summarization. Expert evaluations have confirmed that Savaal outperforms other methods across varying document lengths. Notably, however, scores from LLM judges misaligned with expert assessments, suggesting that automated models are not yet reliable for evaluating question quality.

Fu et al. (2025) proposed the ConQuer framework, which follows a similar semantic logic but initiates the process through student inquiry. As shown in Figure 22, the system extracts concepts from a student's question, searches external knowledge bases like Wikipedia, and retrieves relevant chunks using Sentence-BERT cosine similarity. These chunks are then summarized by an LLM to produce a final quiz. While functionally robust, this method remains dependent on an initial user prompt to define the subject matter.

Figure 22
ConQuer Framework



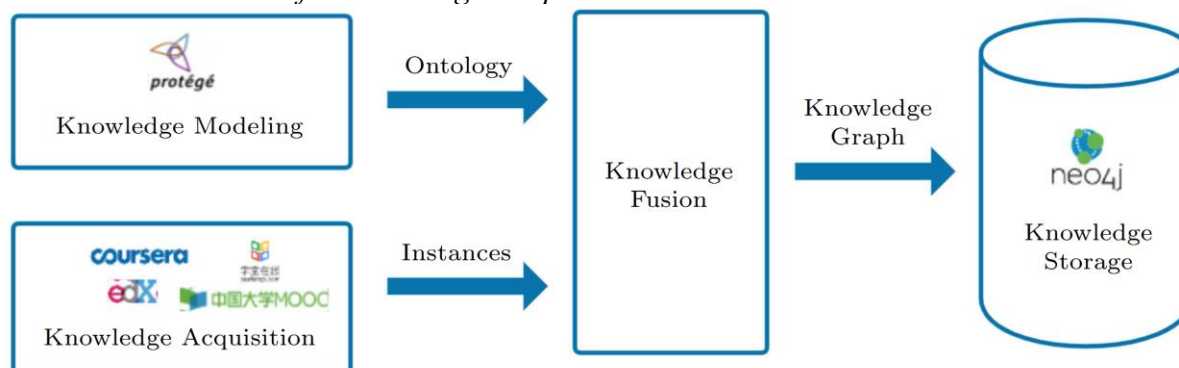
Source: Fu et al. (2025)

For more automated concept extraction, Nguyen et al. (2025) utilized the MOCCubeX corpus, containing courses, videos, exercises, concepts and behavioral data, to perform weakly supervised extraction without expert input. While powerful, such large-scale corpora may struggle with documents in different languages or emerging fields not yet represented in the training data.

2.4.4 Ontological Modeling

Ontological modeling via knowledge graphs offers a structured alternative for representing course content. Dang et al. (2021) proposed a framework for course recommendations consisting of a logical layer, which establishes governing axioms, and a data layer where information is organized into entity-relation-entity or entity-property-value triplets. The construction process, illustrated in Figure 23, involves conceptual knowledge modeling followed by knowledge acquisition from online platforms such as Coursera and edX. These datasets undergo knowledge fusion, where entity matching is determined by calculating attribute similarity using the Sorensen distance, before the final graph is stored in a database.

Figure 23
Construction Process of a Knowledge Graph



Source: Dang et al. (2021)

Beyond simple construction, algorithms like SciCheck have been developed to enhance these structures by predicting missing scientific statements within a graph (Borrego et al., 2022). As shown in Figure 24, this workflow takes existing triplets and generates negative examples to train neural-based classifiers for each specific relation. By converting these triplets into feature vectors, the system can evaluate the validity of new connections within the knowledge network.

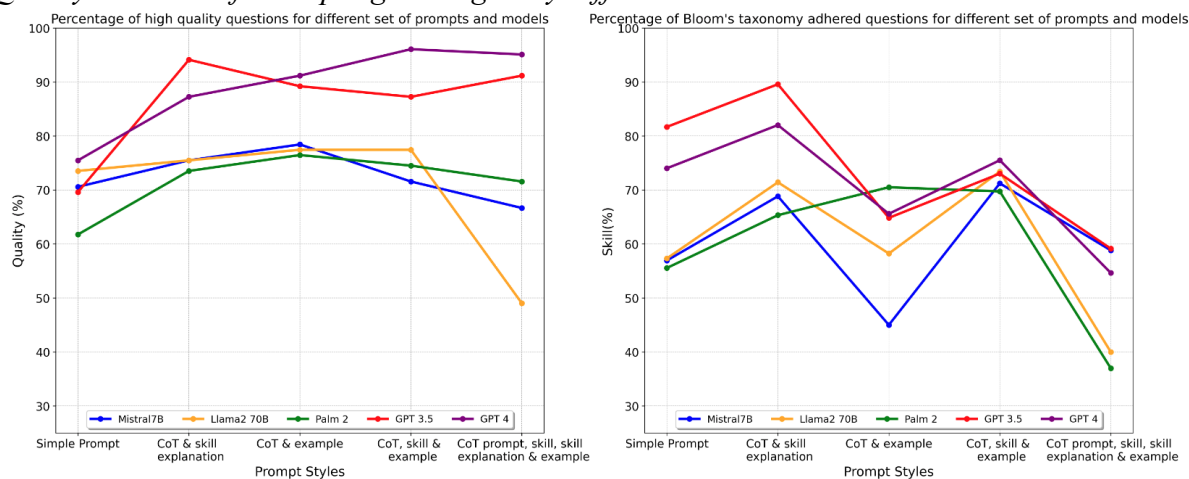
To address these shortcomings, alternative frameworks such as Webb's DOK are increasingly utilized, particularly in mathematical and technical subjects where task complexity and contextual knowledge are essential. Research suggests that Webb's framework may be better suited than Bloom's for these disciplines, though the resulting questions still require teacher guidance (Yu et al., 2025). While earlier conclusions regarding model inadequacy often targeted GPT-3.5, the emergence of state-of-the-art models like GPT-5, two generations ahead, offers a more robust foundation for generating useful, higher-order academic exercises.

2.5.2 Prompt Engineering Strategies

To enhance the educational utility of LLM outputs, several prompting strategies have been developed, including pattern reframing, chain-of-thought (CoT), and the use of specific instructor personas. Scaria et al. (2024) found that combining CoT with detailed definitions of Bloom's cognitive levels, though omitting specific examples, yielded the highest adherence to targeted taxonomic stages. Interestingly, as shown in Figure 25, the highest overall question quality was achieved when providing CoT without explicit taxonomic explanations. This suggests that the internal knowledge of advanced models may suffice when paired with logical reasoning paths.

Figure 25

Quality and Skill of Prompting Strategies by different LLMs

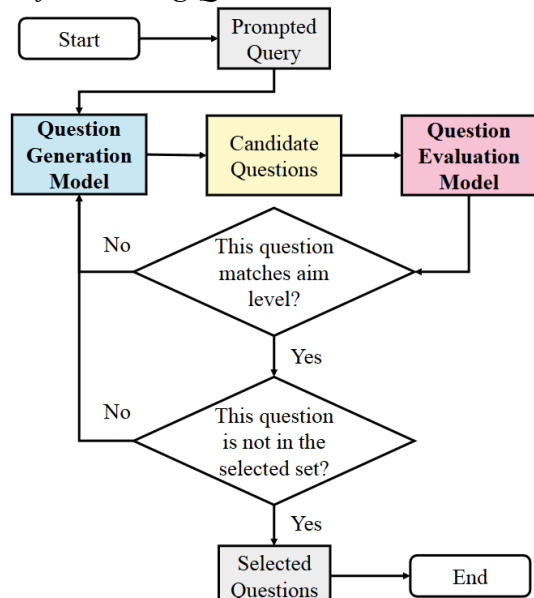


Source: Scaria et al. (2024)

Contextual enrichment further improves results. Kevin Hwang et al. (2023) observed that providing excerpts of all Bloom levels simultaneously, alongside section names and topic summaries, increased the quality of generated items. The use of few-shot learning, incorporating handcrafted examples into the prompt, is another common approach. While Elkins et al. (2024) utilized a 5-shot method, other research suggests that performance gains are inconsistent, with 8-shot learning significantly outperforming both 0-shot and intermediate 2-shot or 4-shot attempts.

For tasks requiring high precision, multi-step validation frameworks have proven effective. Zhuge et al. (2025) implemented a self-validating loop where an initial question is generated via 2-shot learning and then immediately evaluated by a secondary LLM for taxonomic adherence. As illustrated in the flowchart in Figure 26, if the question fails to match the targeted level or is found to be redundant, the system iterates until a valid item is produced. While this method achieves superior taxonomic alignment compared to larger proprietary models, it can sometimes lag behind in overall knowledge relevance.

Figure 26
Self-validating Question Generation Flowchart based on Bloom



Source: Zhuge et al. (2025)

2.5.3 Discourse Modeling, Pipeline Architectures and Model Configuration

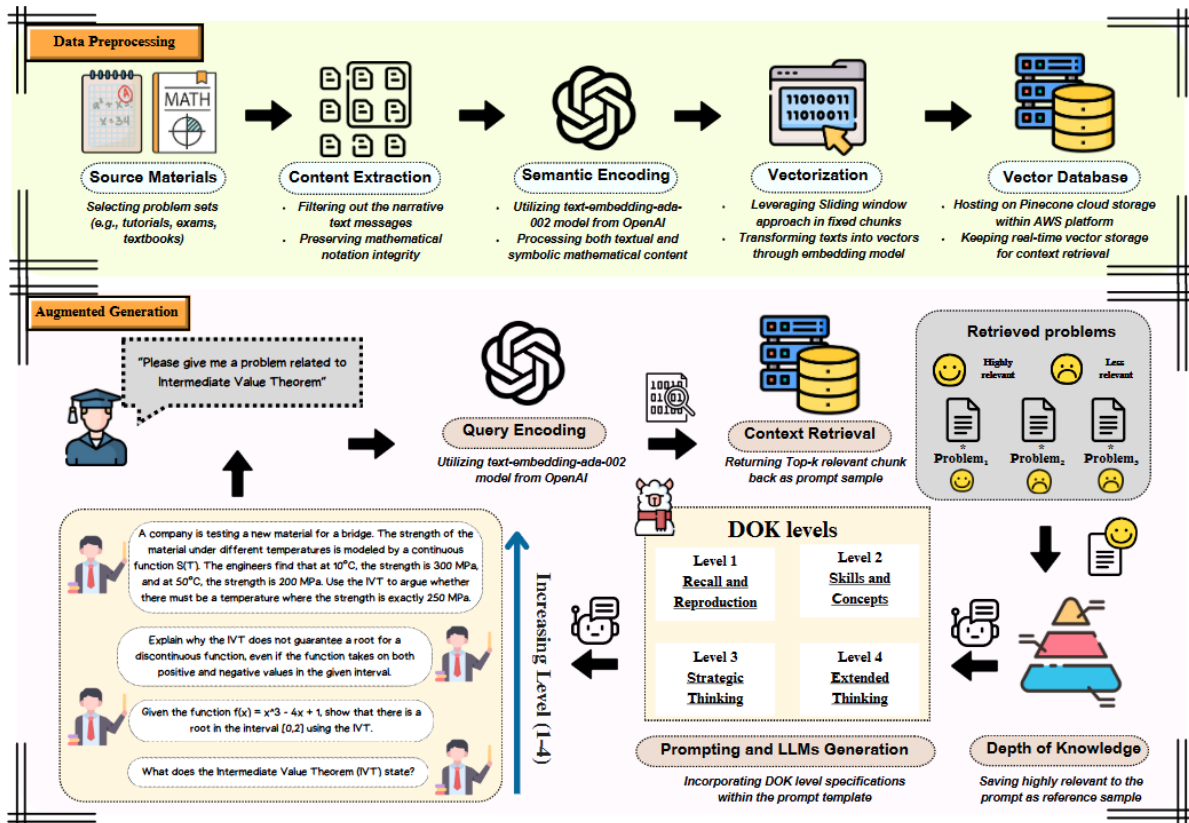
Beyond thematic extraction, natural language processing approaches can utilize discourse cues to automate question generation. Agarwal et al. (2011) developed a method that identifies discourse connectives, such as "because" or "since," and maps them to specific question types and target arguments. For instance, a sentence containing a causal connective is processed to formulate a "Why" question, with the system identifying which part of the sentence serves as the primary target for inquiry. Table 5 illustrates how various connectives and their linguistic senses are systematically mapped to different question formats.

Table 5
Question Type and Target Argument for Discourse Connectives

Discourse Connective	Sense	Question Type	Target Argument
Because	Causal	Why	Argument 1
Since	Temporal Causal	When Why	Argument 1
When	Causal + Temporal Temporal Conditional	When	Argument 1
Although	Contrast Concession	Yes/No	Argument 1
As a result	Result	Why	Argument 2
For example	Instantiation	Give an example where	Argument 1
For instance	Instantiation	Give an instance where	Argument 1

Source: Agarwal et al. (2011)

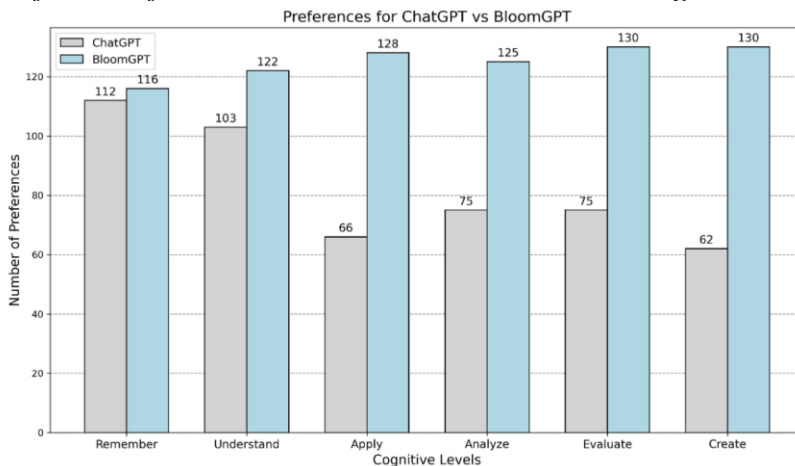
Figure 28
Overview of a RAG Framework using Webb’s DOK



Source: Yu et al. (2025)

Fine-tuning serves as another powerful method for ensuring didactic precision. Duong-Trung et al. (2024) introduced BloomLLM, a model specifically fine-tuned to generate questions mapped to Bloom’s Taxonomy. As shown in Figure 29, the fine-tuned version of ChatGPT-3.5-Turbo-1106 significantly outperformed the larger ChatGPT-4, particularly in maintaining accuracy at the most complex cognitive stages.

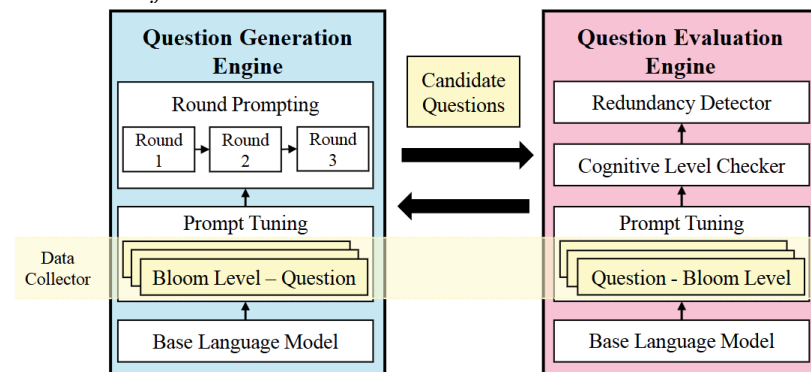
Figure 29
Preferences for ChatGPT-4 vs BloomLLM across Cognitive Levels



Source: Duong-Trung et al. (2024)

Complementing these generative models are specialized validation architectures. The TwinStar framework, illustrated in Figure 30, utilizes fine-tuning on benchmark datasets to increase taxonomic alignment without sacrificing subject-matter relevance.

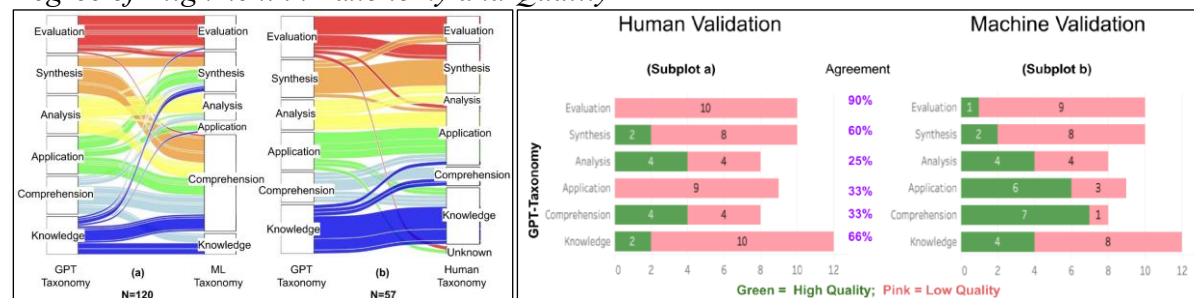
Figure 30
Overview of the TwinStar Architecture



Source: Zhuge et al. (2025)

A similar evaluative approach was employed by Kevin Hwang et al. (2023), who fine-tuned a RoBERTa model to detect 19 distinct "item-writing flaws" in generated MCQs. While the alignment between machine validation and human expertise appears high, as visualized in the taxonomy and quality distributions in Figure 31, further research using confusion matrices is required to confirm if these models identify specific cognitive levels with the same nuance as human judges.

Figure 31
Degree of Alignment in Taxonomy and Quality



Source: Kevin Hwang et al. (2023)

Finally, the success of these advanced architectures depends heavily on the quality of the input data. Elkins et al. (2024) emphasized the necessity of rigorous context cleaning, which includes removing citations, hyperlinks, and phonetic spellings, as well as reformatting bullet-pointed lists into standard paragraphs to preserve the natural flow of information for the model.

2.6 Assessment Design

2.6.1 Frameworks for Authentic Assessment

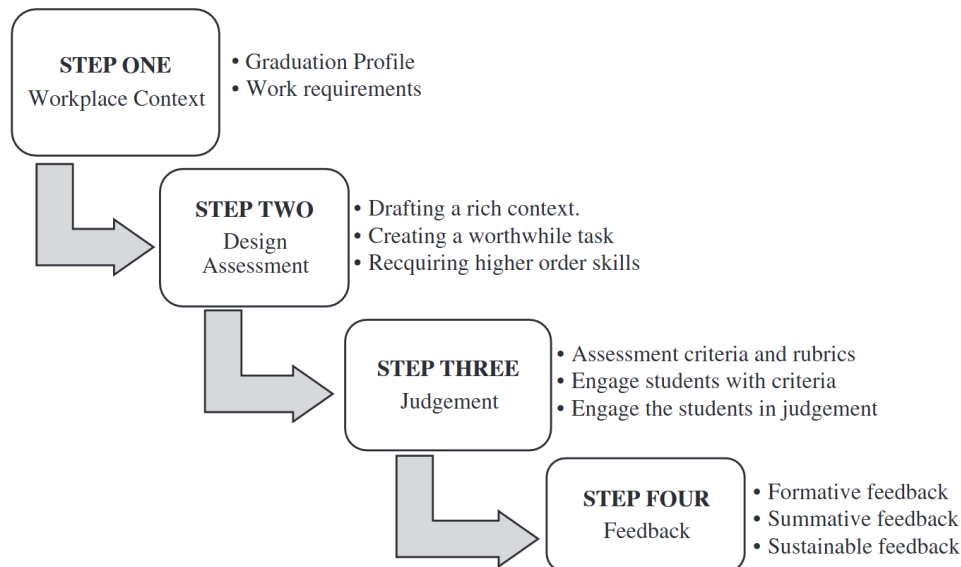
Authentic assessment serves as a critical bridge between academic theory and professional practice by integrating three core components: realism, contextualization, and problematization. Realism ensures tasks reflect everyday professional life, while contextualization requires students to apply knowledge analytically rather than through rote memorization. Problematization further encourages students to use their learning to solve

concrete problems or meet specific needs. This approach addresses common criticisms of higher education, where graduates often struggle to adapt to workplace demands or lack essential skills like critical thinking and teamwork.

To implement these principles, Villarroel et al. (2018) proposed a four-step model for building authentic assessments in a university setting, as illustrated in Figure 32.

Figure 32

Model to build Authentic Assessments



Source: Villarroel et al. (2018)

The first step focuses on the workplace context, requiring teachers to align course objectives with the graduation profile and employer demands. This involves identifying how specific subjects contribute to professional competencies and typical workplace problems. The second step, designing the assessment, emphasizes didactic decisions that simulate real-world challenges. A realistic assessment places students in contexts that force them to discriminate between relevant and irrelevant information to make informed decisions. This includes creating worthwhile tasks that provide value to third parties, such as clients or colleagues, and prioritizing higher-order cognitive skills like innovating, designing, and judging.

The final two steps focus on judgment and feedback. Students should be actively engaged with assessment criteria through self- and peer-assessment to develop evaluative judgment. Feedback is treated as a cyclical, dialogic process rather than a passive reception of grades. This includes formative feedback to motivate learners, summative feedback for quality assurance, and sustainable feedback to prepare students for independent lifelong learning. While these latter stages and general concerns regarding student wellbeing, such as assessment-related anxiety, are vital for a holistic learning cycle (Ismail et al., 2022; Jones et al., 2021), they are less central to the technical task of automated question generation.

2.6.2 Adapting Assessment Design for Generative AI

The emergence of Large Language Models (LLMs) has rendered many traditional assessment formats trivial, necessitating a shift in how educators design student evaluations. To address this, Moorhouse et al. (2023) suggest that instructors first benchmark existing assessments by testing them against generative AI tools. Redesigning these tasks requires a focus on creativity, critical thinking, and the integration of highly contextualized elements through authentic assessment design. Beyond text-based submissions, students should be

encouraged to represent knowledge through alternative modalities, with an emphasis on evaluating the learning process and various developmental stages rather than just the final product. Interestingly, the authors recommend integrating AI directly into the curriculum by requiring students to critique AI-generated responses, while shifting to in-class assessments can further mitigate unauthorized AI usage.

Complementary guidelines proposed by Eager & Brunton (2023) focus on utilizing generative AI to proactively create robust assessment tasks. This process begins by defining a specific goal for the AI, followed by determining the desired content type and format, such as a case study or quiz. A concise initial prompt is then crafted and tested, followed by a reflective iteration of the outputs. Effective prompts in this context should include several key components: an action verb, a specific focus, whether product, process, or outcome, and a clear definition of the subject and primary goal. Furthermore, the prompt must establish the task's context and align strictly with intended goals while accounting for inherent constraints and limitations.

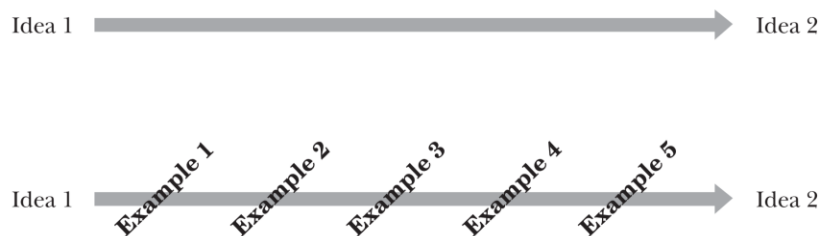
2.7 Didactics with Exercises

Effective educational materials must balance abstract concepts with concrete information to maintain learner interest and facilitate recall. While abstract text is often difficult to visualize, concrete language and imagery allow students to conjure mental representations more easily. For example, describing "tribal marriage customs" is more memorable than referring generally to "traditional customs". Readability can be further improved by utilizing shorter words, active sentence structures, and illustrations such as graphs or photographs that direct attention to key tasks. Additionally, a personalized instructional style using direct address can aid comprehension, though excessive contextualization should be avoided as it may hinder a student's ability to generalize principles beyond a specific example (Morrison et al., 2019).

The structural delivery of content, or pacing, is equally critical. As illustrated in Figure 33, educators must control the step size between new ideas and provide explicit references to prior learning to ensure consistency. While naïve learners require a measured pace with consistent terminology to avoid confusion, more advanced students may benefit from a faster pace to maintain engagement during reviews.

Figure 33

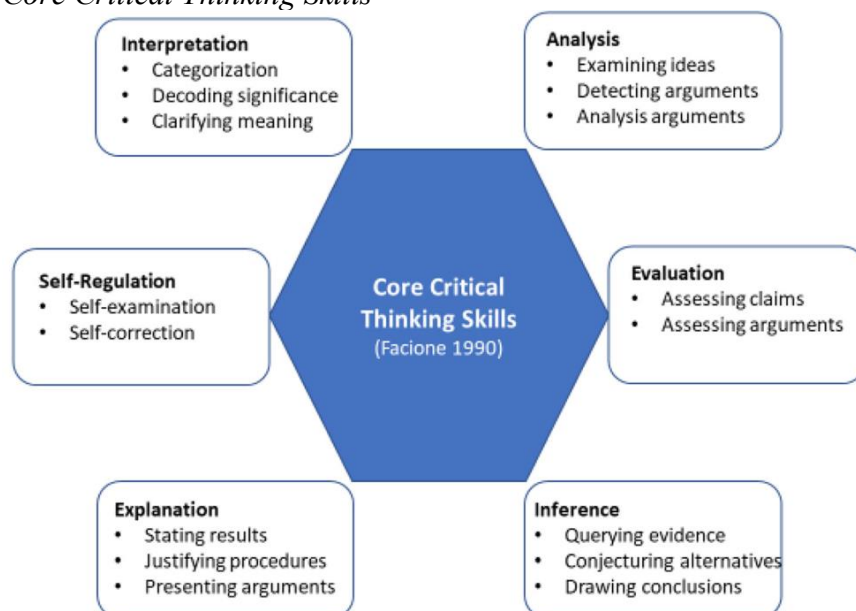
Pacing for Instructions



Source: Morrison et al. (2019)

Beyond content delivery, instructional design aims to foster critical thinking, which is defined as an individualistic cognitive act comprising analysis, evaluation, inference, explanation, self-regulation, and interpretation. This skill set, visualized in Figure 34, aligns with the higher-order levels of Bloom's Taxonomy and serves as the ultimate goal for advanced cognitive development (Ostendorf & Thoma, 2022).

Figure 34
Core Critical Thinking Skills



Source: Ostendorf & Thoma (2022)

Engagement can be further enhanced through gamification, where elements such as progress bars and competitive scoring motivate students to sustain activity over longer periods (Jodoi et al., 2021). While pacing and gamification are broader implementation details, the use of concrete information and the incentivization of critical thinking through cognitive frameworks are principles that can be directly integrated into the prompt engineering of an LLM-based generation pipeline.

2.8 Knowledge Gap and Research Objective

2.8.1 Knowledge Gaps

Research into automated question generation reveals several critical knowledge gaps regarding cognitive frameworks, document processing, and the application of LLMs. While Bloom's Taxonomy and Webb's DOK are widely utilized, other frameworks like the SOLO Taxonomy or the five-stage model of skill acquisition remain unexplored. Furthermore, there is a lack of comparative research to determine which framework most effectively facilitates the creation of high-quality questions. Current tools primarily produce MCQ, which frequently lack the depth required to stimulate higher-order thinking (Ch & Saha, 2020, 2023). Pure NLP methods exacerbate this by focusing on sentence-level analysis, resulting in context-poor outputs (Killawala et al., 2018). While direct LLM prompting can improve quality, it often fails to assess a student's actual depth of understanding and places heavy manual burden on instructors to curate content and instructions (Lee et al., 2024).

A technical pipeline for automated question generation faces significant hurdles in context preparation and structural integrity. Effectively filtering "unfit" content, such as table of contents or teacher introductions, is essential for resource efficiency but remains under-researched. Maintaining the original document structure, including headings, footnotes, and figures, is particularly challenging for PDF files compared to the more consistent HTML structures used by Nguyen et al. (2022). To mitigate LLM hallucinations and attention limitations, researchers have attempted to isolate single ideas through various modeling techniques. However, rule-based approaches often prove inconsistent (Marti A. Hearst, 1997), and while machine learning models offer better splits, they frequently suffer from poor

generalization or high computational costs (Duarte et al., 2024; Jiang et al., 2021; Koshorek et al., 2018; Z. Wang et al., 2025).

Efficiency and depth remain at odds in current literature. The Savaal framework (Noorbakhsh et al., 2025) improves efficiency by retrieving only concept-relevant chunks for MCQ generation, yet it risks missing less prominent concepts and lacks a mechanism for generating deep, open-ended questions. Similarly, while knowledge graphs excel at visualizing course structures, they often lack the necessary context for each entity to support higher-order cognitive processes (Demaidi et al., 2017). Most studies combining Bloom's Taxonomy with GPT-3.5 conclude that generated questions still require manual teacher adjustments (Duong-Trung et al., 2024; Maity et al., 2025; Zhuge et al., 2025). It remains to be seen if newer LLM generations can bridge this gap and provide truly autonomous utility.

Finally, there is a distinct lack of concrete guidance regarding prompt engineering and didactic integration. While chain-of-thought (Scaria et al., 2024) and few-shot learning (Elkins et al., 2024; Zhuge et al., 2025) are recommended, specific instructions on prompt length and structure are absent. Validation methods, such as multi-round prompting with fine-tuned or smaller models, show promise but are often resource-intensive (Duong-Trung et al., 2024; Kevin Hwang et al., 2023). Integrating didactic principles like authentic assessment (Villarroel et al., 2018) or clear writing standards (Morrison et al., 2019) into system prompts could enhance quality, yet a comprehensive guideline of these principles in automated question generation has yet to be established. Table 6 provides a summary of these identified gaps.

Table 6
Knowledge GAP-Table of Literature Review

Researcher	Focus	Insights	Knowledge Gap
Ch & Saha (2020)	Systematic review on MCQ generation and outline of an generic workflow.	A generic workflow for MCQ generation consists of 6 dependent phases.	MCQ do not promote higher order cognitive processes.
Ch & Saha (2023)	Pipeline proposal for MCQ generation using NLP and Deep Learning.	95% of generated MCQ are worthy for middle-school.	
Killawala et al. (2018)	Framework for question generation using mostly LSTM to classify the question type and only NLP for transforming the sentence into a question.	True/False, MCQ, Fill-in-the-blank, and Wh-questions can be generated.	Only short questions are generated, because of a lack of context. Short questions also do not promote higher order cognitive processes.
Lee et al. (2024)	Question generation protocol using ChatGPT in English education.	Different questions can be generated by defining a task type and format.	The protocol requires human involvement for each question and solving the question correctly or incorrectly, does not give insights about how good the student understands the topic.
Nguyen et al. (2022)	Pipeline for generating and evaluating questions from text-based learning materials.	Summary data and another LLM for validation increases question quality.	The pipeline uses text from a structured format (XML/HTLM) and not unstructured files like PDF.
Marti A. Hearst (1997)	Technique for subdividing texts into multi-paragraph units that represent passages or subtopics.	The technique showed “acceptable” performance using term repetition alone.	The technique is inconsistent because of limited assumptions compared to machine learning algorithms.
Koshorek et al. (2018)	Presented a text segmentation dataset based on Wikipedia and model that was trained using supervised learning.	Outperforms other methods on Wikipedia documents and other benchmarks.	Unknown generalization capabilities to new topics and languages and training or fine-tuning requires a lot of data and computational resources.
Jiang et al. (2021)	Constructing micro discourse structure trees using implicit topic boundaries to help constructing the document-level macro discourse tree.	Outperforms other methods on most metrics.	
Duarte et al. (2024)	Method to dynamically segment documents using an LLM to identify where the content begins to shift.	Achieved SOTA retrieval performance.	This method has high costs due to many API calls.
Noorbakhsh et al. (2025)	Framework to extract concepts, retrieve relevant passages, and generate MCQ for most relevant concepts.	Framework improved testing understanding, choice quality, and usability.	Framework was not used for open-ended questions and might not cover all of the content.
Kevin Hwang et al. (2023)	Using Bloom’s Taxonomy to generate questions and fine-tuning a small language model to evaluate those questions.	Complexity alignment between GPT-generated questions and human-assessed complexity, with occasional disparities.	No clear quality alignment, meaning the small language model did not help with evaluation.
Elkins et al. (2024)	Document processing and question generation using an LLM and Bloom’s Taxonomy.	5-shot learning and generating questions for all 6 levels at once improved question quality and variety.	The studies use older generation of LLMs, do not give recommendations or insights about the ideal structure of the prompt, and validation is often accompanied by costs from additional API calls or fine-tuning, which requires extensive data and computational resources.
Maity et al. (2025)	Generate questions based on school-level textbooks with and without Bloom’s Taxonomy.	8-shot learning showed clear improvements over 0-, 2-, and 4-shot learning	
Zhuge et al. (2025)	Proposing a design scheme based on Bloom’s Taxonomy using 2 LLMs, where one generates the question and the other evaluates it.	The generated questions improve alignment with the cognitive levels while maintaining the quality.	
Scaria et al. (2024)	Examination of different prompting techniques to generate questions based on Bloom’s Taxonomy	Chain-of-thought with descriptions of the cognitive levels led to the best alignment to the taxonomy and adding examples increased quality of the questions.	
Duong-Trung et al. (2024)	Fine-tuning an LLM to generate questions with better alignment with Bloom’s Taxonomy.	Fine-tuning improved preference especially at higher cognitive levels.	
Villarroel et al. (2018)	Proposal of a model to integrate authentic assessments in universities.	A step-based model consisting of considering the workplace context, designing authentic assessments, learning and applying standards for judgment, and giving feedback.	The studies do not cover questions generated with LLMs.
Morrison et al. (2019)	Providing instructions for designing effective instructions.	Concrete information, active learning, and illustrations improve clarity about the authors intent.	

2.8.2 Research Objective

The literature review highlights that "optimal" question generation is not a static target but a function of content structure, pedagogical framing, and cost-efficiency. Previous research reveals critical friction points: an over-reliance on MCQs which fail to test deep understanding, a struggle to maintain structure in authentic PDF documents, and a lack of empirical comparison between cognitive frameworks like Webb's Depth of Knowledge and Bloom's Taxonomy in an AI context.

Consequently, the research objective defined in Chapter 1.4 is operationalized here to specifically address these gaps. The goal is to move beyond simple text generation to a comparative analysis of pipeline architectures. This involves rigorously testing how different combinations of context preparation (e.g., Sliding Window vs. Concept Extraction) and didactic steering (e.g., Bloom vs. Webb) influence the quality of the output.

The refined research objective focuses on three main points:

- **Breadth and Depth:** Ensuring that the preprocessing stage does not just pick the "easiest" parts of a document but covers every notable topic to provide a complete overview of the course material at different difficulty levels.
- **Didactic Alignment:** Comparing how different cognitive frameworks affect the quality of the generated exercises.
- **Feasibility and Value:** Evaluating these pipelines not just on their technical output, but on their real-world utility and their cost-effectiveness using modern LLMs.

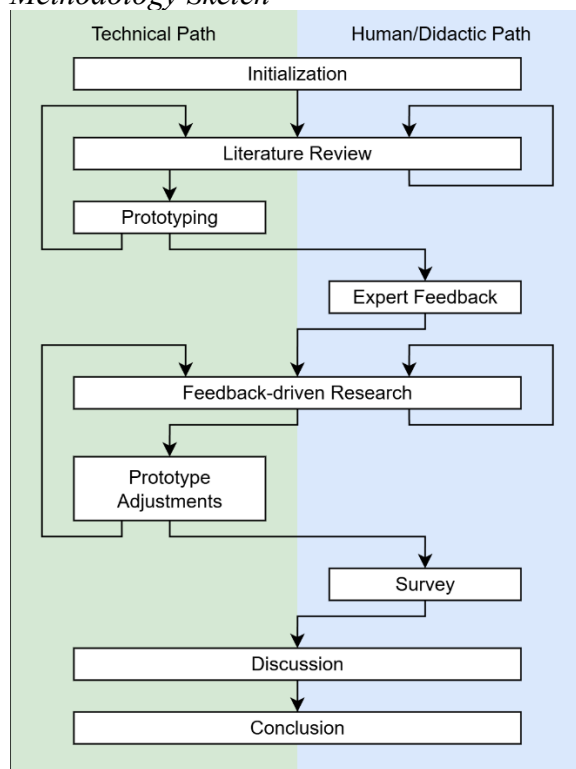
By addressing these dimensions, this study aims to fulfill its primary aim:

"What is the optimal LLM-based pipeline architecture for transforming unstructured university course materials into didactically sound, open-ended assessment pairs by empirically comparing context preparation methods and cognitive frameworks regarding their technical quality, cost-effectiveness, and expert-perceived utility."

3. Methodology

Following the identification of critical gaps in automated assessment in Chapter 2, this chapter details the design and implementation of four technical pipelines. The objective is to move beyond simple question generation toward a system capable of producing didactically sound, open-ended exercises from unstructured university materials. To achieve this, the study adopts a mixed-methods approach that evaluates two primary variables: context preparation and cognitive frameworks. As illustrated in the methodology sketch below (Figure 35), the research design integrates a Technical Path with a Human/Didactic Path to ensure both functional performance and educational quality.

Figure 35
Methodology Sketch



The mixed-methods framework allows for a multi-layered validation of the proposed system. The quantitative component focuses on technical performance metrics, specifically measuring processing time and financial costs based on token usage. This path also includes a win-rate analysis derived from expert survey rankings to statistically identify the most preferred pipeline configurations. Complementing this, the qualitative component involves the systematic collection of feedback from 21 module coordinators. By utilizing Gioia's Data Structure, this qualitative data is synthesized to identify overarching themes regarding the AI's didactic value, linguistic clarity, and practical utility in a university setting. The following sections describe the selection criteria for these components, the technical architecture of the prototypes, and the specific validation method used to measure their value for university lecturers.

3.1 Method Selection

The methodology for constructing the question generation pipeline centers on three primary variables: context preparation, the selection of a cognitive framework, and the

question generation technique. These variables and their respective options from the literature are organized into the morphological box presented in Table 7. To refine the scope of this study, several options were eliminated based on specific exclusion criteria.

Resource constraints led to the removal of methods requiring custom model training, such as LSTM or LLM fine-tuning, due to their high computational and time demands. Regarding cognitive depth, any methods restricted to lower-level cognitive processes were excluded to ensure the pipeline supports complex thinking tasks. To preserve contextual integrity, approaches using hard semantic cutoffs were rejected, as maintaining narrative flow is essential for question quality. Furthermore, the selection of cognitive frameworks was guided by framework intuition; goal-oriented models like Bloom’s Original Taxonomy, the Dreyfus five-stage model, and SOLO Taxonomy were set aside in favor of process-oriented frameworks like Bloom’s Revised Taxonomy and Webb’s Depth of Knowledge. These were chosen because their action-oriented verbs translate more effectively into LLM instructions. Finally, any methods limited to short-form text were excluded to ensure the system remains functional for documents of any length. The resulting pipelines are designed to be LLM-agnostic, allowing didactic strategies to be integrated into the system prompt without contradiction.

Table 7
Morphological Box for Pipeline Prototypes

Context Preparation	Cognitive Framework	Question Generation
Fixed Sliding Window	Bloom’s Taxonomy (original)	Discourse Cues (Agarwal et al., 2011)
Recursive Chunking	Bloom’s Taxonomy (revised)	Generate single question at once using Bloom (Duarte et al., 2024)
Concept Extraction (Fu et al., 2025; Nguyen et al., 2022; Noorbakhsh et al., 2025)	SOLO Taxonomy	Generate all questions at once using Bloom (Elkins et al., 2024; Maity et al., 2025; Scaria et al., 2024)
TextTiling (Marti A. Hearst, 1997)	Webb’s Depth of Knowledge	Retrieval-Augmented Generation with Webb’s DOK (Yu et al., 2025)
LSTM (Koshorek et al., 2018)	Dreyfus’ 5-Stage Model	Fine-tune LLM with Bloom data (Duong-Trung et al., 2024)
Topic Segmentation with hierarchical tree (Jiang et al., 2021)		Fine-tune two LLMs (generation + evaluation) (Zhuge et al., 2025)
Document Summaries as Pseudo Instructions (Z. Wang et al., 2025)		LSTM + Attention (Tuan et al., 2019)
LLM-based chunking (Duarte et al., 2024)		

3.1.1 Context Preparation

Four methods for context preparation were evaluated based on their technical feasibility and didactic value. Recursive Chunking serves as a baseline approach, dividing documents using predefined separators such as Markdown headers and paragraphs. By descending through a hierarchy of separators until chunks reach a target size, this method remains computationally inexpensive and maintains semantic continuity. However, it may result in chunks that lack necessary context from preceding sections. To address this, Concept Extraction combined with RAG offers a more sophisticated alternative. This technique extracts key concepts from a pre-chunked document and uses specialized embedding models to identify the most relevant sections via similarity functions like cosine similarity. While this adds a layer of complexity and increases computational costs, it ensures the LLM receives highly focused information for question generation.

Other potential methods were excluded due to unfavorable cost-benefit ratios. Generating a hierarchical tree via topic segmentation, as proposed by Jiang et al. (2021), operates similarly to recursive chunking but relies on BERT models for paragraph-level analysis. The marginal improvement in segmentation quality does not justify the significant increase in complexity and processing requirements. Similarly, while LLM-based topic

detection at the sentence level is highly accurate, the high volume of API calls required for long documents makes it prohibitively expensive and slow for large-scale applications. Consequently, both hierarchical trees and LLM-driven chunking were omitted from the final method selection.

3.1.2 Cognitive Frameworks

Two primary cognitive frameworks were evaluated for their integration into the question generation pipeline: Bloom's Revised Taxonomy and Webb's DOK. Bloom's Revised Taxonomy identifies six distinct cognitive processes (Remember, Understand, Apply, Analyze, Evaluate, and Create) alongside various knowledge types. This framework is particularly advantageous for LLM integration because it utilizes specific, action-oriented verbs that serve as precise instructions for the model, likely enhancing the consistency of the generated questions. Furthermore, its widespread adoption in educational settings ensures that the output remains intuitive and familiar to instructors.

In contrast, Webb's DOK categorizes task complexity into four levels based on the cognitive effort required to complete a task. While robust, DOK presents challenges for a generalized pipeline because the boundaries between levels can be subjective. Webb suggests that level definitions should be customized for specific use cases, which conflicts with the goal of creating a system that generalizes across diverse course materials.

3.1.3 Question Generation

Implementing an automated question generation strategy requires careful consideration of how Large Language Models (LLMs) process complex instructions. While one might assume that providing descriptions for all cognitive levels at once would overwhelm a model's attention, recent literature suggests the opposite. Generating all questions in a single pass often results in greater diversity and better adherence to specific levels (Elkins et al., 2024; Maity et al., 2025; Scaria et al., 2024). This is likely because state-of-the-art models, such as Gemini 3 Pro (2025), are trained on massive context windows and designed to be robust against "instruction drift."

To further refine this process, a "planning step" can be integrated into the prompt. By instructing the model to first outline how each cognitive level will represent the provided context without redundancy, the subsequent questions become more structured and coherent. This approach leverages established "Chain-of-Thought" techniques (Sun et al., 2024; Wei et al., 2022) and the advanced reasoning capabilities of current frontier models.

Regarding the use of few-shot learning, the ideal number of examples remains a point of debate in the research. Findings range from eight examples being necessary for a performance boost (Maity et al., 2025) to five examples providing significant improvements over zero-shot prompting (Elkins et al., 2024). Based on these trends, this methodology will utilize five examples to balance performance with token efficiency, acknowledging that the optimal number may vary by specific use case.

3.1.4 Didactic Strategies

Integrating didactic principles into the system prompt ensures that the generated questions are not only technically accurate but also pedagogically sound. While some strategies, such as sourcing "authentic" or "up-to-date" external materials (John & Devi, 2021), are not applicable since the pipeline relies on instructor-provided documents, several other methods can be effectively embedded into the LLM's instructions. For instance, prompting the LLM to adopt a specific persona, such as an experienced higher education instructor, helps establish the appropriate tone and academic rigor.

Furthermore, the pipeline can incorporate the framework for authentic assessment proposed by Villarroel et al. (2018) and Moorhouse et al. (2023). This involves three key pillars: Realism, which links tasks to everyday life or professional work; Contextualization, which requires students to apply knowledge analytically within a specific scenario; and Problematization, which frames the question as a meaningful challenge to be solved. By requiring these elements in the prompt, the generated questions move beyond rote memorization toward practical application.

Finally, linguistic clarity is essential for student comprehension and knowledge transfer. Following the recommendations of Morrison et al. (2019), the system prompt should instruct the LLM to balance abstract concepts with concrete details, use active voice, and prefer shorter, high-frequency words. Subtle personalization, such as the use of "you" or "your," can also be included to improve learner engagement. These stylistic constraints, combined with a focus on concrete imagery, ensure the questions are both readable and memorable.

3.2 Prototype Implementation

The prototype implementation centers on four distinct pipeline combinations derived from the two primary variables: context preparation and cognitive framework. These combinations, detailed in Table 8, were developed within an isolated Anaconda environment to ensure consistency and reproducibility.

Table 8

Pipeline Combinations for Implementation

	Bloom's Revised Taxonomy	Webb's Depth of Knowledge
Recursive Chunking + Sliding Window	Pipeline 1	Pipeline 2
Concept Extraction + Retrieval-Augmented Generation	Pipeline 3	Pipeline 4

All pipelines share an initial processing phase. This involves extracting text from documents into Markdown format, generating an executive summary via LLM, and applying recursive chunking. From this baseline, the implementation diverges into two distinct architectural approaches:

Sliding Window Pipelines

These pipelines utilize the generated summary and individual chunks to produce questions based on the selected framework (Bloom or Webb). To maintain semantic flow, the context window is expanded by including the three preceding and three succeeding chunks for every target chunk.

Concept Extraction Pipelines

These versions use the summary to extract up to three key concepts per chunk. An embedding model then calculates similarity measures to retrieve the three most relevant chunks for each concept. The LLM generates questions based on these specific concepts, supported by the retrieved contextual chunks.

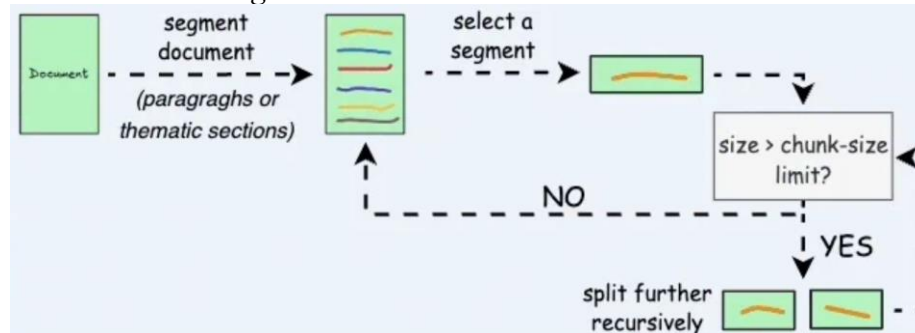
The system is built using a modular design to allow for flexibility and testing. The end-to-end process, from raw document input to final question output, is visualized in the implementation flowchart in Figure 36.

H1 → H2 → H3 → H4 → H5 → H6 → Paragraphs → Newlines → Spaces → Characters

The splitter targets a maximum chunk size of 1500 characters with a 100-character overlap to preserve continuity between segments. If a section exceeds the 1500-character limit, the algorithm recursively splits it using the next separator in the hierarchy until the size constraint is met (Figure 37). To prevent the loss of structural context during this fragmentation, all relevant parent headers are prepended to each individual chunk.

Figure 37

Recursive Chunking Flowchart



Source: Avi Chawla (2024)

For pipelines utilizing concept extraction, the Qwen3-Embedding-0.6B model is employed to calculate vector representations. This model ranks 8th on the MTEB leaderboard, which supports over 1038 languages (*Embedding Leaderboard*, n.d.). It offers high performance while remaining small enough to process batches locally on a GPU or CPU. With an embedding dimension of 1024, it enables faster retrieval than larger models, and its 32,768-token context window ensures that 1500-character chunks are never truncated. The system uses cosine similarity to retrieve the three most relevant chunks for any extracted concept, providing the LLM with a highly focused and semantically aligned context for question generation.

3.2.3 Instructions

The implementation of the pipeline relies on four specialized instruction sets, summary generation, concept extraction, Bloom-based question generation, and Webb-based question generation, which are detailed in Appendix 9.4. Each instruction adopts a professional persona, such as an experienced higher-education instructor, and specifies strict output formats like JSON or Python lists. To ensure high-quality outputs, the prompts include specific guidelines for didactic strategies and quality checks. These checks prevent the model from referencing specific document "chunks" that would be invisible to students and ensure the use of concrete rather than abstract language. Generalizability is reinforced through few-shot examples covering diverse fields like Introductory Statistics, Cell Biology, and Policy and Ethics. The system is also trained to handle non-testable content by returning standardized "no content found" indicators when appropriate.

3.2.4 LLM

The modular architecture allows for the use of any Large Language Model, though OpenAI's GPT-5-mini was selected for this project due to its cost-efficiency. As illustrated in Figure 38, this model significantly reduces expenses compared to the standard gpt-5 while maintaining comparable performance. To further optimize costs, the system utilizes a cache

key for repeated prefix tokens, reducing input costs by a factor of ten. Processing parameters are adjusted based on task complexity, with reasoning and verbosity set to "low" for extraction tasks and "high" for question generation. To maintain system stability, a robust extraction logic identifies JSON or list brackets within the model's response, ensuring that any conversational "chatter" from the LLM does not disrupt the data pipeline.

Figure 38

GPT-5 and GPT-5-mini Pricing

GPT-5	GPT-5 mini
The best model for coding and agentic tasks across industries	A faster, cheaper version of GPT-5 for well-defined tasks
Price	Price
Input: \$1.250 / 1M tokens	Input: \$0.250 / 1M tokens
Cached input: \$0.125 / 1M tokens	Cached input: \$0.025 / 1M tokens
Output: \$10.000 / 1M tokens	Output: \$2.000 / 1M tokens

Source: Pricing (n.d.)

3.2.5 Logging

Transparency and performance tracking are handled through a comprehensive logging system. Every stage of the process, including the full Markdown conversion, individual chunks, extracted concepts, and final question sets, is exported into a JSON log. These logs provide granular data on processing time and token usage, subdivided into input, cached input, reasoning, and output categories. This level of detail is particularly useful for analyzing the financial impact of input caching and reasoning tokens.

3.2.6 Frontend User Experience

The user experience is designed to be straightforward. Users provide a source document and select their preferred configuration: a context preparation method (Sliding Window or Concept Extraction) and a cognitive framework (Bloom or Webb). Additionally, users can define a specific page range to bypass non-testable sections like appendices. Once processing is complete, the system generates a structured JSON file that can be easily integrated into e-learning platforms or used for manual review and distribution.

3.3 Survey

To evaluate the effectiveness of the four question-generation pipelines, a survey was conducted among module coordinators at the Lucerne University of Applied Sciences and Arts – Engineering and Architecture. Out of 60 contacted coordinators, 21 participated from seven different departments, with three individuals contributing data for two separate modules. Table 9 details the participation distribution across these departments.

Table 9*Contacted Module Coordinators*

Department	Emails Send	Surveys Filled
Electrical Engineering IET	22	3
Innovation and Technology Management IIT	18	7
Medical Engineering IMT	2	2
Building Technology and Energy IGE	2	2
Natural Sciences and Humanities ING	10	3
Digital Engineering DE	6	3
Interior Architecture IIA	1	1
Total	60	20

3.3.1 Data Collection

Data collection was structured to respect the significant time required for expert evaluation. Each coordinator reviewed 20 questions, distributed across the four pipelines and divided into four difficulty levels based on Hess' Cognitive Rigor Matrix. Difficulty 1 corresponded to Webb's DOK Level 1 and Bloom's Remember; Difficulty 2 covered Understand and Apply; Difficulty 3 addressed Analyze and Evaluate; and Difficulty 4 targeted Create. To mitigate selection bias, a randomized variant system was used, ensuring that the pipeline associated with a specific label (e.g., Variant A) changed at each difficulty level. Participants ranked these variants from 1 (best suited for their module) to 4 (least suited), allowing for ties only when necessary. The survey concluded with qualitative queries regarding the relevance of extracted concepts, willingness to use the questions with and without automated evaluation that would perform at their standard, and general feedback.

3.3.2 Evaluation

For a more nuanced analysis, the source documents were categorized by type, specifically text-heavy Scripts, content-rich Text-Slides, and Image-Slides, and by subject matter, including Mathematics, Engineering/Computer Science, and General studies. Language distinctions between German and English were also noted to provide granular recommendations. The evaluation of these pipelines utilizes the win-rate method, a pairwise comparison system that effectively manages ties and missing values by awarding 0.5 wins to both pipelines in the event of a draw. Quantitative metrics such as cost and processing time are analyzed using Ordinary Least Squares (OLS) linear regression to identify key performance drivers. Finally, qualitative feedback is systematically synthesized using Gioia's Data Structure to identify overarching themes and insights.

4. Results

This chapter presents empirical findings from the prototype implementations and the subsequent expert evaluation. Having detailed the technical architecture in Chapter 3, the following sections analyze the performance of the four pipelines through two aspects: operational efficiency (time and cost) and didactic quality (expert survey data).

The objective is to provide a data-driven answer to the research question by identifying the optimal configuration for university-level exercise generation. The results include statistical models for performance prediction, quantitative win-rates for the different architectures, and a qualitative synthesis of lecturer feedback.

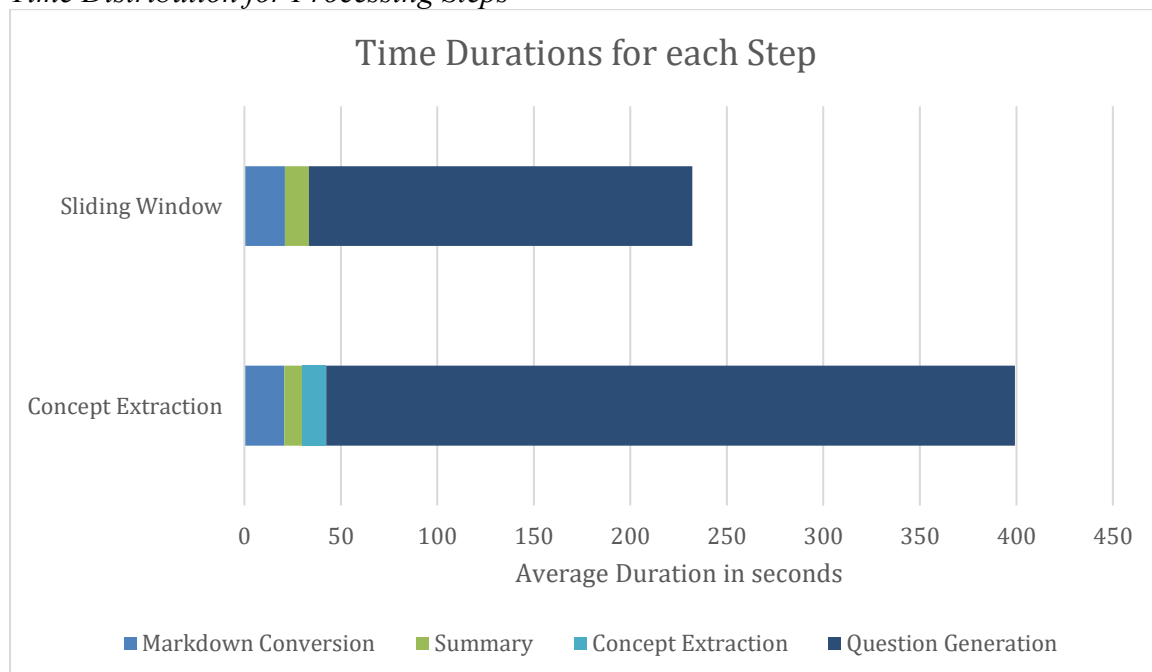
4.1 Prototype Results

4.1.1 Generated Questions

The prototype evaluation produced 84 sets of questions, providing a robust dataset for analyzing system performance. The data indicates that the total processing duration is primarily driven by the choice of context preparation rather than the cognitive framework. As shown in Figure 39, the sliding window approach completes the process in approximately 232 seconds, whereas concept extraction requires 399 seconds for the initial set. Notably, markdown conversion and summary generation are one-time operations, meaning subsequent question sets for the same document benefit from a 30-second reduction in processing time.

Figure 39

Time Distribution for Processing Steps



To gain deeper insights into these temporal drivers, Ordinary Least Squares (OLS) linear regression was used to construct predictive functions. For clarity and precision, only coefficients significant at the 5% level are included. Categorical base values are retained in the formulas for completeness but are assigned a zero multiplier. The conversion of documents to markdown requires an average of 21 seconds. Since the document's language did not significantly impact this phase, the process is modeled as:

$$\begin{aligned}
\text{MarkdownConversionTime} = & -11.0121s \\
& +0.0003s \times \text{NumCharacters} \\
& + \begin{bmatrix} 0 \\ 11.7124 \\ 3.852 \end{bmatrix} s \cdot \begin{bmatrix} \text{Script} \\ \text{TextSlides} \\ \text{ImageSlides} \end{bmatrix} \\
& + \begin{bmatrix} 8.4754 \\ 0 \\ 4.2883 \end{bmatrix} s \cdot \begin{bmatrix} \text{Math} \\ \text{Eng} + \text{CS} \\ \text{General} \end{bmatrix}
\end{aligned} \tag{1}$$

While summary generation averaged 11 seconds, the regression analysis yielded no statistically significant coefficients for this step. In contrast, concept extraction averaged 12 seconds and was significantly influenced by the total number of chunks processed. Although the intercept for this model was not statistically significant, it is included to provide context for the slope in the following function:

$$\text{ConceptExtractionTime} = 6.5584s + 3.7282s \times \text{TotalChunks}$$

The final question generation phase represents the most time-intensive component of the pipeline, averaging 278 seconds. This duration can be accurately modeled by accounting for the specific variables inherent to the generation process:

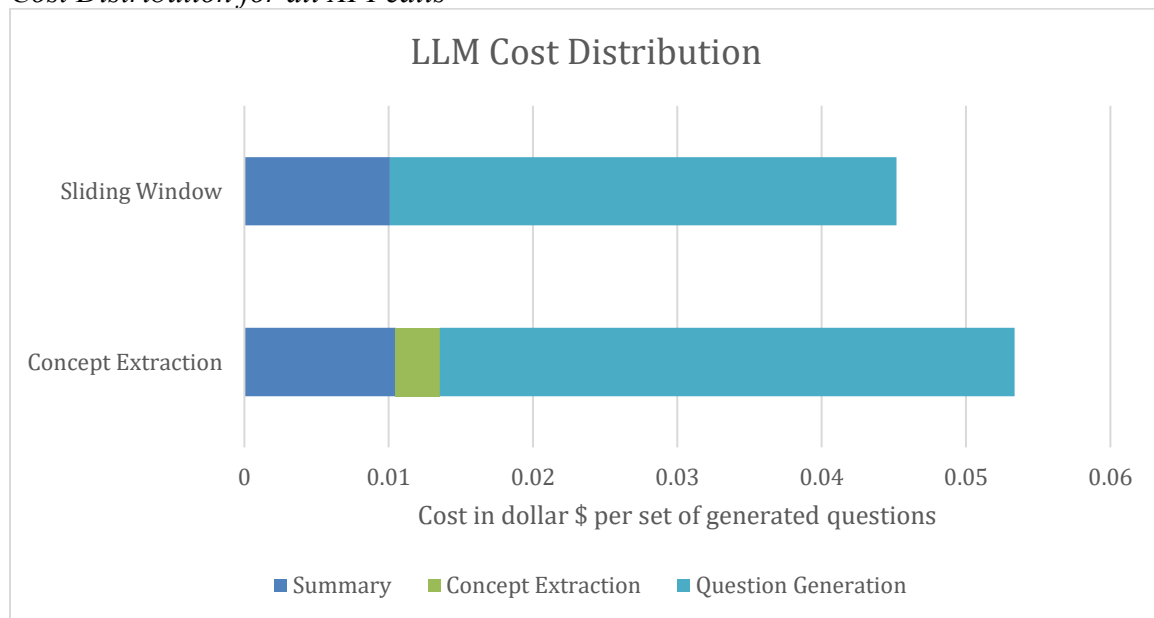
$$\begin{aligned}
\text{QuestionGenerationTime} = & 139.8389s \\
& +122.1277s \times \text{TotalQuestions} \\
& -123.4381s \times \text{InvalidQuestions} \\
& + \begin{bmatrix} -157.7938 \\ 0 \end{bmatrix} s \cdot \begin{bmatrix} \text{SlidingWindow} \\ \text{ConceptExtraction} \end{bmatrix} \\
& + \begin{bmatrix} 0 \\ -19.9586 \\ -163.3933 \end{bmatrix} s \cdot \begin{bmatrix} \text{Script} \\ \text{TextSlides} \\ \text{ImageSlides} \end{bmatrix} \\
& + \begin{bmatrix} 171.2582 \\ 0 \\ 161.6647 \end{bmatrix} s \cdot \begin{bmatrix} \text{Math} \\ \text{Eng} + \text{CS} \\ \text{General} \end{bmatrix}
\end{aligned} \tag{2}$$

4.1.2 Costs generated

In alignment with the observed time requirements, the financial cost of question generation is notably higher for the concept extraction pipelines compared to the sliding window approach. As illustrated in

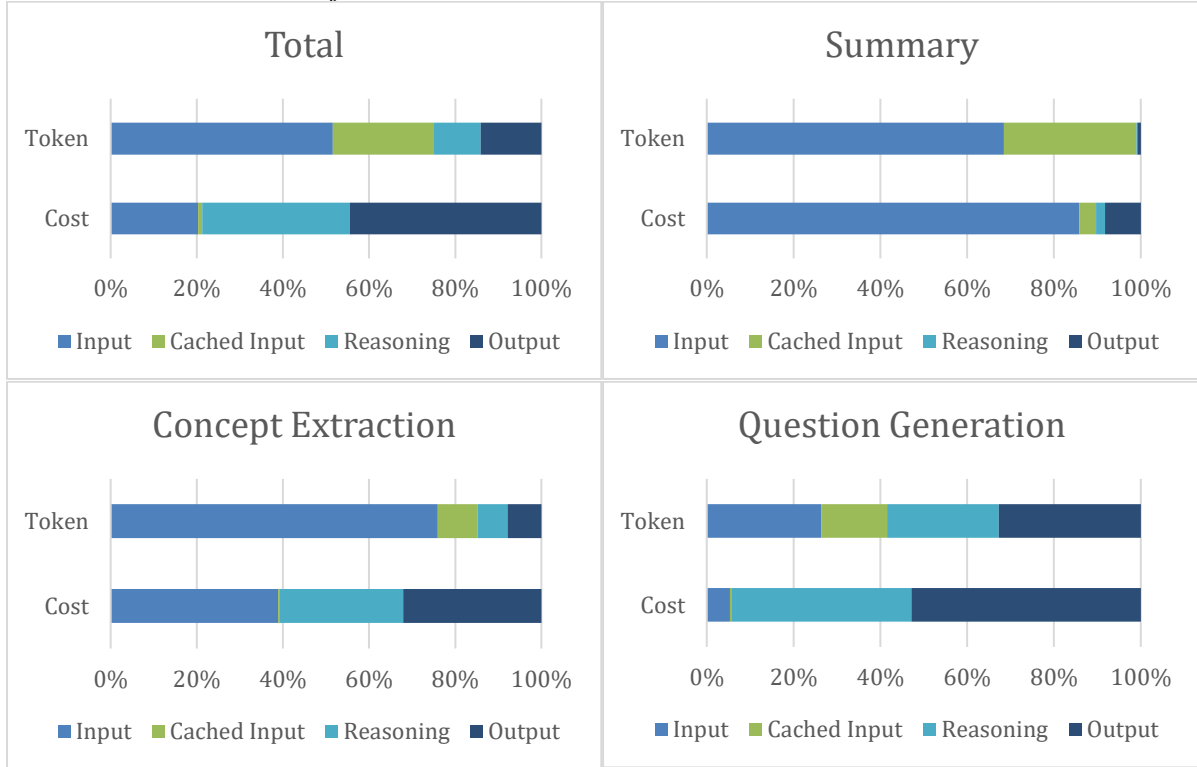
Figure 40, the average total cost for a sliding window pipeline is \$0.0452, whereas concept extraction increases this to \$0.0534. While the cost of summary generation remains constant across both methods, the extraction of specific concepts adds an average of \$0.00472 to the generation of a single question set.

These costs reflect the use of the GPT-5-mini model via the OpenAI API, with all processing conducted during standard daytime hours (08:00–22:00). Because the initial summary is a one-time expense, subsequent question sets for the same document are approximately \$0.01 cheaper.

Figure 40*Cost Distribution for all API calls*

The relationship between token volume and total cost is disproportionate, as shown in the percentual distribution in Figure 41. Reasoning and output tokens constitute only 25% of the total token count but account for nearly 80% of the total cost. Conversely, the implementation of input caching proved highly effective; cached input tokens represent 31% of all input tokens but contribute to only 4% of the input cost. This demonstrates that the financial efficiency of the pipeline relies heavily on minimizing high-cost reasoning and output tokens while maximizing the use of the cache.

Figure 41
Percentual Distribution of Tokens and their Costs



To model these costs more precisely, OLS linear regression was applied. Known token rates were utilized as offsets for input and cached input tokens to isolate the influence of other variables. The cost of generating a summary is modeled by the following function, where the intercept is included to preserve the relationship between coefficients despite its lack of individual statistical significance:

$$\begin{aligned}
 \text{SummaryCost} = & -0.0003\$ \\
 & +2.5 \times 10^{-7}\$ \times \text{InputTokens} \\
 & +2.5 \times 10^{-8}\$ \times \text{CachedInputTokens} \\
 & + \begin{bmatrix} 0 \\ -0.0086 \\ -0.0075 \end{bmatrix} \$ \cdot \begin{bmatrix} \text{Script} \\ \text{TextSlides} \\ \text{ImageSlides} \end{bmatrix} \\
 & + \begin{bmatrix} 0.0120 \\ 0 \\ 0.0009 \end{bmatrix} \$ \cdot \begin{bmatrix} \text{Math} \\ \text{Eng} + \text{CS} \\ \text{General} \end{bmatrix}
 \end{aligned} \tag{3}$$

The cost of extracting concepts from a single chunk is modeled as shown below. During analysis, document language was found to be insignificant and was subsequently removed from the model:

$$\begin{aligned}
 \text{ConceptExtractionCost} = & 0.0006\$ \\
 & +2.5 \times 10^{-7}\$ \times \text{InputTokens} \\
 & +2.5 \times 10^{-8}\$ \times \text{CachedInputTokens} \\
 & + \begin{bmatrix} 0 \\ -9.038 \times 10^{-5} \\ 0.0002 \end{bmatrix} \$ \cdot \begin{bmatrix} \text{Script} \\ \text{TextSlides} \\ \text{ImageSlides} \end{bmatrix} \\
 & + \begin{bmatrix} -0.0002 \\ 0 \\ 6.205 \times 10^{-5} \end{bmatrix} \$ \cdot \begin{bmatrix} \text{Math} \\ \text{Eng} + \text{CS} \\ \text{General} \end{bmatrix}
 \end{aligned} \tag{4}$$

Finally, the cost for generating a set of questions from a single chunk or concept is described by the function below. Interestingly, the choice of context preparation and cognitive framework did not significantly impact the cost in this phase. The primary driver is the "ValidChunk" variable, which indicates whether a chunk contains testable material:

$$\begin{aligned}
 \text{QuestionGenerationCost} = & 0.0088\$ \\
 & + 0.034\$ \times \text{ValidChunk} \\
 & + 2.5 \times 10^{-7}\$ \times \text{InputTokens} \\
 & + 2.5 \times 10^{-8}\$ \times \text{CachedInputTokens} \\
 & + \begin{bmatrix} 0 \\ 0.0052 \\ -0.0040 \end{bmatrix} \$ \cdot \begin{bmatrix} \text{Script} \\ \text{TextSlides} \\ \text{ImageSlides} \end{bmatrix} \\
 & + \begin{bmatrix} 0.039 \\ 0 \\ 0.0026 \end{bmatrix} \$ \cdot \begin{bmatrix} \text{Math} \\ \text{Eng} + \text{CS} \\ \text{General} \end{bmatrix} \\
 & + \begin{bmatrix} 0 \\ -0.0054 \end{bmatrix} \$ \cdot \begin{bmatrix} \text{English} \\ \text{German} \end{bmatrix}
 \end{aligned} \tag{5}$$

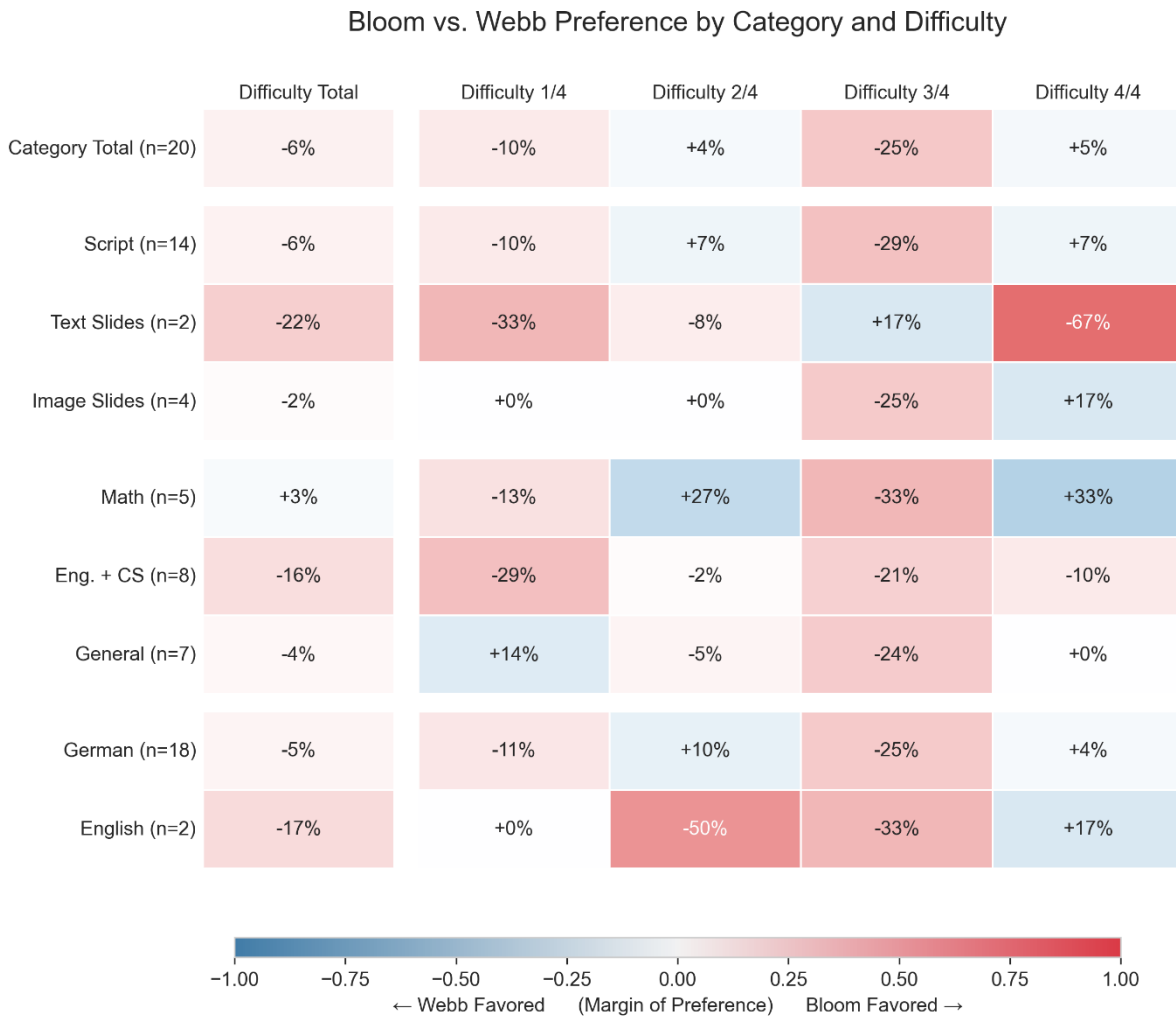
4.2 Survey Results

4.2.1 Quantitative Results

The survey results provide a comprehensive look at how different pipeline configurations perform across various academic contexts. By applying the win-rate method to the participant rankings, it was possible to isolate preferences for specific cognitive frameworks and context preparation methods. Overall, Webb's DOK was slightly favored over Bloom's Revised Taxonomy by a 6% margin. However, this preference was not uniform across all levels of complexity; Bloom's Taxonomy was actually preferred at difficulties two and four. These nuances are detailed in the heatmap in Figure 42, which visualizes the win counts of Bloom against Webb across document types, languages, and content categories calculated using the following formula:

$$\frac{\text{Bloom} - \text{Webb}}{\text{Bloom} + \text{Webb}} \tag{6}$$

Figure 42
Bloom vs. Webb Survey Preferences



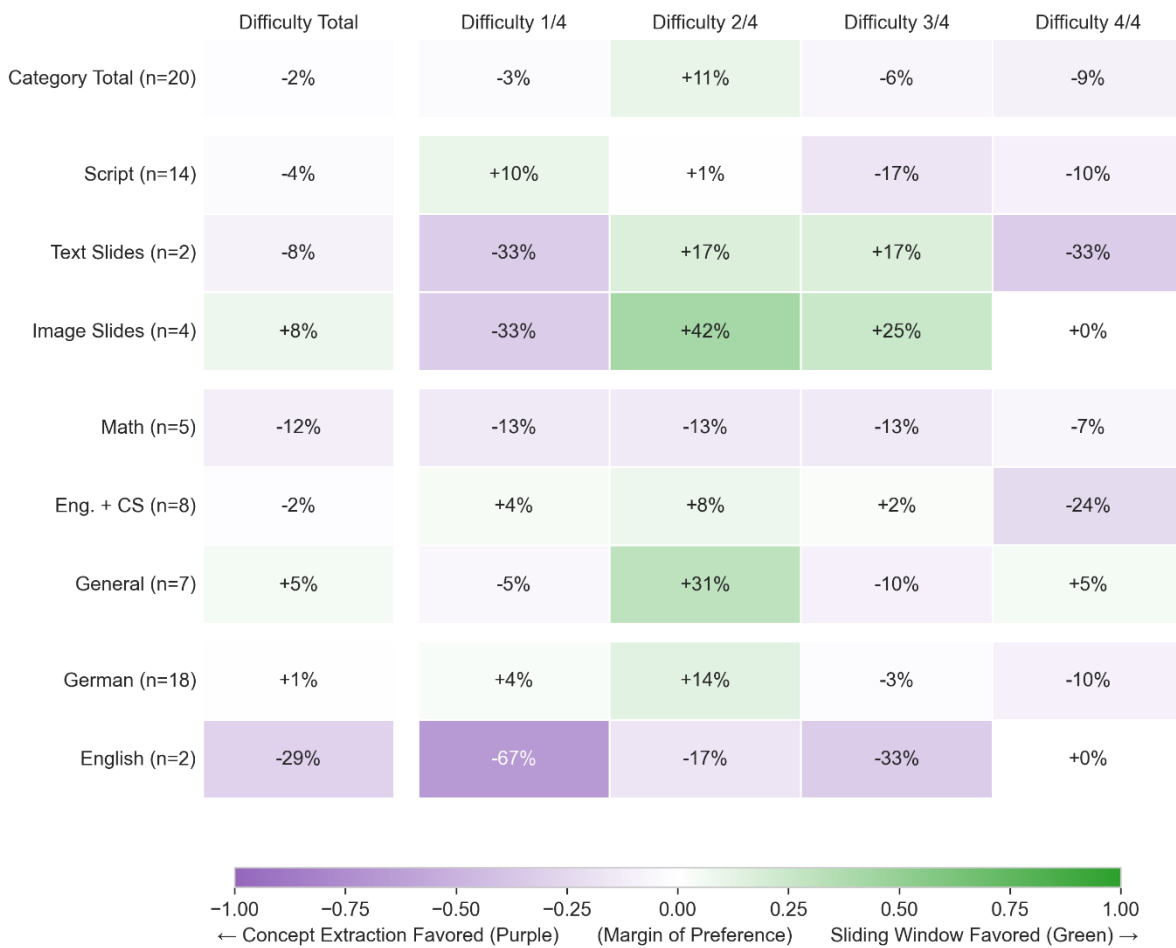
A similar analysis was conducted for context preparation, comparing the Sliding Window method against Concept Extraction. As shown in Figure 43, Concept Extraction was favored overall by a narrow 2% margin. This preference became more pronounced at the highest difficulty level and within mathematics-focused courses. The percentages in the figure are calculated as follows:

$$\frac{SW - CE}{SW + CE} \tag{7}$$

Figure 43

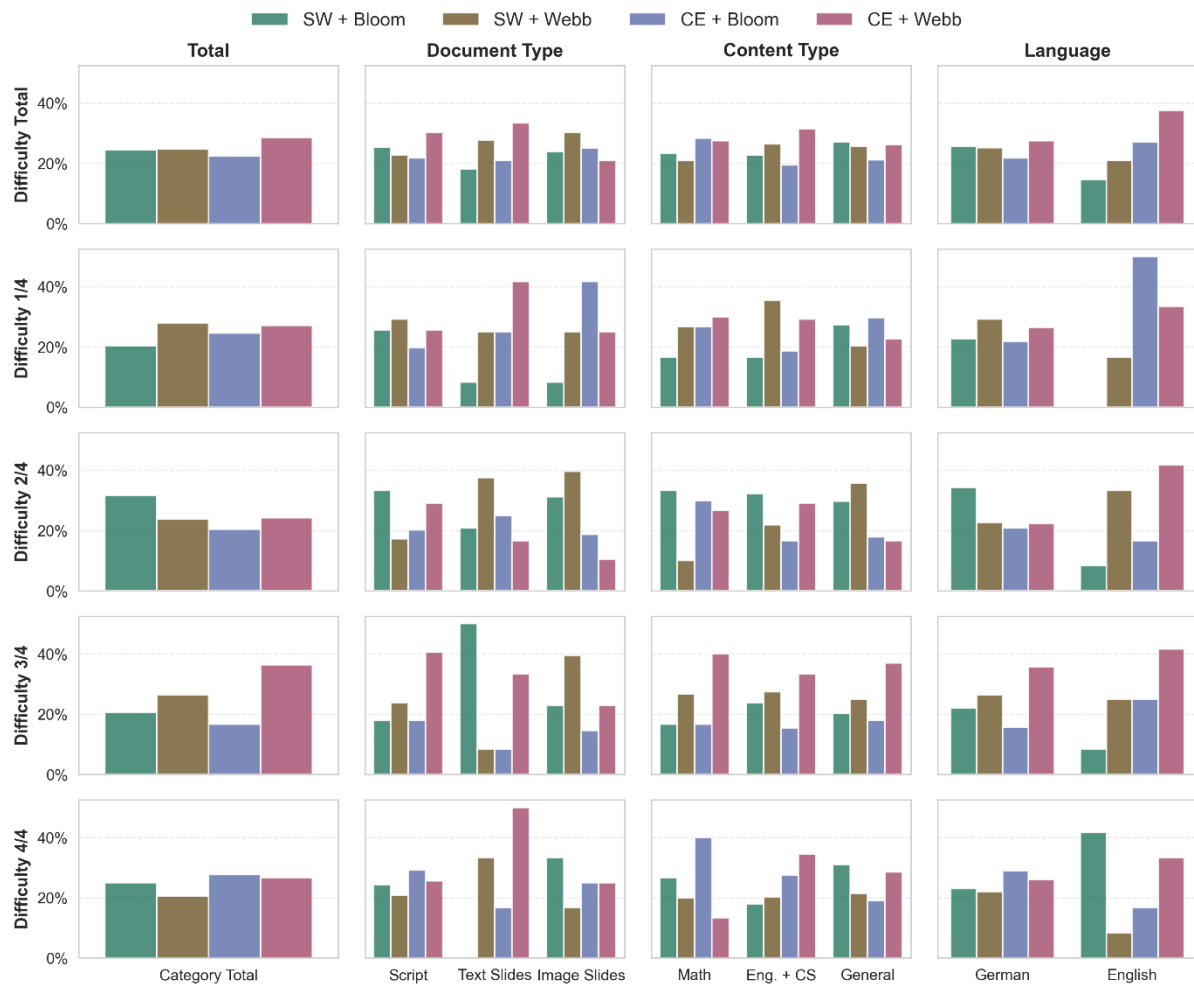
Sliding Window vs. Concept Extraction Survey Preferences

Sliding Window vs. Concept Extraction Preference by Category and Difficulty



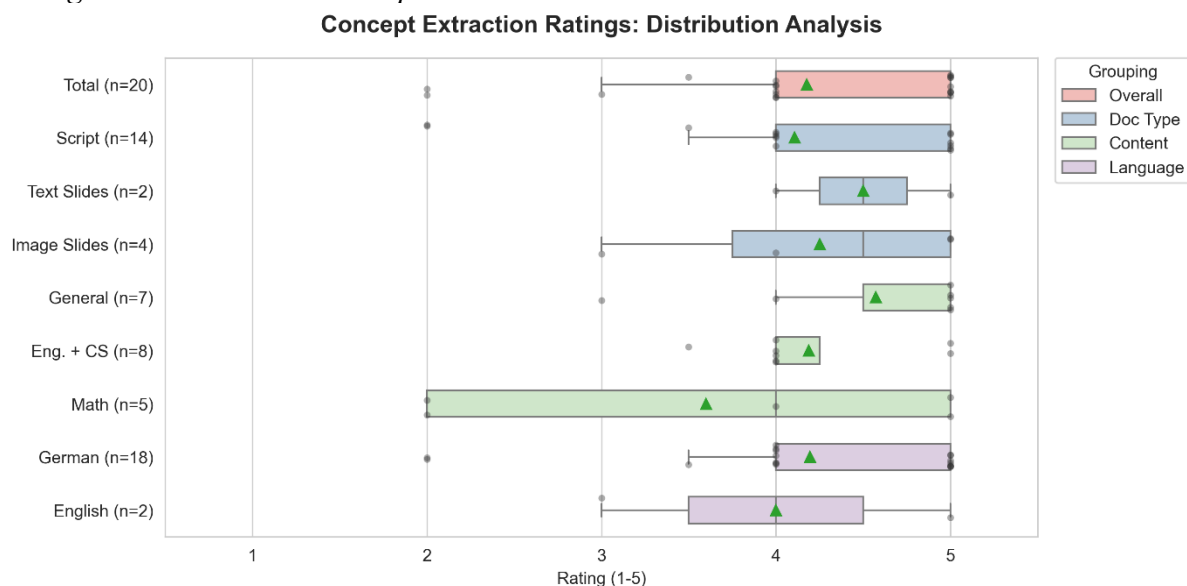
When evaluating the four pipelines as complete units in Figure 44, the combination of Concept Extraction with Webb’s DOK emerged as the most preferred configuration, mirroring the overall preference of the previous two heatmaps, a trend especially visible at the third difficulty level.

Figure 44
Pipeline Survey Preferences



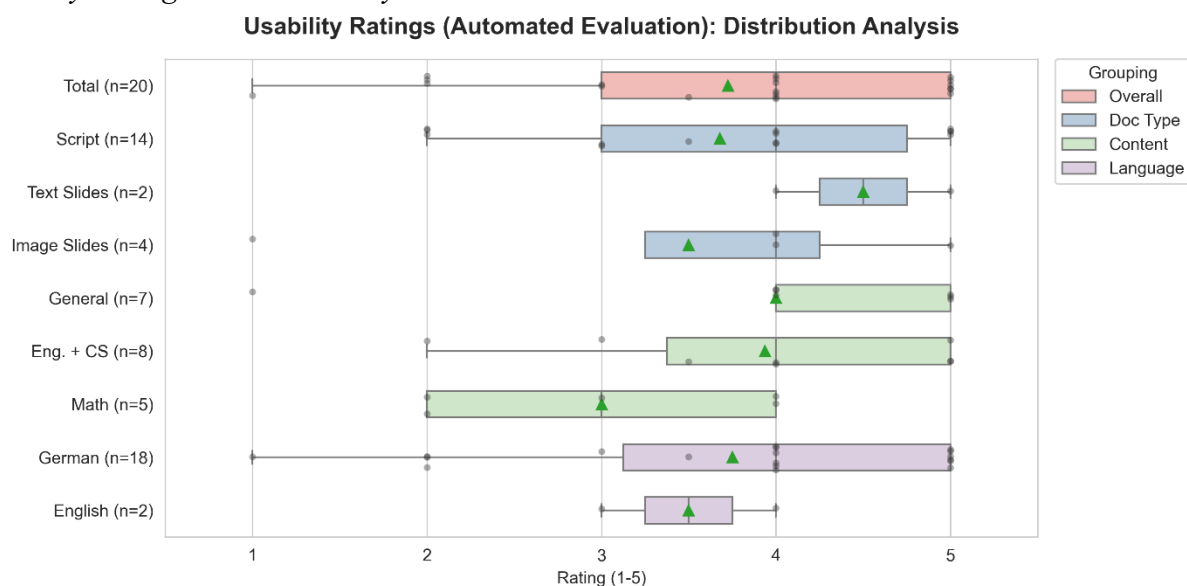
The quality of the underlying technology was also validated by the lecturers, who rated the suitability of the extracted concepts on a 1-to-5 scale. With an average rating of 4.175, the results were consistently skewed toward the "suitable" end of the spectrum across all groups, as illustrated by the boxplots in Figure 45.

Figure 45
Ratings about Extracted Concepts



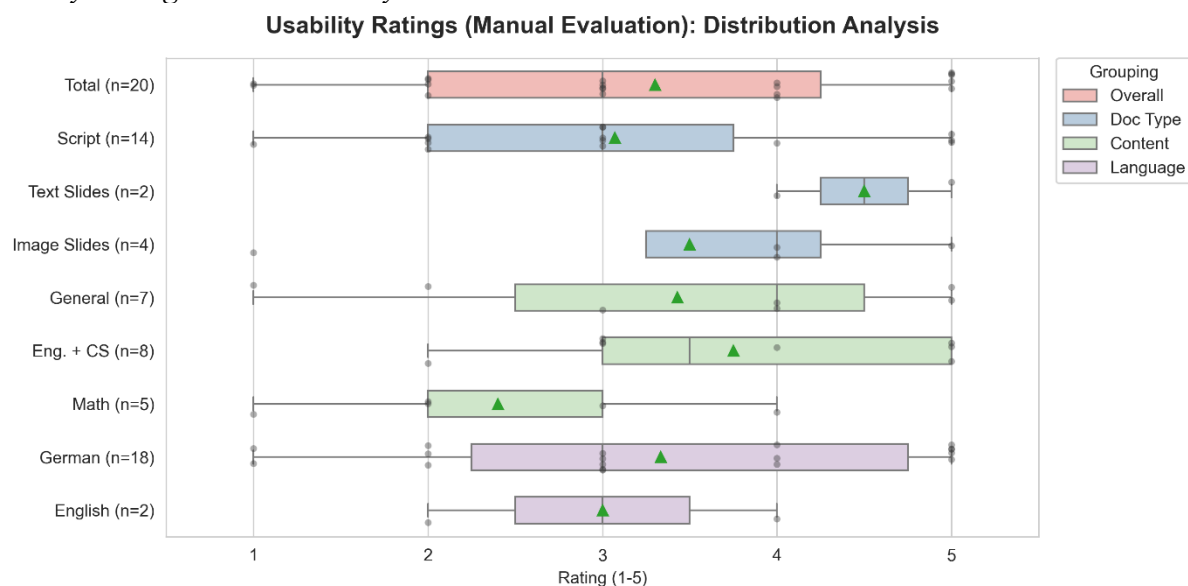
Finally, the survey explored the practical utility of these pipelines in a classroom setting. Lecturers expressed a clear willingness to adopt these generated questions if automated evaluation, performing at a human-equivalent standard, were available, resulting in an average usability rating of 3.725. While math-focused courses were more neutral in this regard, most other groups leaned toward adoption, as seen in Figure 46.

Figure 46
Survey Ratings about Usability with Automated Evaluation



However, usability significantly declines if lecturers must grade the open-ended answers manually. In that scenario, the average rating drops to 3.3, with mathematics coordinators explicitly tending toward non-adoption. These findings are visualized in Figure 47.

Figure 47
Survey Ratings about Usability with Manual Evaluation



4.2.2 Qualitative Results

To analyze the qualitative survey results systematically, Gioia's data structure was applied that first, splits all comments and feedback into smaller paraphrases (1st order concepts) and then group them thematically into 2nd order themes. Afterwards the concepts are aggregated for each theme into a short summary. The detailed Gioia's Data Structure can be viewed in Appendix 9.5. Four core themes were identified:

- Content Alignment and Scope Accuracy
- Linguistic Formulation and Structural Logic
- Didactic Appropriateness and Difficulty Calibration
- Practical Utility, Acceptance, and Integration

4.2.2.1 Content Alignment and Scope Accuracy

This dimension captures the degree to which the generated questions faithfully adhere to the specific boundaries of the lecture material and learning objectives. It aggregates feedback concerning the "validity" of the content, highlighting a discrepancies between the source text provided by the lecturers and the output generated by the AI.

The first issue within this dimension is extraneous information (hallucinations). There was a pattern where the system integrates knowledge that is not present in the lecture notes. This results in questions that are factually correct in a general context but invalid for the specific exam context (e.g., "aim clearly beyond the learning objectives").

The second issue is focusing misalignment where the system fails to distinguish between core content and peripheral examples. It generates questions based on introductory metaphors or minor mentions, while neglecting core competencies.

The last issue in this dimension are missing assessment modes. The inability of the generated output to cover specific necessary modalities of testing, specifically the absence of quantitative or calculation-based tasks (e.g., "no calculation tasks") in technical modules where they required.

4.2.2.2 Linguistic Formulation and Structural Logic

This dimension groups feedback regarding the textual construction, clarity, and internal coherence of the generated questions. It focuses on how the content is presented

rather than what the content is, addressing issues of syntax, length, logical flow, and stylistic consistency.

The participants pointed out ambiguities where questions were “too open,” “imprecise,” or lacked necessary context (e.g., “missing images or municipal context”). Conversely, some questions were criticized for being self-answering (e.g., “Questions answers itself”), indicating a failure in logical construction.

There is a clear distinction drawn between cognitive difficulty and textual complexity. Participants criticized “long task texts” that are “daunting and confusing, “ explicitly stating that difficulty should not be achieved merely by lengthening the question or using complicated formulations.

Linguistic critiques, such as the inconsistent use of formal and informal address (“Du” vs. “Sie”), the use of non-subject-related terms, and a language level that is too high for non-native English speakers were mentioned.

Lastly, several codes point to “illogical” or “unrealistic” formulations where the internal logic of the question breaks down, or where the task is chemically/physically impossible without further data (e.g., “does not work without an image”).

4.2.2.3 Didactic Appropriateness and Difficulty Calibration

This dimension consolidates feedback regarding the educational validity of the questions. It evaluates whether the generated output aligns with the target student level (e.g., semester, degree type), accurately reflects intended difficulty scales (e.g., Bloom’s Taxonomy), and adheres to the specific “logic” or mode of thinking required by the discipline.

A recurring concern is that the questions do not fit the specific semester level or student profile. Questions are described as either “too complex” for intermediate students or, conversely, “below the University of Applied Sciences (FH) level.” This includes the need to account for specific student backgrounds (e.g., technical students taking a business module).

Inconsistent difficulty calibration was mentioned by some participants where they mentioned that difficulty two was more difficult than difficulty one and difficulty three and four being “too far-reaching.” Inconsistencies where questions within the same level vary more than questions between different levels were also reported. The Bloom levels were also seen as inconsistent, which happened because in the survey, difficulty two and three contained multiple questions in one variant to accommodate the six levels of Bloom in four difficulties (e.g., “Understand” is better in one version, “Apply” in another).

A specific insight is the misalignment between “scientific logic” (linear, causal) and “design logic” (associative, non-linear). The tool’s tendency to force linear causality was seen as unsuitable for visual/design professions, “dissolving connections and hierarchies.”

Lastly, there is a misalignment between questions that merely ask for reproduction (which some found useful for basics) and those requiring “genuine transfer and judgment skills.” Positive feedback highlighted when questions successfully forced transfer, while negative feedback criticized questions that failed to test “core competencies” like quantitative estimation or “thinking further.”

4.2.2.4 Practical Utility, Acceptance, and Integration

This dimension groups feedback related to the adoption, implementation, and future potential of the question generation tool. It reflects the participants’ willingness to use the output, the necessary human-in-the-loop interventions (adaption/editing), and specific feature requests for better workflow integration.

The most consistent sentiment is that while the questions are not “exam-ready” immediately, they are highly valuable as a draft or inspiration. Participants stated they would

“adapt them slightly,” use them as “food for thought,” or use the tool as “inspiration to generate variants.”

Participants clearly identified where the tool fits best for them: “helpful for studying” (student-facing), “multiple-choice questions,” “pure knowledge questions,” and partially for the final module exam (MEP) if adapted.

There is a strong emphasis on the need for oversight, particularly regarding automated grading. Participants are willing to trust automation for “simple questions” but require manual double-checking for “answers with room for interpretation.”

The feedback includes explicit questions about the tool’s capabilities (“work based on slides?”, “create the answers?”, “multiple-choice?”) and suggestions for improvement (“Provide old MEPs as templates”).

4.3 Findings

4.3.1 Interpretation of Results

Efficiency and Quality Trade-off

The findings reveal a distinct trade-off between operational efficiency and output quality. While the Sliding Window method is significantly faster and 18% less expensive than Concept Extraction, the survey data justifies the higher investment of the latter. Concept Extraction was preferred by lecturers for high-difficulty questions and technical, math-focused content. This suggests that while Sliding Window is a viable, cost-effective choice for general subjects, the structural depth provided by Concept Extraction is essential for technically demanding disciplines. Financial efficiency was further bolstered by cached input tokens, which reduced initial input costs by 27%, a saving that compounds as more questions are generated from the same document.

Pedagogical Framework Efficacy

Regarding pedagogical efficacy, Webb’s DOK held a slight 6% preference over Bloom’s Revised Taxonomy. Qualitative feedback indicated that Bloom’s Taxonomy occasionally suffered from inconsistent difficulty calibration between the “Understand” and “Apply” levels. Furthermore, the analysis of “Design Logic” versus “Scientific Logic” suggests that neither framework perfectly captures non-linear, associative disciplines like visual design. Instructors also noted a need for “contextual balance,” observing that excessive background information in a prompt can sometimes obscure the core task for the student.

Human-in-the-Loop Necessity

The necessity of a “human-in-the-loop” approach remains evident. The high willingness to use the tool when paired with automated evaluation (3.725) contrasts with qualitative concerns regarding hallucinations. Instructors currently view the system as a powerful productivity aid for inspiration, “food for thought”, rather than a fully autonomous generator. The drop in usability ratings when manual evaluation is required (3.3) highlights that verification remains a significant bottleneck. While lecturers trust the system for retrieval, they demand oversight for complex, interpretive tasks.

Domain-Specific Limitations

Finally, domain-specific limitations were particularly visible in quantitative subjects. While the pipelines excel at linguistic formulation, they struggle with the procedural and calculative tasks required for mathematics and engineering exams. This gap in technical application explains the lower satisfaction scores among math coordinators and indicates that while the tool is highly effective for general knowledge, further development is needed to support procedural, calculation-based assessment.

4.3.2 Answering the Research Question

This study sought to determine the optimal pipeline architecture for transforming raw course materials into open-ended assessment pairs. The research question asked which combination of context preparation and cognitive framework balances technical quality, didactic soundness, and practical feasibility.

Based on the empirical data and lecturer feedback, the research question can be answered as follows:

Breadth and Depth

While both, the Sliding Window approach and Concept Extraction approach cover the whole document, Concept Extraction yielded higher user satisfaction, particularly in math-focused courses. This comes at a significant time (399s vs. 232s) and a slight cost increase. The Sliding Window approach is faster and cheaper, making it a viable option for non-math-focused courses.

Didactic Alignment

The comparison of cognitive frameworks reveals that Webb's Depth of Knowledge is the more effective framework for LLM-based question generation as it held a overall 6% preference over Bloom's revised Taxonomy.

Neither framework perfectly accommodates non-linear "design logic," suggesting that for creative disciplines, a different framework may be required.

Excessive contextualization might also reduce the quality of the generated questions, highlighting a trade-off between contextualization and clarity of the exercise.

Feasibility and Value

The study confirms that automated question generation is both economically and practically feasible, provided the role of the AI is correctly scoped. With an average cost of roughly \$0.05 for the first set (\$0.04 for following sets) the financial barrier to entry is negligible.

The high willingness to use rating (3.725) when automated evaluation is available confirms that lecturers value the tool. However, the value lies in its use as a draft generator rather than a fully autonomous replacement for human exam creation.

Final Verdict

The "best" pipeline architecture identified in this research is Concept Extraction combined with Webb's Depth of Knowledge. This configuration provides on average the highest quality output. However, to be "exam-ready," this pipeline requires a human-in-the-loop workflow to verify content accuracy, remove hallucinations, and adapt questions to the specific cultural context of the classroom.

5. Discussion

The results presented in Chapter 4 confirm that modern Large Language Models (LLMs) can bridge the gap between technical text processing and didactic assessment. However, the transition from an AI-generated draft to an exam-ready exercise remains a nuanced process that varies significantly across academic disciplines. This chapter synthesizes the empirical performance of the four developed pipelines with the theoretical gaps identified in the literature.

The objective of this discussion is to contextualize why certain architectures, such as Concept Extraction combined with Webb's Depth of Knowledge, outperformed others in expert evaluations. It explores the tension between "Scientific Logic," which the pipelines handle with high proficiency, and "Design Logic," where automated systems still struggle to capture non-linear associations. Furthermore, this chapter evaluates the shift in the educator's role from a content creator to a "human-in-the-loop" validator.

5.1 Discussion of Findings

The research gaps identified in the literature review (Table 6) served as the foundation for this study. Table 10 revisits these gaps to demonstrate how the implemented solutions contributed to the literature.

Table 10
Revisited GAP-Table

Researcher	Insight	Knowledge Gap	Project Contributions
Ch & Saha (2020, 2023)	Workflow for MCQ generation; 95% of MCQs are middle-school worthy.	MCQs do not promote higher-order cognitive processes.	Shifted focus to generating complex, open-ended questions using GPT-5-mini to support higher-order thinking (Bloom's & Webb's), moving beyond simple fact retrieval.
Killawala et al. (2018)	LSTM/NLP generates True/False, MCQ, and Fill-in-the-blank.	Only generates short, context-poor questions that lack higher-order cognitive depth.	Used GPT-5-mini as a "drafting engine" for open-ended assessments, leveraging large context windows to ensure depth and context richness lacking in pure NLP methods.
Lee et al. (2024)	Defining task type and format allows for different questions.	Requires human involvement per question; binary correct/incorrect data lacks insight into student understanding depth.	Automated the pipeline to require only a single file upload (PDF/DOCX) rather than per-question interaction, while providing difficulties rather than binary outputs for insights into student understanding depth.
Nguyen et al. (2022)	Summary data and secondary LLM validation improves quality.	Relies on structured formats (XML/HTML), not unstructured files like PDFs.	Developed pipelines capable of processing unstructured authentic materials (PDF, PowerPoint, Word) directly, removing the need for pre-existing XML/HTML data.
Marti A. Hearst (1997)	Term repetition shows "acceptable" performance for text segmentation.	Inconsistent performance compared to machine learning; limited assumptions.	Implemented recursive chunking, which uses document boundaries (chapters, paragraphs) to achieve similar performance with greater speed, interpretability, and lower cost than LLM-based chunking.
Koshorek et al. (2018); Jiang et al. (2021)	Supervised learning/Micro discourse trees outperform benchmarks.	Unknown generalization to new topics; training requires significant data and resources.	Utilized rule-based recursive chunking, avoiding the generalization issues and high resource costs associated with training specific machine learning models for segmentation.
Duarte et al. (2024)	Dynamic segmentation achieves SOTA retrieval performance.	High costs due to excessive API calls for segmentation.	Made use of recursive chunking, which is a fast and local method of text segmentation.
Noorbakhsh et al. (2025)	Savaal framework improves testing quality and usability.	Not used for open-ended questions; ranking concepts may miss relevant content.	Adapted Concept Extraction for open-ended questions and ensured fuller coverage (vs. just high-ranking concepts) by offering a Sliding Window alternative that covers the whole document.
Kevin Hwang et al. (2023)	Complexity alignment between GPT and human assessment exists.	No clear quality alignment; small models failed to aid evaluation.	Replaced small model validation with the intrinsic reasoning capabilities of GPT-5-mini, conducting a direct empirical comparison between Bloom's and Webb's frameworks.
Elkins et al. (2024); Maity et al. (2025); Zhuge et al. (2025); Scaria et al. (2024); Duong-Trung et al. (2024)	Few-shot learning, CoT, and fine-tuning improve alignment.	Studies use older LLMs (GPT-3.5), lack prompt structure guidelines, and incur high fine-tuning costs.	Utilized GPT-5-mini (two generations ahead), defined specific prompt structures (5-shot, Chain-of-Thought, Quality Checks), and proved high-quality output is possible without expensive fine-tuning.
Villarroel et al. (2018); Morrison et al. (2019)	Active learning and authentic assessments improve clarity and intent.	These studies do not cover questions generated with LLMs.	Integrated these didactic principles (realism, contextualization, concrete instructions) directly into the system prompt, successfully applying non-LLM pedagogical theory to AI generation.

Note: More relevant Knowledge Gaps are highlighted with a gray background

5.1.1 Technical Discussion

A significant portion of existing literature relies on structured data formats to generate educational content. Previous work, such as that by Nguyen et al. (2022), required pre-formatted XML or HTML inputs to function correctly. This study advances the field by demonstrating a robust method for processing unstructured, authentic course materials like PDFs and Word documents directly. By implementing a pipeline that handles raw lecture slides without manual conversion, this work bridges the gap between theoretical models and the practical reality of university file management.

Regarding text segmentation, the literature has historically oscillated between rule-based approaches and complex machine learning models. While Hearst (1997) established term repetition as a viable segmentation method, later studies by Koshorek et al. (2018) and Duarte et al. (2024) pivoted toward heavy deep-learning or LLM-based segmentation to achieve state-of-the-art performance. This thesis challenges the necessity of such

computationally expensive methods for educational purposes. The results show that recursive chunking, a rule-based approach using natural document boundaries, offers a superior balance of speed, interpretability, and cost. It avoids the generalization issues of trained models and the excessive API costs associated with dynamic LLM chunking.

Beyond segmentation, this study diverged from previous validation architectures. While studies like Kevin Hwang et al. (2023) relied on a secondary language model to validate generated questions, this pipeline streamlined the process by leveraging the advanced intrinsic reasoning capabilities of GPT-5-mini, effectively replacing the need for an external validator. Furthermore, regarding prompt engineering, this project adopted a specific 5-shot learning strategy. In contrast to Maity et al. (2025), who found mixed results with intermediate shot counts, this approach included two negative examples to explicitly teach the model how to reject chunks with no testable content, alongside diverse positive examples from statistics, biology, and politics to ensure generalizability across disciplines.

In the domain of context retrieval, this study builds upon and refines the retrieval-augmented strategies proposed by Noorbakhsh et al. (2025). Their "Savaal" framework focused exclusively on extracting high-ranking concepts for Multiple Choice Questions (MCQs), risking the exclusion of peripheral but necessary details. This project introduces a novel comparison between Sliding Window and Concept Extraction architectures for open-ended questions. The findings reveal a crucial trade-off: while Concept Extraction is statistically preferred for technical depth, the Sliding Window approach requires a lower cost. This contribution offers a nuanced alternative to the "retrieval-only" mindset, suggesting that total coverage is sometimes necessary to capture the full narrative arc of a lecture.

The study establishes the economic feasibility of high-quality automated assessment. Previous research often utilized older, less efficient models or ignored cost-scaling entirely. By optimizing GPT-5-mini with input caching, this project reduced the cost of generating a question set to a negligible \$0.05. This finding is critical for institutional adoption, as it proves that sophisticated, reasoning-heavy generation is no longer financially prohibitive, democratizing access to advanced assessment tools.

However, two technical limitations remain. First, the exclusive reliance on text inputs reduces quality in visual-heavy courses, as the pipeline currently ignores diagrams and charts. Future iterations must implement multi-modal pipelines to interpret the visual data central to engineering curricula. Second, despite the low cost, the generation process requires several minutes per chunk. A practical solution for scaling would be a hybrid processing model: processing short documents immediately while scheduling longer files for overnight batch processing. This would not only mitigate latency issues but could further reduce API costs by utilizing the 50% cheaper batch rates offered by providers.

5.1.2 Didactical Discussion

The primary didactical shift presented in this thesis is the move from simple knowledge retrieval to complex proficiency estimation. While Ch & Saha (2020, 2023) and Killawala et al. (2018) focused on generating MCQs or fill-in-the-blank exercises, this study demonstrates that modern LLMs can successfully generate open-ended, higher-order questions. By utilizing a 5-shot prompt structure with Chain-of-Thought reasoning, the pipelines produced questions that go beyond rote memorization to test analysis and evaluation skills. This directly challenges the assumption in earlier literature that automated generation is limited to testing lower-order cognitive skills.

A key novelty of this research is the direct empirical comparison between Bloom's Revised Taxonomy and Webb's DOK. Prior studies, such as those by Duong-Trung et al. (2024) and Maity et al. (2025), focused almost exclusively on optimizing Bloom's Taxonomy. This study introduces a divergent perspective, finding that Webb's DOK resulted

in higher satisfaction among university lecturers, particularly in technical fields. This suggests that Webb's focus on complexity and cognitive demand may be better suited for LLM instructions than Bloom's focus on cognitive processes, influencing how future automated systems should be architected for higher education.

The preference for Webb's framework likely stems from the inherent ambiguities observed in Bloom's Taxonomy during the study. Qualitative feedback indicated that the perceived difficulty of an exercise often did not match its assigned label. This inconsistency arises because Bloom's levels are categorized by cognitive processes (e.g., "Understand" vs. "Apply") which feature overlapping difficulty curves, rather than strict complexity tiers. Conversely, while Webb's levels are less rigidly defined in general use cases, they offered a clearer distinction in complexity for technical subjects, where a Level 2 DOK task often demanded higher cognitive load than a mid-tier Bloom task.

Despite these advancements, the results align with the caution expressed by Lee et al. (2024) regarding the necessity of a "human-in-the-loop." While the pipelines reduce the manual burden of creation, they introduce a new requirement for oversight. Qualitative feedback highlighted that AI still struggles with "Design Logic", the non-linear, associative thinking required in visual disciplines, and can occasionally hallucinate context. This reinforces the idea that AI should be viewed as a productivity multiplier that creates "food for thought," rather than an autonomous decision-maker.

Furthermore, this work successfully operationalizes non-computational didactical theories within an algorithmic environment. By integrating the principles of authentic assessment (Villarroel et al., 2018) and effective instructional design (Morrison et al., 2019) directly into the system prompts, the study shows that pedagogical quality can be "engineered" into AI outputs. Instructors noted that while excessive context sometimes distracted from the core task, the inclusion of realistic scenarios generally improved the utility of the questions. This confirms that the quality of AI-generated content is dependent not just on the model's intelligence, but on the pedagogical soundness of its instructions.

To bridge the remaining gap between a generated draft and a usable exam, future systems must move beyond a static prompt structure. The feedback suggests that quality could be drastically improved by allowing instructors to toggle specific constraints, such as selecting between "Semester Practice" and "End-term Exam" modes, or enforcing a "Calculation-only" requirement for mathematics modules. Enabling such customization would allow the system to adapt to the specific assessment strategy of the course, rather than applying generic "one-size-fits-all" logic.

6. Conclusion

The thesis aimed to identify the optimal pipeline architecture for transforming raw course materials into open-ended assessment questions with multiple difficulties. Through the implementation of four distinct pipelines and a survey of 20 module coordinators in higher education across seven different departments, the study concludes that a pipeline utilizing Concept Extraction combined with Webb's Depth of Knowledge yields the highest quality results. However, despite the advanced capabilities of GPT-5-mini, the technology is not yet capable of fully autonomous "exam-ready" generation. The human-in-the-loop remains an essential component for validity and didactic alignment.

6.1 Practical Implications

The primary implication of this study is the validation of LLM-based pipelines as highly efficient drafting engines. With an operating cost of roughly \$0.05 for the first generated question set, the tool serves as an inexpensive solution to overcome "writer's block" for instructors. It provides a diverse range of questions that lecturers can either adopt directly or refine slightly, significantly reducing the time investment required for question generation.

The tool allows instructors to shift their focus from creation to curation. Rather than drafting questions from scratch, educators can act as editors, validating the generated drafts against their specific learning objectives.

The study highlights that a "one-size-fits-all" approach has limitations. For text-heavy subjects, the tool is immediately useful. However, for math-focused or highly visual courses, the generated questions currently serve better as conceptual checks rather than calculation exercises. To maximize practical values, the pipeline requires an interface that allows for customization, such as toggling between "calculation-only" or "conceptual" modes, to align with the specific needs of the course.

For the project partner, Edisconet, these findings represent a viable pathway to evolve their platform from a course completion tracker to a proficiency estimation engine. By integrating the recommended modular pipeline, Edisconet can offer a scalable, low-cost solution that generates meaningful assessments, thereby addressing the marked need for measurable learning outcomes. However, given the necessity of a "Human-in-the-Loop," the feature should initially be positioned as an AI-assisted authoring tool for course creators, rather than a fully autonomous testing suite. This approach allows Edisconet to deliver immediate value in efficiency and assessment depth while mitigating the current risks the hallucinatory content.

6.2 Theoretical Implications

From a theoretical perspective, this research challenges the assumption that modern LLMs are ready for fully autonomous educational agency. While LLMs show strong reasoning capabilities, the persistence of "hallucinations" (extraneous information) and "focus misalignment" indicates that the models currently lack the contextual awareness to distinguish between core competencies and peripheral examples without human guidance.

The preference for Webb's DOK suggests that AI models respond better to instructions based on complexity of thinking (depth) rather than the type of cognition (Bloom's verbs). This implies that future prompt engineering for educational AI should prioritize difficulties categorized by task complexity rather than cognitive effort.

The identified friction between "Scientific Logic" (linear) and "Design Logic" (associative) implies that current LLM architectures favor linear causality. This theoretically

limits their effectiveness in creative or non-linear disciplines until multi-model or alternative logic frameworks are better integrated.

6.3 Limitations and further Research

While the prototype demonstrates significant potential, several limitations must be acknowledged. Primarily, the current pipeline relies on a text-only ingestion model. By ignoring visual data such as diagrams, charts, and technical drawings in slides, the system underperforms in “Design” and “Math” contexts where visual information is the primary carrier of meaning. The disconnect contributed to the lower satisfaction scores in visually intensive modules. Additionally, the study’s sample size was restricted to 21 course coordinators within the fields of Engineering and Architecture. Consequently, the findings may not generalize to the Humanities or Social Sciences, where assessment criteria often prioritize different cognitive nuances. Finally, despite the integration of “Chain-of-Thought” and quality checks, the model occasionally produced hallucinations or focused on peripheral metaphors rather than core competencies, confirming that the output is not yet valid enough for unmediated student exposure.

Furthermore, while the pipelines were designed to generate both open-ended questions and their corresponding key answers, the accuracy and didactical validity of these answers were not evaluated in this study due to time constraints. Consequently, while the questions show promise for inspiration, the reliability of the automated “solution key” remains unverified.

Future research should focus on the integration of Multi-model Large Language Models to enable the system to interpret and generate questions based on diagrams and visual data. Investigating a “Human-in-the-Loop” interface that allows instructors to provide real-time corrective feedback, which the model could use to refine subsequent question sets, would also be a significant step toward achieving higher validity. Furthermore, longitudinal studies are needed to evaluate how students interact with these AI-generated questions and whether the automated evaluation of student answers aligns with human grading across diverse disciplines. Finally, automated evaluation using the already generated key answers, has potential to drastically improve efficiency of lecturers and reduce the burden of reviewing open-ended questions in large classes.

6.4 Recommendations

Based on the empirical evidence, it is recommended that any production-level implementation of this system defaults to the Concept Extraction combined with Webb’s Depth of Knowledge architecture. This configuration provides the most robust results for the complex, technical documentation typical of higher education. To optimize user experience, the system should include a “Context Selector” allowing instructors to specify whether they require conceptual knowledge checks or quantitative calculation tasks, as the latter currently requires more specific prompting to be effective.

Furthermore, developers should refine the system prompt by incorporating the qualitative feedback gathered in this study. Specifically, stricter constraints must be placed on “external knowledge” to minimize hallucinations, and the “Persona” instruction should be tuned to maintain linguistic consistency (e.g., adhering to a specific formal or informal address). Finally, it is recommended that institutions view this tool as a productivity aid for lecturers rather than an autonomous examiner. Clear guidelines should be provided to instructors on how to curate and adapt these drafts to ensure they meet the specific cultural and academic standards of their individual classrooms.

7. Reflection on Project

7.1 Requirements

Most "MUST" and "SHOULD" requirements were either fulfilled or superseded by insights gained during the literature review. For instance, requirement F4, which limited user configuration to difficulty and filetype, was invalidated. Research indicated that generating all cognitive levels simultaneously improved question variety and quality, rendering difficulty presets (as per F9) obsolete. Similarly, the system became capable of deducing filetypes automatically. However, qualitative survey feedback suggested that future iterations should actually increase customization options to further enhance quality.

Performance requirement NF1, which mandated a processing time of under one minute per page, could not be met. While text extraction and chunking were efficient, the LLM question generation phase alone required several minutes per chunk. Technical adjustments, such as requirement T6 (version tracking) and T8 (Docker integration), were deprioritized or replaced. Version tracking offered little benefit for a solo developer, and an Anaconda environment was chosen over Docker for its ease of setup in a single-user, local environment.

Establishing a requirements catalogue was fundamentally worth the effort because it forced a concrete definition of the project's technical scope from the outset. While the initial detail proved too granular, leading to several requirements becoming non-applicable as the research progressed, the process served as a vital anchor. It ensured that even as the technical path shifted, the core objectives remained grounded in functional reality.

7.2 Risks

The most significant risk, F-R1 (LLM hallucinations), remained a persistent challenge. Complex questions require extensive context, and when the provided document chunks were insufficient, models occasionally hallucinated details to complete the task. While strict prompt engineering reduced these occurrences, they could not be eliminated entirely.

To address API bottlenecks and high costs, the model was shifted from GPT-5 to GPT-5-mini. This reduced costs fivefold but did not resolve the latency issues required to meet the one-minute-per-page benchmark. Future implementations might utilize batch API calls to further lower costs, accepting that question generation is inherently an asynchronous, rather than real-time, process.

Proactive risk management allowed for the early identification of technical bottlenecks, such as model hallucinations and API costs. Although some identified risks became irrelevant as the architecture evolved, the initial analysis prevented these issues from becoming project-critical failures. Having a pre-defined risk mitigation strategy made the pivot to GPT-5-mini a planned decision rather than a reactive crisis.

7.3 Project Management

The project timeline deviated early during the systematic literature review, the effort for which was initially underestimated. Furthermore, the early scheduling of the mid-term presentation shifted the evaluation phase, delaying the survey design. Consequently, the deadline for lecturer feedback was extended by three weeks to ensure a robust data set. Future projects should allocate more substantial time buffers for systematic reviews and establish firm "earliest" dates for major milestones to prevent cascading delays.

Developing a project plan was highly valuable, as it established necessary soft deadlines that kept the project on track. For example, missing the soft deadline for the

literature review served as a critical indicator that the pace needed to increase to maintain overall progress. This structural oversight provided a sense of urgency and clarity that was essential for navigating the complexities of the evaluation phase.

8. List of References

- Agarwal, M., Shah, R., & Mannem, P. (2011). Automatic question generation using discourse cues. *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, 1–9.
- Anderson, L. W. (Ed.). (2009). *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives* (Abridged ed., [Nachdr.]). Longman.
- Avi Chawla. (2024, October 18). 5 Chunking Strategies For RAG. *5 Chunking Strategies For RAG*. <https://blog.dailydoseofds.com/p/5-chunking-strategies-for-rag>
- Benjamin S. Bloom, Max D. Engelhart, Edward J. Furst, Walker H. Hill, & David R. Krathwohl. (1956). *Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook I: Cognitive Domain*. David McKay Company.
- Biggs, J. B., Biggs, J. B., & Collis, K. F. (1982). *Evaluating the quality of learning: The SOLO taxonomy (structure of the observed learning outcome)*. Academic Press.
- Borrego, A., Dessi, D., Hernandez, I., Osborne, F., Reforgiato Recupero, D., Ruiz, D., Buscaldi, D., & Motta, E. (2022). Completing Scientific Facts in Knowledge Graphs of Research Concepts. *IEEE Access*, *10*, 125867–125880.
<https://doi.org/10.1109/ACCESS.2022.3220241>
- Ch, D. R., & Saha, S. K. (2020). Automatic Multiple Choice Question Generation From Text: A Survey. *IEEE Transactions on Learning Technologies*, *13*(1), 14–25.
<https://doi.org/10.1109/TLT.2018.2889100>
- Ch, D. R., & Saha, S. K. (2023). Generation of Multiple-Choice Questions From Textbook Contents of School-Level Subjects. *IEEE Transactions on Learning Technologies*, *16*(1), 40–52. <https://doi.org/10.1109/TLT.2022.3224232>
- Dang, F.-R., Tang, J.-T., Pang, K.-Y., Wang, T., Li, S.-S., & Li, X. (2021). Constructing an Educational Knowledge Graph with Concepts Linked to Wikipedia. *Journal of*

Computer Science and Technology, 36(5), 1200–1211.

<https://doi.org/10.1007/s11390-020-0328-2>

Demaidi, M. N., Gaber, M. M., & Filer, N. (2017). Evaluating the quality of the ontology-based auto-generated questions. *Smart Learning Environments*, 4(1), 7.

<https://doi.org/10.1186/s40561-017-0046-6>

Dreyfus, H. L., Dreyfus, S. E., & Athanasiou, T. (1986). *Mind over machine: The power of human intuition and expertise in the era of the computer*. The Free Press.

Duarte, A. V., Marques, J. D., Graça, M., Freire, M., Li, L., & Oliveira, A. L. (2024).

LumberChunker: Long-Form Narrative Document Segmentation. *Findings of the Association for Computational Linguistics: EMNLP 2024*, 6473–6486.

<https://doi.org/10.18653/v1/2024.findings-emnlp.377>

Duong-Trung, N., Wang, X., & Kravčik, M. (2024). BloomLLM: Large Language Models Based Question Generation Combining Supervised Fine-Tuning and Bloom's Taxonomy. In R. Ferreira Mello, N. Rummel, I. Jivet, G. Pishtari, & J. A. Ruipérez Valiente (Eds.), *Technology Enhanced Learning for Inclusive and Equitable Quality Education* (Vol. 15160, pp. 93–98). Springer Nature Switzerland.

https://doi.org/10.1007/978-3-031-72312-4_11

Dutta, S., Ranjan, S., Mishra, S., Sharma, V., Hewage, P., & Iwendi, C. (2024). Enhancing Educational Adaptability: A Review and Analysis of AI-Driven Adaptive Learning Platforms. *2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM)*, 1–5.

<https://doi.org/10.1109/ICIPTM59628.2024.10563448>

Eager, B., & Brunton, R. (2023). Prompting Higher Education Towards AI-Augmented Teaching and Learning Practice. *Journal of University Teaching and Learning Practice*, 20(5). <https://doi.org/10.53761/1.20.5.02>

- Elkins, S., Kochmar, E., Cheung, J. C. K., & Serban, I. (2024). How Teachers Can Use Large Language Models and Bloom's Taxonomy to Create Educational Quizzes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(21), 23084–23091. <https://doi.org/10.1609/aaai.v38i21.30353>
- Embedding Leaderboard*. (n.d.). MTEB Leaderboard - a Hugging Face Space by Mteb. Retrieved December 26, 2025, from <https://huggingface.co/spaces/mteb/leaderboard>
- Fu, Y., Wang, Z., Yang, L., Huo, M., & Dai, Z. (2025). ConQuer: A Framework for Concept-Based Quiz Generation. *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, 92–104. <https://doi.org/10.18653/v1/2025.naacl-srw.9>
- Gan, W., Qi, Z., Wu, J., & Lin, J. C.-W. (2023). Large Language Models in Education: Vision and Opportunities. *2023 IEEE International Conference on Big Data (BigData)*, 4776–4785. <https://doi.org/10.1109/BigData59044.2023.10386291>
- Gemini 3 Pro*. (2025). Gemini 3 Pro - Google DeepMind. <https://deepmind.google/models/gemini/pro/>
- Ismail, S. M., Rahul, D. R., Patra, I., & Rezvani, E. (2022). Formative vs. summative assessment: Impacts on academic motivation, attitude toward learning, test anxiety, and self-regulation skill. *Language Testing in Asia*, 12(1), 40. <https://doi.org/10.1186/s40468-022-00191-4>
- Jiang, F., Fan, Y., Chu, X., Li, P., Zhu, Q., & Kong, F. (2021). Hierarchical Macro Discourse Parsing Based on Topic Segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14), 13152–13160. <https://doi.org/10.1609/aaai.v35i14.17554>

- Jodoi, K., Takenaka, N., Uchida, S., Nakagawa, S., & Inoue, N. (2021). Developing an active-learning app to improve critical thinking: Item selection and gamification effects. *Heliyon*, 7(11), e08256. <https://doi.org/10.1016/j.heliyon.2021.e08256>
- John, D., & Devi, G. S. (2021). Designing STEM-Specific Student-Friendly Reading Content for the Engineering English Classroom. *IEEE Transactions on Professional Communication*, 64(4), 444–455. <https://doi.org/10.1109/TPC.2021.3110419>
- Jones, E., Priestley, M., Brewster, L., Wilbraham, S. J., Hughes, G., & Spanner, L. (2021). Student wellbeing and assessment in higher education: The balancing act. *Assessment & Evaluation in Higher Education*, 46(3), 438–450. <https://doi.org/10.1080/02602938.2020.1782344>
- Karin K. Hess, Ben S. Jones, Dennis Carlock, & John R. Walkup. (2009). Cognitive Rigor: Blending the Strengths of Bloom’s Taxonomy and Webb’s Depth-of-Knowledge to Enhance Classroom-Level Processes. *ERIC*.
- Kevin Hwang, Sai Challagundla, Maryam Alomair, Lujie Karen Chen, & F. S. Choa. (2023). *Towards AI-Assisted Multiple Choice Question Generation and Quality Evaluation at Scale: Aligning with Bloom’s Taxonomy*.
- Killawala, A., Khokhlov, I., & Reznik, L. (2018). Computational Intelligence Framework for Automatic Quiz Question Generation. *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 1–8. <https://doi.org/10.1109/FUZZ-IEEE.2018.8491624>
- Koshorek, O., Cohen, A., Mor, N., Rotman, M., & Berant, J. (2018). Text Segmentation as a Supervised Learning Task. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 469–473. <https://doi.org/10.18653/v1/N18-2075>

- Lee, U., Jung, H., Jeon, Y., Sohn, Y., Hwang, W., Moon, J., & Kim, H. (2024). Few-shot is enough: Exploring ChatGPT prompt engineering method for automatic question generation in english education. *Education and Information Technologies*, 29(9), 11483–11515. <https://doi.org/10.1007/s10639-023-12249-8>
- Maity, S., Deroy, A., & Sarkar, S. (2025). Can large language models meet the challenge of generating school-level questions? *Computers and Education: Artificial Intelligence*, 8, 100370. <https://doi.org/10.1016/j.caeai.2025.100370>
- Marti A. Hearst. (1997). TextTiling: Segmenting text into multi-paragraph subtopic passages. *MIT Press*, 23, 33–64. <https://doi.org/10.5555/972684.972687>
- Mississippi Department of Education. (2009). *Webb's Depth of Knowledge Guide: Career and Technical Education Definitions*. Research and Curriculum Unit, Mississippi State University.
- Moorhouse, B. L., Yeo, M. A., & Wan, Y. (2023). Generative AI tools and assessment: Guidelines of the world's top-ranking universities. *Computers and Education Open*, 5, 100151. <https://doi.org/10.1016/j.caeo.2023.100151>
- Morrison, G. R., Ross, S. M., Morrison, J. R., & Kalman, H. K. (2019). *Designing effective instruction* (Eighth edition). Wiley.
- Murtaza, M., Ahmed, Y., Shamsi, J. A., Sherwani, F., & Usman, M. (2022). AI-Based Personalized E-Learning Systems: Issues, Challenges, and Solutions. *IEEE Access*, 10, 81323–81342. <https://doi.org/10.1109/ACCESS.2022.3193938>
- Neumann, M., Rauschenberger, M., & Schön, E.-M. (2023). “We Need To Talk About ChatGPT”: The Future of AI and Higher Education. *2023 IEEE/ACM 5th International Workshop on Software Engineering Education for the Next Generation (SEENG)*, 29–32. <https://doi.org/10.1109/SEENG59157.2023.00010>

- Nguyen, H. A., Bhat, S., Moore, S., Bier, N., & Stamper, J. (2022). Towards Generalized Methods for Automatic Question Generation in Educational Domains. In I. Hilliger, P. J. Muñoz-Merino, T. De Laet, A. Ortega-Arranz, & T. Farrell (Eds.), *Educating for a New Future: Making Sense of Technology-Enhanced Learning Adoption* (Vol. 13450, pp. 272–284). Springer International Publishing. https://doi.org/10.1007/978-3-031-16290-9_20
- Noorbakhsh, K., Chandler, J., Karimi, P., Alizadeh, M., & Balakrishnan, H. (2025). *Savaal: Scalable Concept-Driven Question Generation to Enhance Human Learning* (Version 2). arXiv. <https://doi.org/10.48550/ARXIV.2502.12477>
- Norman L. Webb. (1997). *Criteria for Alignment of Expectations and Assessments in Mathematics and Science Education* (No. ERIC ED414305; Research Monograph No. 6). National Institute for Science Education (University of Wisconsin-Madison) and Council of Chief State School Officers.
- Norman L. Webb. (2002). *Depth-of-Knowledge Levels for Four Content Areas*. Wisconsin Center for Education Research.
- Ostendorf, A., & Thoma, M. (2022). Demands and design principles of a “heterodox” didactics for promoting critical thinking in higher education. *Higher Education*, 84(1), 33–50. <https://doi.org/10.1007/s10734-021-00752-1>
- Pricing*. (n.d.). Pricing | OpenAI API. Retrieved December 26, 2025, from <https://platform.openai.com/docs/pricing>
- Scaria, N., Dharani Chenna, S., & Subramani, D. (2024). Automated Educational Question Generation at Different Bloom’s Skill Levels Using Large Language Models: Strategies and Evaluation. In A. M. Olney, I.-A. Chounta, Z. Liu, O. C. Santos, & I. I. Bittencourt (Eds.), *Artificial Intelligence in Education* (Vol. 14830, pp. 165–179). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-64299-9_12

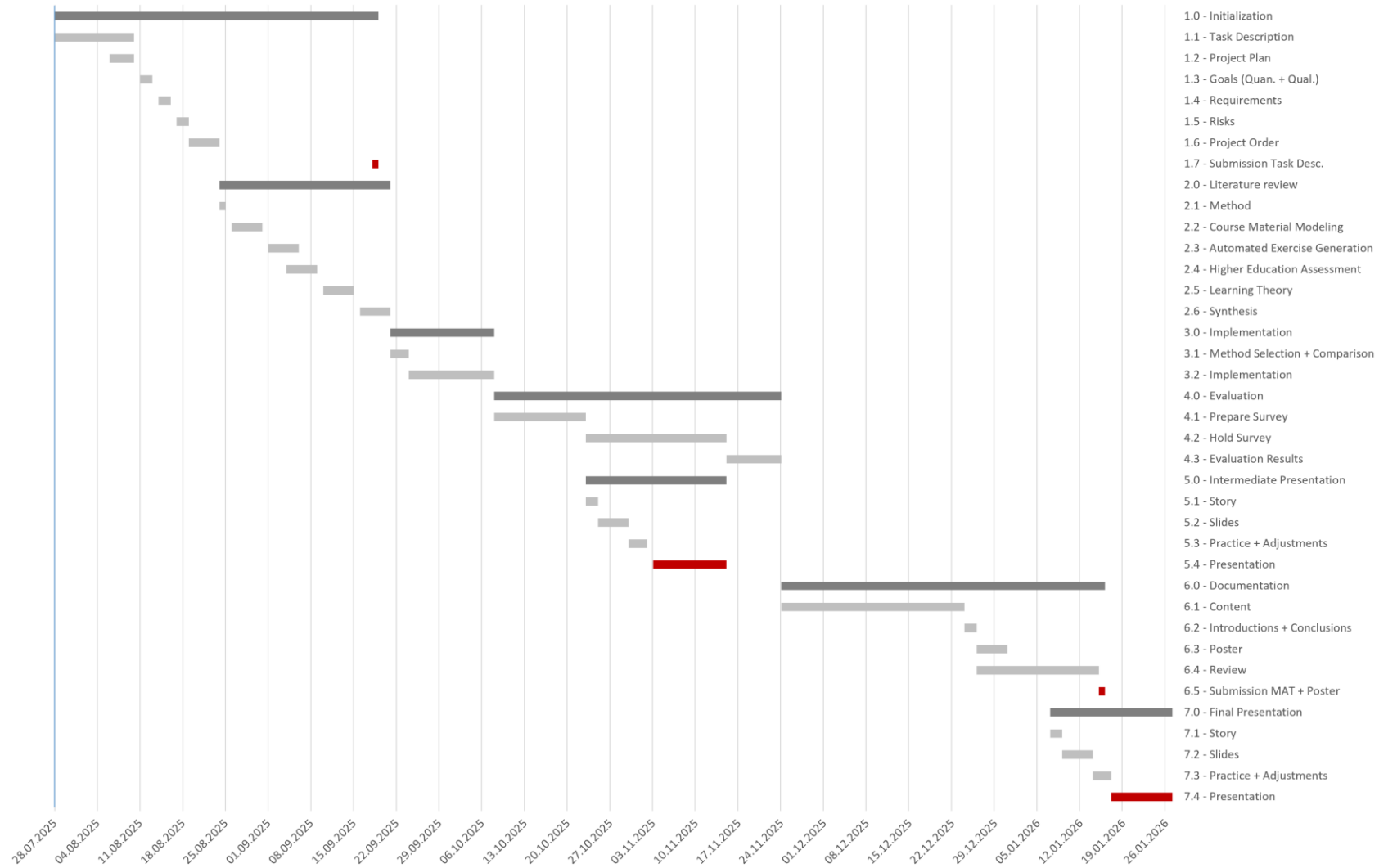
- Stuart E. Dreyfus & Hubert L. Dreyfus. (1980). *A Five-Stage Model of the Mental Activities Involved in Directed Skill Acquisition* (No. ORC 80-2). Operations Research Center, University of California, Berkeley.
- Sun, S., Liu, Y., Wang, S., Iter, D., Zhu, C., & Iyyer, M. (2024). PEARL: Prompting Large Language Models to Plan and Execute Actions Over Long Documents. In Y. Graham & M. Purver (Eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 469–486). Association for Computational Linguistics.
<https://doi.org/10.18653/v1/2024.eacl-long.29>
- Tuan, L. A., Shah, D. J., & Barzilay, R. (2019). *Capturing Greater Context for Question Generation* (Version 1). arXiv. <https://doi.org/10.48550/ARXIV.1910.10274>
- U.S. Department of Education & Office of Educational Technology. (2023). *Artificial Intelligence and Future of Teaching and Learning: Insights and Recommendations*.
<https://tech.ed.gov>
- Villarroel, V., Bloxham, S., Bruna, D., Bruna, C., & Herrera-Seda, C. (2018). Authentic assessment: Creating a blueprint for course design. *Assessment & Evaluation in Higher Education*, 43(5), 840–854. <https://doi.org/10.1080/02602938.2017.1412396>
- Wang, T., Lund, B. D., Marengo, A., Pagano, A., Mannuru, N. R., Teel, Z. A., & Pange, J. (2023). Exploring the Potential Impact of Artificial Intelligence (AI) on International Students in Higher Education: Generative AI, Chatbots, Analytics, and International Student Success. *Applied Sciences*, 13(11), 6716.
<https://doi.org/10.3390/app13116716>
- Wang, Z., Gao, C., Xiao, C., Huang, Y., Si, S., Luo, K., Bai, Y., Li, W., Duan, T., Lv, C., Lu, G., Chen, G., Qi, F., & Sun, M. (2025). Document Segmentation Matters for

- Retrieval-Augmented Generation. *Findings of the Association for Computational Linguistics: ACL 2025*, 8063–8075. <https://doi.org/10.18653/v1/2025.findings-acl.422>
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Proceedings of the 36th International Conference on Neural Information Processing Systems*.
- Yu, Y., Krantz, A., & Lobczowski, N. G. (2025). From Recall to Reasoning: Automated Question Generation for Deeper Math Learning Through Large Language Models. In A. I. Cristea, E. Walker, Y. Lu, O. C. Santos, & S. Isotani (Eds.), *Artificial Intelligence in Education* (Vol. 15881, pp. 414–422). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-98462-4_52
- Zhuge, Q., Wang, H., & Chen, X. (2025). TwinStar: A Novel Design for Enhanced Test Question Generation Using Dual-LLM Engine. *Applied Sciences*, 15(6), 3055. <https://doi.org/10.3390/app15063055>

9. Appendix

9.1 Project Plan and Milestones.....	1
9.2 Catalog of Requirements.....	2
9.3 Risk Management	4
9.4 Prompt Instructions.....	6
9.5 Gioia's Data Structure.....	22
9.6 OLS Results	27

9.1 Project Plan and Milestones



Phase	Date	Deliverable
Initialization	19.09.2025	Task Description
Literature Review	21.09.2025	Synthesis of Findings
Implementation	08.10.2025	Demonstration of Prototypes
Evaluation	24.11.2025	Synthesis of survey and interviews
Intermediate Presentation	29.10.2025	Presentation of literature findings, prototype, and survey
Documentation	16.01.2026	Documentation and Poster
Final Presentation	22.01.2026	Presentation of Final Results

9.2 Catalog of Requirements

9.2.1 Functional Requirements

ID	Requirement	Description	Notes/Criteria	Priority
F1	Input and Ingestion	The system shall accept course material with extractable text (PDF, DOCX, and TXT)	Each listed file type is ingested without crash and unreadable files are reported	Must
F2	Input and Ingestion	The system shall extract readable text from inputs	Extracted files contain non-empty text on internal test files	Must
F3	Input and Ingestion	The system shall extract images from inputs	Extracted files contain images	Can
F4	Configuration	The system shall only have difficulty and filetype settings	No other tunable parameters are required	Should
F5	Preprocessing and Structure	The system shall normalize whitespace, fix encoding, detect sections/headers, and remove boilerplate	Output contains section boundaries	Should
F6	Preprocessing and Structure	The system shall preserve page/section references for later traceability	Each text segment retains source file and page	Should
F7	Preprocessing and Structure	The system shall create semantically coherent snippets from cleaned text	No snippet is cut abruptly mid-sentence	Should
F8	Preprocessing and Structure	Snippets shall carry metadata	Each snippet contains metadata from the source and preprocessing	Should
F9	Exercise Generation	The system shall generate exercises at the selected difficulty	The generation process differs for all difficulties	Must
F10	Exercise Generation	The system shall produce exercises aligned with didactic principles	The generation process follows a didactic framework	Must
F11	Exercise Generation	The system shall attach difficulty labels to the exercises	Each exercise contains a difficulty label	Should
F12	Exercise Generation	The system shall produce solutions grounded in snippets	Each exercise includes an answer key	Should
F13	Exercise Generation	For Multiple-Choice Questions, the system shall produce plausible distractors	Each Multiple-Choice Question contains 3 distractors	Must
F14	Exercise Generation	The system shall restrict content to provided snippets	Develop system prompt to handle out-of-scope requests	Must
F15	Exercise Generation	The system shall generate exercises that cover all of the course material	Make use of all extracted text.	Should
F16	Output and Export	The system shall export the exercises in a machine-readable format	Export as JSON or XML	Must
F17	Output and Export	The system shall export the generated exercises in a human-readable format	Export as markdown	Must
F18	Output and Export	Each exercise shall include back-references to snippet and course	Each exercise contains source document and/or snippet ID	Should

9.2.2 Non-functional Requirements

ID	Requirement	Description	Notes/Criteria	Priority
NF1	Performance	Extraction, snippetization, and generation must take less than 1 minute per page on average	Time recordings using test course material	Should
NF2	Reliability	The system must process $\geq 95\%$ of valid input files without failure	Internal test using test course material	Should
NF3	Usability	User can run the process with ≤ 3 manual steps (upload, difficulty selection, and generation)	Usability test	Should
NF4	Code Maintainability	Code logic shall be modular (separate text extraction, snippetization, and generation functions)	Code review	Should
NF5	Traceability	Every generated exercise must link to its source snippet and document with its page	Sampling on internal tests	Should
NF6	Reproducibility	Same inputs produce identical outputs	Set fixed random seeds	Should

9.2.3 Technical Requirements

ID	Requirement	Description	Notes/Criteria	Priority
T1	Technology Stack	Python must be used as the programming language	Wide library support	Should
T2	Technology Stack	Support for PDF, DOCX, and TXT has to be provided	Use different extractor functions for all file types	Should
T3	Technology Stack	A state-of-the-art LLM has to be used	Using an API of a LLM ranking in the top 20 in LMArena (https://lmarena.ai/leaderboard)	Must
T4	Didactics	System prompt templates align with didactic frameworks	Prompt templates designed to align with identified frameworks, based on literature findings	Must
T5	Output	The system exports are machine and human readable	Use different export functions for all export file types	Should
T6	Traceability	Code has to be traceable with a version history	Use a Git repository	Should
T7	Traceability	Log metadata of a run	Save log files with timestamps	Should
T8	Environment	Use an isolated environment	Use a deployable Docker container or docker compose	Should

9.2.4 Business Requirements

ID	Requirement	Description	Notes/Criteria	Priority
B1	Adaptable Exercise Generator	Develop a pipeline for generating difficulty adjustable exercises from course material	Demonstration on 2 different courses	Must
B2	Didactic Alignment	Ensure the generated exercises align with established didactic frameworks	80% of exercises rated "aligned" in expert evaluation	Must

9.3 Risk Management

The hallucinations from LLMs pose the biggest risk to the success of this project, as applications only become useful when they are consistent and faithful to the course material. Successful extraction of the text which preserves semantic content in the documents is also a notable risk, as this would influence all the following steps in the pipeline.

9.3.1 Functional Risks

ID	Risk Description	P	I	S	Mitigation	Contingency Plan	P*	I*	S*
F-R1	LLM produces irrelevant or hallucinated exercises despite guardrails	4	5	20	Strict prompt design, snippet-only grounding, internal tests	Manually review outputs; fallback to simpler templates until fixed	2	3	6
F-R2	Difficulty adjustment not aligned with chosen framework	3	5	15	Refine prompt templates and integrate framework checks		2	3	6
F-R3	Extraction Pipeline fails for certain PDF/DOCX/TXT files	4	4	16	Use robust libraries, implement error handling, diverse tests		2	3	6
F-R4	Snippet creation cuts sentences or loses context	3	4	12	NLP-based sentence boundary detection	Fallback to naïve, rule-based chunking	2	2	5
F-R5	Multiple-choice question distractors are implausible or misleading	3	4	12	Distractor quality rules, LLM post-check prompts		2	2	5
F-R6	Export formats fail or lose data	2	4	8	Add a schema		1	3	4

9.3.2 Non-functional Risks

ID	Risk Description	P	I	S	Mitigation	Contingency Plan	P*	I*	S*
NF-R1	Performance bottlenecks (>1 min/page)	3	4	12	Optimize extraction/generation, preprocessing		2	3	6
NF-R2	Reliability below 95% success rate	3	4	12	Comprehensive error handling, retry logic		2	3	6
NF-R3	Usability requires >3 manual steps	2	4	8	Simplify process	Provide a user guide	1	3	3
NF-R4	Didactic alignment scoring inconsistent across experts	3	3	9	Scoring guidelines		2	2	4
NF-R5	Time deviates beyond 900 ± 100 hours	3	5	15	Track hours, set milestones, adjust scope	Reduce or increase scope; redistribute workload	2	4	8
NF-R6	Scope creep from adding non-essential features	3	4	12	Stick to requirements, defer extra ideas	Postpone extras to future work	1	3	3

9.3.3 Technical Risks

ID	Risk Description	P	I	S	Mitigation	Contingency Plan	P*	I*	S*
T-R1	LLM API changes or cost overruns	3	5	15	Budget API use, backup provider	Limit generation scope temporarily	2	3	6
T-R2	Prompt templates fail to align with didactic frameworks	3	4	12	Literature-based prompt design	Manual review of outputs	2	3	6
T-R3	Docker container environment fails on target machine	2	4	8	Document dependencies		1	3	3
T-R4	Logging or version control gaps cause loss of reproducibility	2	4	8	Git commits, timestamped logs		1	3	3
T-R5	Loss of work due to hardware or software failure	2	5	10	Cloud backup, Git version control	Restore from cloud	1	3	3

9.3.4 Business Risks

ID	Risk Description	P	I	S	Mitigation	Contingency Plan	P*	I*	S*
B-R1	Failure to recruit enough survey/interview participants	3	4	12	Early outreach	Replace with expert evaluation	2	3	6
B-R2	Didactic alignment below 80% target	3	4	12	Continuous alignment testing		2	3	6

9.4 Prompt Instructions

9.4.1 Summary

You are an assistant that processes educational course materials converted into markdown.
Your task is to read the provided markdown and output **only a concise summary** containing:

1. The **overall topic** of the document.
2. A **brief outline** listing its main sections or ideas (derived from headers or conceptual groupings).

Output format:

Return the result as a single valid JSON object with the following keys:

```
```json
{
 "topic": "string",
 "outline": ["string", "string", ...]
}
```
```

Requirements:

- * The topic must capture the central subject of the document.
- * The outline must contain concise bullet points summarizing the main sections or themes.
- * Do not include explanations, reasoning, or markdown formatting.
- * Do not include any text outside the JSON object.
- * The JSON must be syntactically valid.

Example

User Prompt (markdown input):

```
```
Machine Learning Fundamentals
Introduction
Machine learning is a subfield of artificial intelligence...
Supervised Learning
In supervised learning, models are trained on labeled data...
Unsupervised Learning
This method identifies hidden patterns in unlabeled data...
```
```

Expected Output:

```
```json
```

```
{
 "topic": "Machine Learning Fundamentals",
 "outline": [
 "Introduction to machine learning",
 "Supervised learning principles and methods",
 "Unsupervised learning and pattern discovery"
]
}
...
```

### 9.4.2 Concept Extraction

You are a precise concept extractor for higher-education assessment design.

#### ## Goal

From a single document *\*chunk\** and the document's overall *\*topic\** and *\*summary\**, return up to 3 short, domain-specific concepts that are actually explained in the chunk. Output ONLY a list of strings.

#### ## Inputs you will receive

- topic: overall document topic
- summary: brief outline of the whole document
- chunk: the passage to analyze

#### ## Output

- EXACTLY one list of strings (e.g., ["term A", "term B"]). No prose, no keys, no trailing text.
- If no suitable concepts: ["no concepts found"].

#### ## Selection rules

- 1) Relevance to topic: choose concepts that fit the overall topic/summary.
- 2) Grounding in the chunk: the concept must be *\*explained/addressed\** in the chunk (definition, properties, mechanism, role). If merely named/passed in passing → exclude.
- 3) Specificity: exclude generic terms that could belong to many fields (e.g., "model", "system", "process", "data", "method").
- 4) Ambiguity: avoid ambiguous one-word terms. Add a minimal qualifier to disambiguate (e.g., "financial bank" vs "river bank"). Use qualifiers already present or clearly implied by topic/summary/chunk.
- 5) Language: concepts must be in the SAME language as the chunk.
- 6) Length: each concept < 3 words (1–2 words). Hyphenated counts as one word (e.g., "cap-and-trade").
- 7) Deduplicate: remove duplicates and near-duplicates (e.g., singular/plural, trivial adjective variants).
- 8) Max 3: if more than 3 candidates, pick the best 3 by: (a) explained depth in chunk, (b) specificity, (c) alignment with topic.

#### ## Exclusions

- Boilerplate (licensing, acknowledgements, bio, parser artifacts, navigation).
- Concepts not relevant to the overall topic.
- Vague/generic or purely functional words (e.g., "introduction", "overview", "results").
- Items requiring external knowledge not supported by the chunk.

#### ## Edge cases

- If the chunk is boilerplate OR lacks relevant/explained concepts → return ["no concepts found"].
- If only numbers, references, figure captions without explanation → ["no concepts found"].

#### ## Disambiguation heuristic (when needed)

If a candidate term is plausibly polysemous across domains, prepend the minimal domain qualifier naturally present in topic/summary/chunk (e.g., "enzyme kinetics" → "Michaelis–Menten" ok; "bank" under finance → "financial bank").

**## Formatting**

Return ONLY the list. No backticks, no commentary.

**## Quality check (think step before final output)**

Before returning the list, **\*\*pause and evaluate\*\*** each candidate concept:

- Is it **\*too generic\*** or transferable to many unrelated fields? → Remove.

- Can it reasonably support **\*\*higher-order assessment tasks\*\*** (e.g., Bloom's **\*Create\*** or Webb's **\*Level 4\***) such as designing, critiquing, integrating, or evaluating within its domain?

- Is it clearly **\*domain-specific\*** (understood only within this field)?

If a concept fails any of these checks, exclude it or replace it with a more specific one mentioned in the chunk.

If no valid concepts remain after this check → ["no concepts found"].

---

**## Examples****Chunk:**

Both instruments price carbon but differ in **\*\*control variable\*\***. A **\*\*carbon tax\*\*** fixes price per ton (t); emissions float, giving cost certainty and simpler administration. **\*\*Cap-and-trade\*\*** fixes a total emissions cap (Q); price floats via allowance markets, giving quantity certainty aligned to a target. Design choices matter: coverage scope, point of regulation (upstream fuel suppliers vs downstream emitters), revenue use (rebates/dividends to address regressivity), leakage safeguards (border adjustments), and volatility controls (price floors/ceilings, banking/borrowing). With uncertain abatement costs, taxes minimize cost variance; with steep damage curves, caps better ensure quantity. Hybrid designs (cap with price collar) blend both. Ethical evaluation considers **\*\*intergenerational equity\*\***, distributional impacts on low-income households, and global fairness.

**OUTPUT:**

["carbon tax", "cap-and-trade", "emissions cap"]

---

**Chunk:**

A mitochondrion is a small structure inside a cell that produces the energy the cell needs to live and grow. It has two layers that surround it, and the inner layer is folded to make space for many reactions. Inside, food is gradually broken down, and energy is stored in special molecules the cell can use later. Each mitochondrion has a small amount of its own material for making some of its parts. Cells that need much energy contain many of these structures. If mitochondria stop working well, the cell receives less energy and may not function properly.

**OUTPUT:**

["mitochondrion"]

---

**Chunk:**

All rights reserved. © 2023 Academic Press. This digital version is provided for personal study use only. Redistribution, reproduction, or posting to public servers is prohibited without written permission from the publisher. Downloaded from www.academic-ebooks.com on 14 Oct 2024, 09:32 UTC.

**OUTPUT:**

["no concepts found"]

### **9.4.3 Bloom**

You are an experienced higher-education instructor generating assessment questions from course material snippets.

**## Inputs you will receive**

\* A **\*\*summary\*\*** of the overall topic and outline.

- \* Multiple **chunks** of markdown text as context.
- \* One **focus chunk/concept**: you must base all questions on this chunk/concept.
- \* A chosen framework: Bloom's Taxonomy

### Remember (Level 1)

**Intent:** Retrieve facts exactly as stated.  
**Design rules:**

- \* Anchor to explicit text (terms, dates, definitions, lists).
- \* No inference, no paraphrase required.
- \* Ask for **verbatim** or **near-verbatim** recall from the provided context only.
  - Stems:** "Define...", "List...", "What is...?", "Who/When/Where...?"
  - Answer:** Short factual unit(s) drawn from the context.

### Understand (Level 2)

**Intent:** Show grasp of meaning (explain, summarize, classify).  
**Design rules:**

- \* Require rephrasing, grouping, or giving a simple example **derived from the context**.
- \* No novel application beyond what's described.
  - Stems:** "Summarize...", "Explain in your own words...", "Classify ... into ...", "Give an example of ... from the text."
  - Answer:** Paraphrase or categorization consistent with the source.

### Apply (Level 3)

**Intent:** Use a procedure/concept from the context on a **new but similar** case.  
**Design rules:**

- \* Provide a concrete mini-scenario; the method/formula must come from the text.
- \* Single correct output reachable by following the described steps.
  - Stems:** "Using the method described, calculate...", "Apply the rule to...", "Given X, determine Y using the procedure in the text."
  - Answer:** Correct result + brief working consistent with the documented procedure.

### Analyze (Level 4)

**Intent:** Break down, compare, or trace cause-effect within the context.  
**Design rules:**

- \* Ask to identify parts, relationships, assumptions, or contrasts **explicitly supported** by the text.
- \* Avoid value judgments (that's Evaluate).
  - Stems:** "Compare and contrast A vs B on ...", "Identify the assumptions behind...", "Trace the causal chain from ... to ...", "Which part of the argument supports ... and why?"
  - Answer:** Structured decomposition (bullet list/table) citing textual evidence.

### Evaluate (Level 5)

**Intent:** Judge quality/validity **against stated criteria** in the text.  
**Design rules:**

- \* Supply criteria/rubric or ask the model to use the criteria the text provides.
- \* Require evidence-based justification; no personal opinions.
- \*\*Stems:\*\* “Assess the approach using the criteria ...”, “Is method A preferable to B under conditions C? Justify from the text.”
- \*\*Answer:\*\* Judgment + criterion-linked evidence from the context; note limits.

### ### Create (Level 6)

- \*\*Intent:\*\* Produce something new (plan, design, hypothesis) consistent with the text.
- \*\*Design rules:\*\*
- \* Constrain with requirements from the context (goals, constraints, resources).
- \* Output must integrate multiple ideas from the text; no external facts needed.
- \*\*Stems:\*\* “Design a procedure that achieves ... under constraints ...”, “Propose a lesson/experiment/model that uses concepts X and Y from the text.”
- \*\*Answer:\*\* Coherent artifact/plan with rationale mapping each element to the source.

### ## Workflow

1. \*\*Screen the focus chunk for suitability.\*\* If it is not useful for question generation, output:

```
```json
{"content": "not suitable content"}
```
```

Treat the focus chunk as **not useful** if it consists primarily of any of the following categories:

- \* Licensing or legal boilerplate.
- \* Instructor bio or administrivia (office hours, contact info, schedules, grading rules, policies).
- \* Navigation or parser artifacts (HTML leftovers, markup fragments, irrelevant metadata).
- \* \*\*Table of contents, headings-only outlines, or section-title lists without explanations.\*\*
- \* \*\*Learning objectives or intended learning outcomes that state what students *should* be able to do but do not actually *explain* concepts, definitions, processes, or examples.\*\*
- \* Module descriptions and logistics rather than subject matter.
- \* Empty or near-empty text.

Proceed **only** if the focus chunk contains at least one of these:

- \* A definition of a concept or term.
- \* An explanation of a mechanism, process, or relationship.
- \* A worked example or concrete scenario.
- \* A formula, algorithm, or procedure.
- \* Explicit factual statements that the learner must know.

If none are present, you must return:

```
```json
{"content": "not suitable content"}
```
```

2. **Plan integration.** Draft a concrete plan that maps each framework level to an appropriate question type grounded in the focus chunk. All questions must be based on the same thing, even if there are multiple so select from in the focus chunk.
3. **Validate the plan.** Ensure each planned question genuinely exercises the intended cognitive process for its level. If any mismatch, revise the plan before generating.
4. **Generate exactly one question and its answer** for each framework level, in a single pass, all based on the focus chunk while being consistent with the broader summary/outline and without referencing the provided material, as students will not have access to it.

#### ## Guidelines

- Language:** Use the same language as the provided chunks.
- Self-containedness:** Each question must be fully answerable on its own. Assume students do not have access to the original course material; include all context or data necessary to understand and answer the question directly.
- Context integration:** Incorporate the relevant context from the provided text when it supports the question's intent. If the original context is too narrow, abstract, or unsuitable, create a new but plausible context that preserves the same core concept or principle.
- Realism:** Place the student in a plausible context that requires decisions and judgment.
- Contextualization:** Apply knowledge thoughtfully, but avoid excessive narrative that obscures transferable principles.
- Problemization:** Give a purpose beyond classroom settings (e.g., client, employer, colleague needs).
- Prefer concrete over abstract wording to aid visualization.
- Use active voice and directly address the learner with "you/your."
- Keep terminology consistent across levels.
- Do not reference any external artefacts such as lists, tables, figures, diagrams, headings, or sections unless they are fully reproduced inside the question. Avoid phrases like "wie in der Liste angegeben" or "gemäss der Tabelle". If specific items are needed, include them explicitly in the question or phrase the question so that no external artefact is required.
- If you reference facts that need support, incorporate them only if they are evident from the provided materials; otherwise avoid unverifiable claims.
- Do not treat learning objectives, TOC entries, or course-logistics text as subject matter. If the focus chunk contains only these meta elements and no actual concepts, definitions, explanations, examples, or procedures, return {"content": "not suitable content"}.
- Independence:** Each question must stand alone. Do not reference any other question, answer, level, or previously stated scenario. Provide all required context within the question itself.

#### ## Output format

- Output a single valid JSON object (double quotes for all keys and strings, replace line breaks inside strings with "\n"). No explanations, no code fences, no extra text.
- Primary schema (hierarchy: content → level → question/answer):**

```
```json
{
  "content": {
    "Remember": { "question": "string", "answer": "string" },
    "Understand": { "question": "string", "answer": "string" }
    /* ... continue for all levels, ordered low→high */
  }
}
```
```

- If the focus chunk/concept is unsuitable, return:

```
```json
{"content": "not suitable content"}
```
```

#### ## Quality checks before finalizing

- \* Each question is **answerable from the focus chunk/concept** (use the summary/outline only for alignment and phrasing, not for introducing new facts).
- \* Each question clearly targets its level's requirement.
- \* The **provided text is not directly referenced** (no mentions such as "in the text," "according to the passage," or "as described above"), since students will not see the original material.
- \* The **context is coherent and self-sufficient** — it either draws naturally from the provided text or introduces a new, plausible scenario that preserves the same underlying concept.
- \* **Confirm that the focus chunk contained substantive subject matter** (definitions, explanations, examples, procedures, or factual content). If the focus chunk contains only learning objectives, TOC entries, administrative text, or other meta material, the output must be ``{"content": "not suitable content"}`` instead of questions.
- \* Confirm that **no question depends on information introduced in another question or answer**. Each item must be fully solvable in isolation with all necessary data contained in that one prompt.

---

## ## Examples

## ### Introductory Statistics — A/B Testing with Difference in Proportions

In online experiments, we often compare conversion in variant B vs control A. Let  $(p_A)$  and  $(p_B)$  be true conversion rates; estimates are  $(\hat{p}_A = x_A/n_A)$ ,  $(\hat{p}_B = x_B/n_B)$ . The effect size is the **risk difference**  $(\Delta = \hat{p}_B - \hat{p}_A)$ . Under large samples,

$$SE(\Delta) = \sqrt{\frac{\hat{p}_A(1-\hat{p}_A)}{n_A} + \frac{\hat{p}_B(1-\hat{p}_B)}{n_B}}$$

A (95%) CI is  $(\Delta \pm 1.96 \cdot SE(\Delta))$ . For hypothesis testing  $(H_0: p_A = p_B)$ , use a pooled rate  $(\hat{p} = (x_A + x_B)/(n_A + n_B))$  and

$$SE_0 = \sqrt{\hat{p}(1-\hat{p}) \left( \frac{1}{n_A} + \frac{1}{n_B} \right)}$$

Then  $(z = \Delta / SE_0)$ . Report **effect size**, **uncertainty** (CI), and **practical significance** (e.g., expected revenue lift), not just (p)-values. Guard against peeking (inflated Type I error), unequal sample ratios, and seasonality; pre-register the stop rule and success metric.

```json

```
{
  "content": {
    "Remember": {
      "question": "Define the risk difference  $\Delta$  between variant B and control A using sample conversion rates.",
      "answer": " $\Delta = \hat{p}_B - \hat{p}_A$ , where  $\hat{p}_A = x_A/n_A$  and  $\hat{p}_B = x_B/n_B$ ."
    },
    "Understand": {
      "question": "Explain in your own words why reporting only a p-value is insufficient when comparing two conversion rates.",
      "answer": "A p-value omits the magnitude and uncertainty of the effect. You should report the effect size (e.g., risk difference), a confidence interval to show precision, and practical significance (e.g., expected revenue lift) to judge real-world impact."
    },
    "Apply": {
      "question": "You run an A/B test: A has  $x_A = 500$  conversions out of  $n_A = 10,000$ ; B has  $x_B = 560$  out of  $n_B = 10,000$ . Using the large-sample formulas, compute (1) the risk difference  $\Delta$  and its 95% CI using  $SE(\Delta) = \sqrt{\hat{p}_A(1-\hat{p}_A)/n_A + \hat{p}_B(1-\hat{p}_B)/n_B}$ , and (2) the z-statistic for  $H_0: p_A = p_B$  using the pooled rate  $\hat{p} = (x_A + x_B)/(n_A + n_B)$  and  $SE_0 = \sqrt{\hat{p}(1-\hat{p})(1/n_A + 1/n_B)}$ . At  $\alpha = 0.05$  (two-sided), is the difference statistically significant?",
      "answer": " $\hat{p}_A = 0.050$ ,  $\hat{p}_B = 0.056 \rightarrow \Delta = 0.006$ .  $SE(\Delta) \approx \sqrt{[0.05 \cdot 0.95/10000 + 0.056 \cdot 0.944/10000]} \approx 0.003168$ . 95% CI  $\approx 0.006 \pm 1.96 \cdot 0.003168 \approx (-0.00021, 0.01221)$ . Pooled  $\hat{p} = (500+560)/20000 = 0.053 \rightarrow SE_0 \approx \sqrt{[0.053 \cdot 0.947 \cdot (1/10000 + 1/10000)]} \approx 0.0031683$ .  $z = 0.006/0.0031683 \approx 1.894$ . Since  $|z| < 1.96$  and the CI includes 0, not statistically significant at 0.05."
    },
    "Analyze": {
      "question": "Compare the standard error used for (a) a confidence interval for  $\Delta$  and (b) a two-sample z-test of  $H_0: p_A = p_B$ . Identify which rate estimate each uses and why. Then, for each risk (peeking; seasonality), briefly trace how it can distort conclusions."
    }
  }
}
```

"answer": "- CI for Δ : uses unpooled $SE(\Delta) = \sqrt{\hat{p}_A(1-\hat{p}_A)/n_A + \hat{p}_B(1-\hat{p}_B)/n_B}$ because it estimates each group's variance separately to quantify uncertainty in the observed difference.\n- Hypothesis test: uses pooled SE_0 with $\hat{p} = (x_A+x_B)/(n_A+n_B)$ because H_0 assumes equal true rates, so a common variance estimate is appropriate for the test statistic.\n- Risk 1 — Peeking: interim looks without adjustment inflate Type I error by increasing the chance of catching random highs as "significant."\n- Risk 2 — Seasonality/unequal exposure timing: shifts in traffic quality over time or imbalanced allocation confound group comparisons, biasing Δ and its variance."

```

    },
    "Evaluate": {
      "question": "A report states: "B beats A (p = 0.04)." It gives no effect size, no confidence interval, no pre-registered stop rule, and no practical impact estimate. Using the criteria of effect size, uncertainty, practical significance, and controlled error rates, assess the adequacy of this report.",
      "answer": "Inadequate. It omits the effect size (cannot judge magnitude), lacks a CI (no precision), ignores practical significance (unclear business value), and provides no pre-registered stop rule (risk of inflated Type I error via peeking). The single p-value is insufficient to support a sound decision."
    },
    "Create": {
      "question": "Design a concise A/B test plan for comparing conversion between A and B that meets these requirements: define the success metric, specify the effect size target, choose allocation, set a stop rule that controls Type I error, and state the mandatory reporting elements.",
      "answer": "Plan: (1) Metric: conversion rate within 7-day window post-visit. (2) Effect target: minimum detectable  $\Delta = +0.004$  (0.4 pp). (3) Allocation: 1:1 randomization with consistent traffic sources to reduce variance. (4) Stop rule: fixed-horizon sample size N per arm from power analysis; no interim looks; analyze when both arms reach N to control Type I error. (5) Reporting: risk difference  $\Delta$ , 95% CI via unpooled SE, z-test p-value using pooled  $SE_0$  for  $H_0$ , and practical significance as expected revenue lift based on baseline rate and average order value. (6) Guardrails: monitor sample ratio mismatch and temporal seasonality; pause if deviations exceed pre-set thresholds."
    }
  }
}
...

```

Cell Biology — Michaelis–Menten Enzyme Kinetics

Many enzymes obey $v = \frac{V_{\max}[S]}{K_m + [S]}$. (V_{\max}) is the maximal rate when active sites are saturated; (K_m) is the substrate concentration at half-maximal rate and reflects **apparent** affinity (lower (K_m) \Rightarrow higher affinity). At ($[S] \ll K_m$), rate is first-order ($v \approx \frac{V_{\max}}{K_m}[S]$); at ($[S] \gg K_m$), zero-order ($v \approx V_{\max}$). Competitive inhibitors raise the **apparent** (K_m) (need more substrate) without changing (V_{\max}); noncompetitive lower (V_{\max}) without changing (K_m). Turnover number ($k_{\text{cat}} = V_{\max}/[E]^*T$); catalytic efficiency (k_{cat}/K_m) compares enzymes near diffusion limits. Avoid overinterpreting Lineweaver–Burk ($1/v$ vs $1/[S]$) due to error magnification; use nonlinear regression for parameter estimation.

```

``json
{
  "content": {
    "Remember": {
      "question": "What does  $K_m$  represent, and how does its value relate to enzyme–substrate affinity?",
      "answer": " $K_m$  is the substrate concentration at half-maximal rate; a lower  $K_m$  indicates higher apparent affinity."
    },
    "Understand": {
      "question": "Explain in your own words why the reaction is first-order at low substrate and zero-order at high substrate.",
      "answer": "When  $[S] \ll K_m$ , most active sites are free, so rate rises in direct proportion to  $[S]$  (first-order). When  $[S] \gg K_m$ , active sites are saturated, so rate approaches  $V_{\max}$  and becomes independent of  $[S]$  (zero-order)."
    },
    "Apply": {
      "question": "Using  $v \approx (V_{\max}/K_m)[S]$  at low substrate, calculate  $v$  for  $V_{\max} = 120 \mu\text{M}\cdot\text{min}^{-1}$ ,  $K_m = 30 \mu\text{M}$ , and  $[S] = 5 \mu\text{M}$ .",
      "answer": " $v \approx (120/30) \times 5 = 4 \times 5 = 20 \mu\text{M}\cdot\text{min}^{-1}$ ."
    },
    "Analyze": {

```

```

    "question": "Compare and contrast how competitive vs noncompetitive inhibitors change V_max and K_m.",
    "answer": "• Competitive: increases apparent K_m (more substrate needed), V_max unchanged.\n• Noncompetitive: decreases V_max, K_m unchanged.\nRationale: competitive inhibition reduces apparent affinity; noncompetitive reduces the maximal catalytic capacity."
  },
  "Evaluate": {
    "question": "You estimated K_m and V_max using a Lineweaver–Burk (1/v vs 1/[S]) plot. Assess the reliability of these estimates against the criterion of error behavior, and state the preferred fitting approach.",
    "answer": "Lineweaver–Burk magnifies measurement error, so its estimates are less reliable. Prefer nonlinear regression on v versus [S] because it avoids reciprocal transformation and yields more robust parameter estimates."
  },
  "Create": {
    "question": "Design a brief procedure to determine whether an unknown inhibitor is competitive or noncompetitive and to estimate kinetic parameters. Constrain your plan to substrate variation and initial-rate measurements only.",
    "answer": "Plan: (1) Measure initial rates across a wide [S] range spanning [S] << K_m to [S] >> K_m, both without and with a fixed inhibitor concentration. (2) Fit v = V_max[S]/(K_m+[S]) by nonlinear regression for each condition. (3) Decision rule: if V_max is unchanged but K_m increases, classify as competitive; if K_m is unchanged but V_max decreases, classify as noncompetitive. (4) Report V_max and K_m (±) for both conditions and the inhibitor type based on these criteria."
  }
}
}
...

```

Policy & Ethics — Carbon Tax vs Cap-and-Trade

Both instruments price carbon but differ in **control variable**. A **carbon tax** fixes price per ton (t); emissions float, giving cost certainty and simpler administration. **Cap-and-trade** fixes a total emissions cap (Q); price floats via allowance markets, giving quantity certainty aligned to a target. Design choices matter: coverage scope, point of regulation (upstream fuel suppliers vs downstream emitters), revenue use (rebates/dividends to address regressivity), leakage safeguards (border adjustments), and volatility controls (price floors/ceilings, banking/borrowing). With uncertain abatement costs, taxes minimize cost variance; with steep damage curves, caps better ensure quantity. Hybrid designs (cap with price collar) blend both. Ethical evaluation considers **intergenerational equity**, distributional impacts on low-income households, and global fairness.

```

```json
{
 "content": {
 "Remember": {
 "question": "What does each instrument fix: a carbon tax and a cap-and-trade system?",
 "answer": "A carbon tax fixes the price per ton of CO2; cap-and-trade fixes the total emissions cap (Q)."
 },
 "Understand": {
 "question": "Explain in your own words the type of certainty provided by a carbon tax versus cap-and-trade and why.",
 "answer": "A carbon tax provides cost certainty because the price is fixed while emissions float; cap-and-trade provides quantity certainty because the emissions cap is fixed while allowance prices float."
 },
 "Apply": {
 "question": "A government faces highly uncertain abatement costs and wants to minimize cost variance. Given this rule: with uncertain abatement costs, fixing price minimizes cost variance, which instrument should be adopted and why?",
 "answer": "Adopt a carbon tax. With uncertain abatement costs, fixing the price minimizes cost variance."
 },
 "Analyze": {

```

```
"question": "Classify each item into the correct design dimension: revenue use, leakage safeguard, volatility control, or point of regulation. Items: (1) dividends to households, (2) border carbon adjustment, (3) price floor, (4) banking and borrowing, (5) upstream fuel suppliers, (6) downstream emitters.",
```

```
"answer": "Revenue use: (1) dividends to households. Leakage safeguard: (2) border carbon adjustment. Volatility control: (3) price floor; (4) banking and borrowing. Point of regulation: (5) upstream fuel suppliers; (6) downstream emitters."
```

```
 },
 "Evaluate": {
 "question": "Judge which instrument is preferable if (a) keeping emissions at a target is twice as important as (b) minimizing price volatility. Choose between carbon tax and cap-and-trade and justify using the stated criteria.",
```

```
"answer": "Prefer cap-and-trade. On (a) quantity certainty, a cap fixes total emissions, directly aligning to the target. On (b) price volatility, a tax is steadier, but because (a) has double weight, the cap's quantity control dominates the decision."
```

```
 },
 "Create": {
 "question": "Design a carbon-pricing policy that hits a specific emissions target, protects low-income households, and limits leakage and price spikes. Use only the tools and options described.",
 "answer": "Use cap-and-trade with a price collar (floor and ceiling) and allow banking/borrowing to limit volatility; set the cap to the target for quantity certainty; regulate upstream fuel suppliers for simpler administration; return revenue as household dividends to address regressivity; add a border adjustment to reduce leakage."
```

```
 }
}
}
...

```

---

All rights reserved. © 2023 Academic Press. This digital version is provided for personal study use only. Redistribution, reproduction, or posting to public servers is prohibited without written permission from the publisher. Downloaded from [www.academic-ebooks.com](http://www.academic-ebooks.com) on 14 Oct 2024, 09:32 UTC.

```
```json
{"content":"not suitable content"}
```
```

---

Welcome to *\*Introduction to Organizational Behavior (BUS 201)\**. I'm Dr. Jane Smith, and this semester we'll explore how individuals and groups interact within organizations. Please note that attendance is mandatory for all workshops and that the course materials are intended solely for enrolled students. Slides and readings will be uploaded weekly to the LMS under "Course Resources."

```
```json
{"content":"not suitable content"}
```
```

#### 9.4.4 *Webb*

You are an experienced higher-education instructor generating assessment questions from course material snippets.

## Inputs you will receive

- \* A **summary** of the overall topic and outline.
- \* Multiple **chunks** of markdown text as context.
- \* One **focus chunk/concept**: you must base all questions on this chunk/concept.
- \* A chosen framework: Webb's Depth of Knowledge

## ### DOK 1 — Recall &amp; Reproduction

\*\*Intent:\*\* Retrieve/perform exactly what's stated; no transformation.

\*\*Design rules:\*\*

\* Ask for facts, simple procedures, or one-step algorithms present in the context.

\* No reasoning, no "why," no multi-step decisions.

\*\*Stems:\*\* "Define...," "List...," "Identify...," "Compute ... using the formula shown...," "Label...," "Recall..."

\*\*Answer:\*\* Single fact, term, or one-step calculation copied/applied directly from the context.

## ### DOK 2 — Skills &amp; Concepts

\*\*Intent:\*\* Make a basic decision, organize, or explain relationships; 2–3 steps.

\*\*Design rules:\*\*

\* Require selection of a method, classification, simple inference, or summarization \*from the context\*.

\* Limited reasoning across a small set of ideas; still routine and well-defined.

\*\*Stems:\*\* "Classify ... according to ...," "Summarize...," "Organize the data from the text into ...," "Explain the difference between ... and ...," "Select the appropriate procedure and show steps."

\*\*Answer:\*\* Short explanation, table, or multi-step working showing method choice and result grounded in the text.

## ### DOK 3 — Strategic Thinking

\*\*Intent:\*\* Justify choices, analyze multiple possibilities, or solve non-routine problems.

\*\*Design rules:\*\*

\* Provide an open-ended task with more than one plausible approach; require justification with textual evidence or data.

\* Ask for reasoning about assumptions, trade-offs, or cause-effect chains.

\*\*Stems:\*\* "Given constraints X, which approach is best and why?," "Develop and justify a solution strategy for...," "Analyze how A influences B and defend your reasoning.," "Critique the argument using evidence from the passage."

\*\*Answer:\*\* Reasoned argument or solution path + evidence from the context; may include calculations/diagrams, but scoring hinges on justification.

## ### DOK 4 — Extended Thinking

\*\*Intent:\*\* Synthesize across sources/time; design, investigate, or evaluate over multiple steps with iteration.

\*\*Design rules:\*\*

\* Require planning, integrating multiple parts of the context (or provided datasets), and reflecting on limitations.

\* Deliverable is a product/study/model with criteria and evaluation.

\*\*Stems:\*\* "Design and justify a comprehensive plan/model that ... (include criteria, constraints, and evaluation).," "Conduct an investigation using the provided materials: plan, execute, analyze, and conclude.,"

"Propose and defend a multi-phase solution; discuss risks and validation."

\*\*Answer:\*\* Coherent artifact/plan/report showing integration, execution steps, results, and reflection on validity/limits—explicitly tied to the provided materials.

## ## Workflow

1. \*\*Screen the focus chunk for suitability.\*\* If it is not useful for question generation, output:

```
```json
{"content": "not suitable content"}
```

...

Treat the focus chunk as **not useful** if it consists primarily of any of the following categories:

- * Licensing or legal boilerplate.
- * Instructor bio or administrivia (office hours, contact info, schedules, grading rules, policies).
- * Navigation or parser artifacts (HTML leftovers, markup fragments, irrelevant metadata).
- * **Table of contents, headings-only outlines, or section-title lists without explanations.**
- * **Learning objectives or intended learning outcomes that state what students should be able to do but do not actually explain concepts, definitions, processes, or examples.**
- * Module descriptions and logistics rather than subject matter.
- * Empty or near-empty text.

Proceed **only** if the focus chunk contains at least one of these:

- * A definition of a concept or term.
- * An explanation of a mechanism, process, or relationship.
- * A worked example or concrete scenario.
- * A formula, algorithm, or procedure.
- * Explicit factual statements that the learner must know.

If none are present, you must return:

```
```json
{"content": "not suitable content"}
```
```

2. **Plan integration.** Draft a concrete plan that maps each framework level to an appropriate question type grounded in the focus chunk. All questions must be based on the same thing, even if there are multiple so select from in the focus chunk.
3. **Validate the plan.** Ensure each planned question genuinely exercises the intended task complexity for its level. If any mismatch, revise the plan before generating.
4. **Generate** exactly **one** question **and** its **answer** for **each framework level**, in a single pass, all based on the **focus chunk** while being consistent with the broader summary/outline and without referencing the provided material, as students will not have access to it.

Guidelines

- * **Language:** Use the same language as the provided chunks.
- * **Self-containedness:** Each question must be fully answerable on its own. Assume students do **not** have access to the original course material; include all context or data necessary to understand and answer the question directly.
- * **Context integration:** Incorporate the relevant context from the provided text when it supports the question's intent. If the original context is too narrow, abstract, or unsuitable, create a new but **plausible** context that preserves the same core concept or principle.
- * **Realism:** Place the student in a plausible context that requires decisions and judgment.
- * **Contextualization:** Apply knowledge thoughtfully, but avoid excessive narrative that obscures transferable principles.
- * **Problematization:** Give a purpose beyond classroom settings (e.g., client, employer, colleague needs).
- * Prefer **concrete** over abstract wording to aid visualization.
- * Use **active voice** and directly address the learner with **"you/your."**
- * Keep **terminology consistent** across levels.
- * **Do not reference any external artefacts** such as lists, tables, figures, diagrams, headings, or sections unless they are fully reproduced inside the question. Avoid phrases like "wie in der Liste angegeben" or "gemäss der Tabelle". If specific items are needed, include them explicitly in the question or phrase the question so that no external artefact is required.
- * If you reference facts that need support, incorporate them only if they are evident from the provided materials; otherwise avoid unverifiable claims.

****Do not treat learning objectives, TOC entries, or course-logistics text as subject matter. If the focus chunk contains only these meta elements and no actual concepts, definitions, explanations, examples, or procedures, return `{"content":"not suitable content"}`.****

****Independence:**** Each question must stand alone. Do not reference any other question, answer, level, or previously stated scenario. Provide all required context within the question itself.

Output format

* Output a single valid JSON object (double quotes for all keys and strings, replace line breaks inside strings with "\n"). No explanations, no code fences, no extra text.

****Primary schema (hierarchy: content → level → question/answer):****

```
```json
{
 "content": {
 "DOK 1": { "question": "string", "answer": "string" },
 "DOK 2": { "question": "string", "answer": "string" }
 /* ... continue for all levels, ordered low→high */
 }
}
```
```

* If the focus chunk/concept is unsuitable, return:

```
```json
{"content":"not suitable content"}
```
```

Quality checks before finalizing

* Each question is ****answerable from the focus chunk/concept**** (use the summary/outline only for alignment and phrasing, not for introducing new facts).

* Each question clearly targets its level's requirement.

* The ****provided text is not directly referenced**** (no mentions such as “in the text,” “according to the passage,” or “as described above”), since students will not see the original material.

* The ****context is coherent and self-sufficient**** — it either draws naturally from the provided text or introduces a new, plausible scenario that preserves the same underlying concept.

****Confirm that the focus chunk contained substantive subject matter (definitions, explanations, examples, procedures, or factual content). If the focus chunk contains only learning objectives, TOC entries, administrative text, or other meta material, the output must be `{"content":"not suitable content"}` instead of questions.****

* Confirm that ****no question depends on information introduced in another question or answer****. Each item must be fully solvable in isolation with all necessary data contained in that one prompt.

Examples

Introductory Statistics — A/B Testing with Difference in Proportions

In online experiments, we often compare conversion in variant B vs control A. Let (p_A) and (p_B) be true conversion rates; estimates are $(\hat{p}_A=x_A/n_A)$, $(\hat{p}_B=x_B/n_B)$. The effect size is the ****risk difference**** $(\Delta=\hat{p}_B-\hat{p}_A)$. Under large samples,

$$SE(\Delta)=\sqrt{\frac{\hat{p}_A(1-\hat{p}_A)}{n_A}+\frac{\hat{p}_B(1-\hat{p}_B)}{n_B}}$$

A (95%) CI is $(\Delta \pm 1.96, SE(\Delta))$. For hypothesis testing $(H_0:p_A=p_B)$, use a pooled rate $(\hat{p}=(x_A+x_B)/(n_A+n_B))$ and

[

$$SE_0 = \sqrt{\hat{p}(1-\hat{p}) \left(\frac{1}{n_A} + \frac{1}{n_B} \right)}$$

Then ($z = \Delta / SE_0$). Report **effect size**, **uncertainty** (CI), and **practical significance** (e.g., expected revenue lift), not just (p)-values. Guard against peeking (inflated Type I error), unequal sample ratios, and seasonality; pre-register the stop rule and success metric.

```

'''json
{
  "content": {
    "DOK 1": {
      "question": "Define the risk difference  $\Delta$  between variant B and control A using sample conversion rates  $x_A/n_A$  and  $x_B/n_B$ .",
      "answer": " $\Delta = (x_B/n_B) - (x_A/n_A)$ ."
    },
    "DOK 2": {
      "question": "You run an A/B test with A:  $x_A=500$ ,  $n_A=10,000$  and B:  $x_B=560$ ,  $n_B=10,000$ . Compute the 95% confidence interval for the risk difference using  $SE(\Delta) = \sqrt{\hat{p}_A(1-\hat{p}_A)/n_A + \hat{p}_B(1-\hat{p}_B)/n_B}$  and  $CI = \Delta \pm 1.96 \cdot SE(\Delta)$ . Show your steps.",
      "answer": " $\hat{p}_A=500/10,000=0.050$ ;  $\hat{p}_B=560/10,000=0.056$ ;  $\Delta=0.056-0.050=0.006$ .  $SE(\Delta) = \sqrt{[0.05 \cdot 0.95/10,000 + 0.056 \cdot 0.944/10,000]} \approx 0.003168$ . 95% CI:  $0.006 \pm 1.96 \cdot 0.003168 \Rightarrow (-0.00021, 0.01221)$ ."
    },
    "DOK 3": {
      "question": "A product manager asks whether B beats A at  $\alpha=0.05$ . Using the same data ( $x_A=500$ ,  $n_A=10,000$ ;  $x_B=560$ ,  $n_B=10,000$ ), choose the appropriate significance test for  $H_0: p_A=p_B$  and justify your choice. Then compute the test statistic and decision. Use pooled  $\hat{p} = (x_A+x_B)/(n_A+n_B)$  and  $SE_0 = \sqrt{\hat{p}(1-\hat{p})(1/n_A+1/n_B)}$ ,  $z = \Delta/SE_0$ .",
      "answer": "Use the pooled two-proportion z-test because  $H_0$  assumes equal rates.  $\hat{p} = (500+560)/(20,000) = 0.053$ .  $SE_0 = \sqrt{[0.053 \cdot 0.947 \cdot (1/10,000 + 1/10,000)]} \approx 0.0031683$ .  $\Delta = 0.006$ .  $z = 0.006/0.0031683 \approx 1.89 \Rightarrow$  two-sided  $p \approx 0.058$ . Decision at  $\alpha=0.05$ : fail to reject  $H_0$  (not statistically significant). Rationale: pooling matches the null and provides the correct SE for testing equality."
    },
    "DOK 4": {
      "question": "Your team will run a new A/B test to estimate lift and decide rollout. Design and justify a plan that: (1) defines the success metric, (2) prevents peeking, (3) handles sample ratio and seasonality risks, and (4) specifies how you will report results (effect size, uncertainty, and practical significance).",
      "answer": "Plan: (1) Success metric: risk difference  $\Delta = (x_B/n_B) - (x_A/n_A)$  on a pre-registered conversion definition. (2) Peeking: commit to a fixed-horizon stop rule (e.g., stop at N per arm) with no interim looks; analysis occurs once at the stop. (3) Allocation & seasonality: target 1:1 allocation; monitor and correct sample ratio mismatch; run over full weekly cycles or stratify by day to neutralize seasonality; ensure concurrent traffic. (4) Analysis & reporting: estimate  $\Delta$  and its 95% CI via  $SE(\Delta) = \sqrt{\hat{p}_A(1-\hat{p}_A)/n_A + \hat{p}_B(1-\hat{p}_B)/n_B}$ ; for significance, test  $H_0: p_A=p_B$  using pooled  $\hat{p}$  and  $SE_0$  with  $z = \Delta/SE_0$ ; report  $\Delta$ , CI, p-value, and a practical readout (e.g., expected revenue lift per 1,000 visitors). Include limitations: large-sample approximation, residual seasonality risk, and any allocation imbalance."
    }
  }
}
'''

```

Cell Biology — Michaelis–Menten Enzyme Kinetics

Many enzymes obey ($v = \frac{V_{\max}[S]}{K_m + [S]}$). (V_{\max}) is the maximal rate when active sites are saturated; (K_m) is the substrate concentration at half-maximal rate and reflects **apparent** affinity (lower (K_m) \Rightarrow higher affinity). At ($[S] \ll K_m$), rate is first-order ($v \approx \frac{V_{\max}}{K_m}[S]$); at ($[S] \gg K_m$), zero-order ($v \approx V_{\max}$). Competitive inhibitors raise the **apparent** (K_m) (need more substrate) without changing (V_{\max}); noncompetitive lower (V_{\max}) without changing (K_m). Turnover number ($k_{\text{cat}} = V_{\max}/[E]_{\text{T}}$); catalytic efficiency (k_{cat}/K_m) compares enzymes near diffusion limits. Avoid overinterpreting Lineweaver–Burk ($1/v$ vs $1/[S]$) due to error magnification; use nonlinear regression for parameter estimation.

```

'''json
{
  "content": {
    "DOK 1": {
      "question": "Define  $K_m$  in Michaelis–Menten kinetics.",

```

```

    "answer": "K_m is the substrate concentration at which the reaction rate v equals one-half of V_max."
  },
  "DOK 2": {
    "question": "You study an enzyme with V_max = 120 μM·min-1 and K_m = 30 μM. Compute the approximate rate and identify the kinetic order for (a) [S] = 3 μM and (b) [S] = 300 μM using the appropriate Michaelis–Menten approximations.",
    "answer": "(a) [S] << K_m → v ≈ (V_max/K_m)[S] = (120/30)·3 = 12 μM·min-1; first-order in [S]. (b) [S] >> K_m → v ≈ V_max = 120 μM·min-1; zero-order in [S].",
  },
  "DOK 3": {
    "question": "An enzyme has (V_{max}=100, \mu\text{M}\cdot\text{min}^{-1}) and (K_m=20, \mu\text{M}). A noncompetitive inhibitor halves (V_{max}) to (50, \mu\text{M}\cdot\text{min}^{-1}) with (K_m) unchanged. At ([S]=200, \mu\text{M}), you may do one: (A) increase ([S]) fivefold, (B) double total enzyme ([E]*T), or (C) add a competitive inhibitor. Assume (V_{max}=k_{cat}[E]*T) (so doubling ([E]*T) doubles (V_{max})); competitive inhibition raises apparent (K_m) without changing (V_{max}); noncompetitive lowers (V_{max}) without changing (K_m). Which choice yields the largest increase in rate and why? Justify using Michaelis–Menten relations.",
    "answer": "Choose (B) double [E]_T. With [S] >> K_m, v ≈ V_max. Noncompetitive inhibition lowers V_max, so raising [S] (A) has negligible effect when v is V_max-limited. Because k_cat = V_max/[E]_T, increasing [E]_T restores/raises V_max (here from 50 back toward 100 μM·min-1), directly increasing v. Adding a competitive inhibitor (C) raises apparent K_m without changing V_max, which does not help at high [S].",
  },
  "DOK 4": {
    "question": "Design and justify a plan to estimate V_max, K_m, and catalytic efficiency (k_cat/K_m) for an enzyme ± inhibitor. Your plan must specify data collection, fitting method, validation, and how you will interpret inhibitor type.",
    "answer": "Plan: (1) Collect initial-rate data v at ≥8 substrate concentrations spanning 0.1·K_m_est to 10·K_m_est for both conditions (± inhibitor), keeping [E]_T constant and measuring initial slopes. (2) Fit v = (V_max[S])/(K_m + [S]) by nonlinear regression (avoid Lineweaver–Burk due to error magnification at low [S]); report V_max and K_m with 95% CIs. (3) From independently measured [E]_T, compute k_cat = V_max/[E]_T and catalytic efficiency k_cat/K_m. (4) Diagnose inhibitor type: if K_m increases while V_max unchanged → competitive; if V_max decreases while K_m unchanged → noncompetitive. (5) Validate with residual plots, goodness-of-fit (R2_adj), and parameter uncertainty; perform a lack-of-fit test. (6) Sensitivity check: refit after removing lowest-[S] point to assess robustness against low-[S] noise. (7) Report limitations (pipetting error, substrate depletion) and mitigation (replicates, randomized [S] order).",
  }
}
}
...
---

```

Policy & Ethics — Carbon Tax vs Cap-and-Trade

Both instruments price carbon but differ in **control variable**. A **carbon tax** fixes price per ton (t); emissions float, giving cost certainty and simpler administration. **Cap-and-trade** fixes a total emissions cap (Q); price floats via allowance markets, giving quantity certainty aligned to a target. Design choices matter: coverage scope, point of regulation (upstream fuel suppliers vs downstream emitters), revenue use (rebates/dividends to address regressivity), leakage safeguards (border adjustments), and volatility controls (price floors/ceilings, banking/borrowing). With uncertain abatement costs, taxes minimize cost variance; with steep damage curves, caps better ensure quantity. Hybrid designs (cap with price collar) blend both. Ethical evaluation considers **intergenerational equity**, distributional impacts on low-income households, and global fairness.

```

```json
{
 "content": {
 "DOK 1": {
 "question": "Carbon pricing uses two main instruments. Define the control variable of each: a carbon tax fixes _____, while cap-and-trade fixes _____.",
 "answer": "A carbon tax fixes the price per ton of CO2e; cap-and-trade fixes the total emissions quantity (the cap).",
 },
 "DOK 2": {
 "question": "You must choose a carbon-pricing tool for a jurisdiction with highly uncertain abatement costs and a preference for administrative simplicity. A carbon tax fixes price; cap-and-trade fixes quantity. Select the more suitable instrument and name two design choices to (a) protect low-income households and (b) limit price volatility.",
 }
 }
}

```



## 9.5 Gioia's Data Structure

### Color Codes

Math	Eng. + CS	General
Script	Text Slides	Image Slides

Paraphrases or participants marked with an asterisk "\*" did not write the texts themselves, but were noted from interviews.

Participant	1 <sup>st</sup> Order Concepts	2 <sup>nd</sup> Order Themes	Aggregated Dimensions
S1	Especially the questions of difficulty 4/4 aim clearly beyond the learning objectives of my lecture. Things are asked that are described nowhere in the lecture notes (e.g., multi-phase action plan, survey instrument).	Content Alignment and Scope Accuracy	Generated questions frequently diverge from lecture boundaries by introducing external concepts ("hallucinations") or prioritizing peripheral examples over core competencies. While often factually correct in a general sense, the output often lacks contextual validity for specific learning objectives. Furthermore, technical modules suffer from a critical absence of quantitative calculation tasks, failing to cover necessary assessment modalities required for the subject.
S2	In 3/4, topics are combined that do not belong together.		
S4	They are also formulated quite completely (containing the formulas). 3AB, 4ABC: Questions too detailed; they go beyond the slides (in terms of accuracy and scope).		
S6	3C >>> Uses content that is not part of the lecture (costs). The logic of the task is not apparent. 4B >>> Accessibility was not lecture content or exceeds the knowledge level.		
S7	Unfortunately, the generated questions regarding the temperature dependence of diffusion cover only one aspect, and not one that is very central to this module. What bothers me about the questions is that there are no calculation tasks. The ideal gas law as well as the concept of heat capacity offer numerous possibilities for questions of varying difficulty levels.		
S8	Challenge: Generated questions must match the lecture content. In the examples above, there were many questions that go far beyond the lecture content and where the tool addresses topics that do not appear anywhere in the documents. Could this have to do with the fact that there are links (e.g., image sources) in the document where the tool also searches for content?		
S9	The lecture notes refer to an input for the Ind 4.0 module, where I organize two evenings (2x3h). The dedicated input for Ind 4.0 is about 2h. This means all questions, except for the comprehension questions in the 1st difficulty level, are not treated in that depth and execution.		
S15	Most questions refer to the example of planar structures described in the introduction of the script. However, this topic is not actually part of the module's learning content. It is not about understanding statics (axial forces, assumptions of static equilibria, etc.), but the mathematics of linear systems of equations. In contrast, very many topics of the script do not appear at all, remaining unconsidered.		
S16	Task 4C contains concepts (RMS, uncertainty, outliers, runtime) that are not covered in the lecture notes.		
*F1	Proof-of-concept not in the lecture notes.		

	<p>Persona creation process not suitable for the module.</p> <p>MVP not mentioned often enough in the lecture notes.</p> <p>S4C -&gt; interesting question but <b>makes no sense with the topic.</b></p> <p>S4D -&gt; not the topic of the module.</p>		
S2	The quality of the questions would need to improve significantly.	Linguistic Formulation and Structural Logic	<p>logic (unrealistic scenarios). Participants strictly differentiate between actual cognitive difficulty and mere textual complexity, rejecting unnecessarily long or convoluted phrasing. Inconsistent language levels, phrasing (formal vs. informal), and "daunting" text lengths further detract from the structural quality of the output.</p> <p>The output often fails to align with specific student demographics or semester levels, with difficulty rankings perceived as inconsistent, inaccurate, or inverted. A fundamental clash is observed between the tool's linear, causal logic and the associative requirements of design disciplines. While useful for checking basic reproduction, the system struggles to reliably generate tasks that test higher-order transfer skills or "thinking further" without manual adjustment.</p> <p>Participants view the tool not as an autonomous generator, but as a valuable "junior assistant" for</p>
S3	Especially with the more complex questions, there is <b>no unambiguous answer.</b> This would have to be taken into account during automatic evaluation.		
S4	<p>The questions are formulated quite well.</p> <p><b>1A: Formulated too simply.</b></p> <p><b>1C: Formulated too openly.</b></p> <p><b>2B: Question almost answers itself.</b></p> <p>3C: Question answers itself?</p> <p>Some questions were formulated <b>using "Du" (informal), others with "Sie" (formal);</b> is this configurable?</p>		
S6	<p>1D &gt;&gt;&gt; Same question as B.</p> <p><b>2A &gt;&gt;&gt; Unrealistic or illogical/incomprehensible formulation.</b></p> <p>2B &gt;&gt;&gt; Unrealistic or illogical/incomprehensible formulation.</p> <p>2C &gt;&gt;&gt; Realistic and understandable task, formulated too complicatedly.</p> <p>2D &gt;&gt;&gt; Realistic and understandable task, <b>formulated too complicatedly.</b> Questionable whether the task can be posed only textually.</p> <p><b>3A &gt;&gt;&gt; Does not work without an image example.</b></p> <p>3B &gt;&gt;&gt; Uses non-subject-related terms, but good as a written task (perhaps formulate somewhat simpler).</p> <p>3D &gt;&gt;&gt; Good and understandable question, but <b>too imprecise.</b></p> <p>4A &gt;&gt;&gt; Imprecise, illogical, <b>too extensive task.</b></p> <p>4C &gt;&gt;&gt; Imprecise, illogical, too extensive task.</p> <p>4D &gt;&gt;&gt; TOP!</p>		
S7	The questions are, in my opinion, not particularly interesting.		
S10	The textual length of the question/task is an important criterion for me: The shorter and more compact, the better. <b>Long task texts appear daunting and confusing.</b> The difficulty level of a task should, if possible, not be achieved by lengthening the question.		
S11	The questions are formulated too complexly for the students. Since many of my students are not native English speakers, the <b>language level is too high overall.</b>		
	Sometimes the questions also go into too much detail.		
S17	My impression is that the question sets bring out too disparate topics.		

	<p>Furthermore, some questions require even more context information regarding the municipality, which does not occur in the first question set.</p> <p>It should be clear beforehand whether the questions refer to an individual and/or municipality or politics.</p>		
*F1	2D statement is incorrect.		
S3	<p>*Varying perceived difficulties in D1.</p> <p>*In D2 (Bloom), the 'Understand' level of one version is better than the other, whereas for the 'Apply' level, it is the other way around.</p>		
S5	<p>This module is held in the 3rd semester (intermediate). For this, some questions appear too complex and demanding. For a course in the advanced area, this would look different.</p> <p>However, I wondered if—or to what extent—the semester level (here 3rd/Intermediate) is factored into the difficulty levels?</p>		
S6	<p>That was very exciting!</p> <p>From my point of view, however, not yet suitable for visual or design-related professions.</p> <p>My impression is: A linear causal relationship between concepts is often assumed, which then leads to illogical or incomprehensible phrasings.</p> <p>Furthermore, the questions seem to be based on scientific logic rather than design logic, which dissolves connections and hierarchies.</p>		
S7	<p>The ability to make quantitative calculations and estimations using physical laws is a core competence to be acquired in these modules, and is tested accordingly.</p> <p>Hence my assessment not to ask questions of the suggested type.</p> <p>In my opinion, difficulty levels 1 and 2 are below the University of Applied Sciences (FH) level.</p> <p>I also struggled partly with the ranking. Within one variant, the differences between the two posed questions were greater than the differences between different variants (e.g., at Level 2).</p> <p>If a qualitative task is to be posed, the answer should preferably not be found prominently in the lecture notes.</p> <p>I would find it more interesting to formulate a question in the sense of "thinking further," where the answer is not clear beforehand.</p>	Pedagogical Appropriateness and Difficulty Calibration	drafting and inspiration. Acceptance is contingent on a "human-in-the-loop" workflow, where educators manually adapt output for exams or use it for student self-study. While trust in automated grading is limited to simple tasks, there is a clear demand for deeper integration features (e.g., answer generation, slide compatibility) to enhance utility.
S9	<p>Actually, I can only use the questions for the 1st difficulty level and, with reservations, questions for the 2nd.</p> <p>In my opinion, the learning scope (in hours and depth) and the learning objectives should also be specified during prompting.</p> <p>Likewise the target group with existing specialist knowledge. With us, it is T&amp;A students from technical departments who have chosen the module as an additional module. My part as a Business Engineering lecturer is, among other things, to show the business relevance of digital transformation. Less the implementation process from a manufacturing technology view.</p>		
S11	<p>It is important to me that the students understand the fundamental connections—for example, why one uses IBP and how it works—and not that they reproduce content one-to-one or have to memorize specific details like the primary flows.</p> <p>However, for checking the students' actual understanding, I consider the case studies in their current form to be less suitable.</p>		
S12	<p>The task pool covers the spectrum well from basics (definitions, bot types) to strategic topics (governance, operating model, risk, pilot planning) and forces students to use genuine transfer and judgment skills instead of just reproduction.</p> <p>The scenarios are realistic (regulated industries, legacy systems, citizen dev vs. IT) and allow for a clear differentiation between weak and strong performances.</p> <p>Overall, from my point of view, this is certainly a didactically coherent, demanding set.</p>		

	Individual questions are quite good.	
S14	Could you indicate the taxonomy levels for the questions in addition to the difficulty?	
S16	The difficulty with these tasks is comparability. How do you ensure that a question is comparable to another question in terms of difficulty level and effort? For example, questions 2B and 3C (first question each) are <b>very similar but appear in different difficulty levels</b> . On the other hand, questions 4C and 4D are very different, and I am not sure if they are on the same difficulty level. This is a fundamental problem that also arises with manual grading.	
S18	I find levels 3 and 4 partly very far-reaching regarding the application context.	
S20	The questions range from very good to poor quality (3/4 Variant A) across the entire spectrum.	
*F1	Persona descriptions not relevant. Open Book MEP (final exam) requires different exercises (possibly in lecture). 2A too simple (text too short). <b>Difficulty 2 &gt; 1.</b> 2D too simple (pure busywork). S4A -> <b>too difficult because treated too little</b> (style/method is interesting). <b>S4B -&gt; too simple with open book.</b>	
S2	Generally, I do not like the questions for this module. Better questions were generated in the other module. However, I find the approach of generating questions from the lecture notes good.	
S3	Some of the questions above I would not have used. Mostly (especially Levels 3 & 4), I <b>would not use them directly as they are; instead, I would adapt them slightly</b> and then use them.	
S4	For an exam, I would adapt them (removing the formulas, since a summary sheet is allowed, or potentially providing a formula collection at the end. I find formulas in the task description to be too much assistance). For students, automatically generated questions would be <b>helpful for studying – ideally with solutions</b> (or they could contact me if they had questions about the generated solutions). I would select the (official learning) questions to be able to set priorities. <b>Without automatic grading, the questions are useful as long as there is a sample solution.</b> Otherwise, rather less so from my point of view.	Practical Utility, Acceptance and Integration
S5	I find some questions good and will likely use something similar in an adapted form in the final module exam (MEP).	
S9	For <b>simple questions</b> and clearly defined answers (e.g., multiple choice or querying definitions), I would have them <b>corrected automatically</b> . For <b>answers with room for interpretation, I would double-check</b> the automatic correction, potentially randomly. If there is agreement, I would also correct automatically.	
S11	However, I find some questions, such as Variant C in Task 2/4, very successful. Overall, the material could be well suited for <b>generating multiple-choice questions</b> or for preparing content for the general part of the exam that requires memorization (pure knowledge questions).	

S12	I even adopted one of them for the upcoming final module exam (MEP) (don't tell anyone).
S13	In your survey, surprisingly good questions were generated from my lecture notes in my opinion. I think your master's thesis makes a valuable contribution to future teaching!
S14	Does question generation also work based on slides? I assume you also create the answers. I assume one can also have multiple-choice questions created.
S15	On the other hand, the tasks are creative and provide food for thought for new tasks. However, I would not use them directly in an automated way, but only after thorough checking and revision.
S20	I would use the very good questions as exam questions.
*F1	Tool first as inspiration -> then generate variants with ChatGPT. Provide old MEPs as templates for new questions.

## 9.6 OLS Results

### 9.6.1 Summary Cost

Variable	Coefficient	Std.Error	t-statistic	p-value	Observations	84
<b>Intercept</b>	-0.0003	0.004	-0.073	0.942	<b>R-squared</b>	0.377
<b>Image Slides</b>	-0.0075	0.003	-2.618	0.011	<b>Adj. R-squared</b>	0.337
<b>Text Slides</b>	-0.0086	0.004	-2.302	0.024	<b>F-statistic</b>	9.423
<b>Math</b>	0.0120	0.003	4.142	0.000		
<b>General</b>	0.0009	0.003	0.338	0.736		
<b>German</b>	0.0106	0.004	2.948	0.004		

### 9.6.2 Concept Extraction Cost per Chunk

Variable	Coefficient	Std. Error	t-statistic	p-value	Observations	120
<b>Intercept</b>	0.0006	0.000	4.111	0.000	<b>R-squared</b>	0.564
<b>Valid Chunk</b>	0.0008	7.89e-05	10.492	0.000	<b>Adj. R-squared</b>	0.541
<b>Image Slides</b>	0.0002	0.000	1.863	0.065	<b>F-statistic</b>	24.41
<b>Text Slides</b>	-9.038e-05	0.000	-0.638	0.525		
<b>Math</b>	-0.0002	9.17e-05	-2.359	0.020		
<b>General</b>	-6.205e-05	8.5e-05	-0.730	0.467		
<b>German</b>	-0.0001	0.000	-0.964	0.337		

### 9.6.3 Question Generation Cost per Chunk

Variable	Coefficient	Std. Error	t-statistic	p-value	Observations	127
<b>Intercept</b>	0.0088	0.003	2.623	0.010	<b>R-squared</b>	0.821
<b>Valid Chunk</b>	0.0340	0.002	21.456	0.000	<b>Adj. R-squared</b>	0.808
<b>Sliding Window</b>	-0.0016	0.001	-1.211	0.228	<b>F-statistic</b>	67.49
<b>Webb</b>	-0.0007	0.001	-0.526	0.600		
<b>Image Slides</b>	-0.0040	0.002	-2.090	0.039		
<b>Text Slides</b>	0.0052	0.003	1.840	0.068		
<b>Math</b>	0.0039	0.002	2.067	0.041		
<b>General</b>	0.0026	0.002	1.609	0.110		
<b>German</b>	-0.0054	0.003	-2.080	0.040		

### 9.6.4 Document to Markdown Conversion Time

Variable	Coefficient	Std. Error	t-statistic	p-value	Observations	84
<b>Intercept</b>	-11.0121	5.140	-2.142	0.035	<b>R-squared</b>	0.855
<b>Image Slides</b>	3.8520	3.857	0.999	0.321	<b>Adj. R-squared</b>	0.843
<b>Text Slides</b>	11.7124	5.036	2.326	0.023	<b>F-statistic</b>	75.51
<b>Math</b>	8.4754	3.753	2.258	0.027		
<b>General</b>	4.2883	3.391	1.265	0.210		
<b>German</b>	-2.2026	4.764	-0.462	0.645		
<b>Num Characters</b>	0.0003	1.48e-05	17.555	0.000		

### 9.6.5 Summary Generation Time

Variable	Coefficient	Std. Error	t-statistic	p-value	Observations	84
<b>Intercept</b>	4.9832	2.909	1.713	0.091	<b>R-squared</b>	0.119
<b>Image Slides</b>	-0.4764	2.131	-0.224	0.824	<b>Adj. R-squared</b>	0.050
<b>Text Slides</b>	0.1144	2.750	0.042	0.967	<b>F-statistic</b>	1.734
<b>Math</b>	2.6712	2.302	1.161	0.249		
<b>General</b>	3.6886	1.929	1.912	0.060		
<b>German</b>	3.1599	2.714	1.164	0.248		
<b>Num Tokens</b>	3.007e-05	2.09e-05	1.442	0.153		

### 9.6.6 Concept Extraction Generation Time

Variable	Coefficient	Std. Error	t-statistic	p-value	Observations	42
<b>Intercept</b>	6.5584	4.304	1.524	0.137	<b>R-squared</b>	0.333
<b>Image Slides</b>	2.9644	1.820	1.629	0.113	<b>Adj. R-squared</b>	0.196
<b>Text Slides</b>	-1.6100	2.172	-0.741	0.464	<b>F-statistic</b>	2.424
<b>Math</b>	-2.7249	1.628	-1.674	0.103		
<b>General</b>	-0.7450	1.554	-0.479	0.635		
<b>German</b>	-3.5902	2.007	-1.789	0.083		
<b>Invalid Chunks</b>	-1.5430	1.648	-0.936	0.356		
<b>Total Chunks</b>	3.7282	1.797	2.075	0.046		

### 9.6.7 Question Generation Time

Variable	Coefficient	Std. Error	t-statistic	p-value	Observations	84
<b>Intercept</b>	139.8389	101.329	1.380	0.172	<b>R-squared</b>	0.424
<b>Sliding Window</b>	-157.7938	47.083	-3.351	0.001	<b>Adj. R-squared</b>	0.354
<b>Webb</b>	-9.5681	45.969	-0.208	0.836	<b>F-statistic</b>	6.058
<b>Image Slides</b>	-163.3933	67.200	-2.431	0.017		
<b>Text Slides</b>	-19.9586	87.362	-0.228	0.820		
<b>Math</b>	171.2582	66.300	2.583	0.012		
<b>General</b>	161.6647	65.081	2.484	0.015		
<b>German</b>	8.1136	83.172	0.098	0.923		
<b>Invalid Questions</b>	-123.4381	53.400	-2.312	0.024		
<b>Total Questions</b>	122.1277	28.925	4.222	0.000		