Concept Extraction and Webb's Depth of Knowledge: Comparing LLM Question Generation Pipelines for Educational Assessment

Dmitriy An, Andrew Paice, Petra Müller-Csernetzky, and Aliaksei Andrushevich
Lucerne University of Applied Sciences and Arts (HSLU)
{dmitriy.an, andrew.paice,
petra.mueller-csernetzky, aliaksei.andrushevich}@hslu.ch

Abstract

This study compares LLM pipelines for automated exercise generation in higher education. We empirically compare two context preparation methods (Sliding Window vs. Concept Extraction) in combination with two instructional frameworks (Bloom's Revised Taxonomy vs. Webb's Depth of Knowledge). Through a mixed-methods evaluation with 21 university course coordinators, we find that Concept Extraction combined with Webb's Depth of Knowledge yields the highest pedagogical quality, especially for technical disciplines. While human oversight remains necessary to mitigate out-of-scope hallucinations, these pipelines serve as efficient drafting engines for scalable, high-quality academic assessments.

1 Introduction

Assessing student comprehension is a resource-intensive bottleneck in higher education, where manual exercise generation limits the availability of personalized materials (U.S. Department of Education and Office of Educational Technology, 2023). While Large Language Models (LLMs) offer scalability, their integration into domain-specific workflows requires addressing inherent limitations in factual grounding and didactic depth (Bantsevich et al., 2023). Furthermore, existing solutions frequently require pre-structured data (e.g., XML), struggling with the unstructured PDFs and slides common in university environments (Nguyen et al., 2022).

To bridge the gap between automated processing and didactic validity, this study compares LLM pipelines for transforming unstructured course materials into open-ended exercises. We empirically compare two context preparation methods (Sliding Window vs. Concept Extraction) and two instructional frameworks: Bloom's Revised Taxonomy (Anderson, 2009) and Webb's Depth of Knowledge (Webb, 2002). Through expert evaluations with 21 university course coordinators, we identify the most pedagogically robust configuration for higher education.

2 Related Work

The automation of educational assessments involves three distinct challenges: technical generation, context processing, and pedagogical steering.

2.1 Automated Exercise Generation

Early automated question generation relied heavily on rule-based Natural Language Processing (NLP) or LSTM networks (Hochreiter and Schmidhuber, 1997) to produce Multiple Choice Questions (MCQs) and fill-in-the-blank exercises (Killawala, Khokhlov, and Reznik, 2018; Ch and Saha, 2020, 2023). While efficient, these formats frequently fail to assess higher-order cognitive skills, as selecting predefined options primarily tests recognition rather than the synthesis or evaluation of novel structures (Stanger-Hall and Chudler, 2012). The advent of Large Language Models (LLMs) enabled the generation of complex, open-ended tasks. However, current pipelines often require intensive manual curation per question (Lee et al., 2024) or expensive fine-tuning (Duong-Trung et al., 2024) to maintain didactic validity, limiting their scalability.

2.2 Context Preparation

Processing unstructured academic documents without losing semantic integrity remains a bottleneck. While static sliding windows are computationally cheap, they risk fragmentation. Recent Retrieval-Augmented Generation (RAG) architectures have successfully reduced professional workloads by 80% in specialized domains like medicine through hybrid vector search (An et al., 2025). Similarly, Noorbakhsh et al. (2025) used concept extraction for MCQs. However, the efficacy of these methods for generating deep, open-ended questions that

preserve a document's narrative arc remains underexplored.

Beyond traditional RAG, recent advancements in document-to-dialogue transformation, such as Dialogue Inpainting (Dai et al., 2022) and Book2Dial (Wang et al., 2024), demonstrate the efficacy of using LLMs to synthesize structured educational interactions from textbooks. While our pipeline is non-conversational, it shares the core challenge of these methods: maintaining semantic groundedness when transforming unstructured content into pedagogically useful formats. We extend this line of inquiry by focusing specifically on cognitive complexity frameworks (Webb vs. Bloom) rather than conversational flow.

2.3 Cognitive Frameworks in LLMs

To steer LLM outputs pedagogically, researchers predominantly rely on Bloom's Taxonomy (Maity, Deroy, and Sarkar, 2025; Scaria, Dharani Chenna, and Subramani, 2024). However, Bloom's focus on cognitive verbs (e.g., "Understand" vs. "Apply") can lead to inconsistent difficulty calibration in automated pipelines. Webb's Depth of Knowledge (DOK), which categorizes tasks by cognitive complexity rather than verbs, has shown theoretical promise for technical LLM prompting (Yu et al., 2025). Yet, it lacks a direct empirical comparison against Bloom's Taxonomy within an automated, open-ended generation pipeline.

3 Methodology

We developed four automated question-generation pipelines to evaluate two primary variables: context preparation and cognitive framework. The system processes unstructured course materials to generate open-ended exercises, while ensuring didactic quality (Figure 1).

3.1 Pipeline Architecture

Documents (PDF, DOCX, PPTX) are summarized and converted to Markdown via `pymupdf4llm` (GNU AGPL-3.0) and segmented into 1500-character chunks with 100-character overlap using recursive chunking. All LLM tasks, including summarization, concept extraction, and question generation, utilize GPT-5-mini (OpenAI, 2025) due to its cost-efficiency. We compare two methods:

Sliding Window. Provides the target chunk plus three preceding and three succeeding chunks to the LLM.

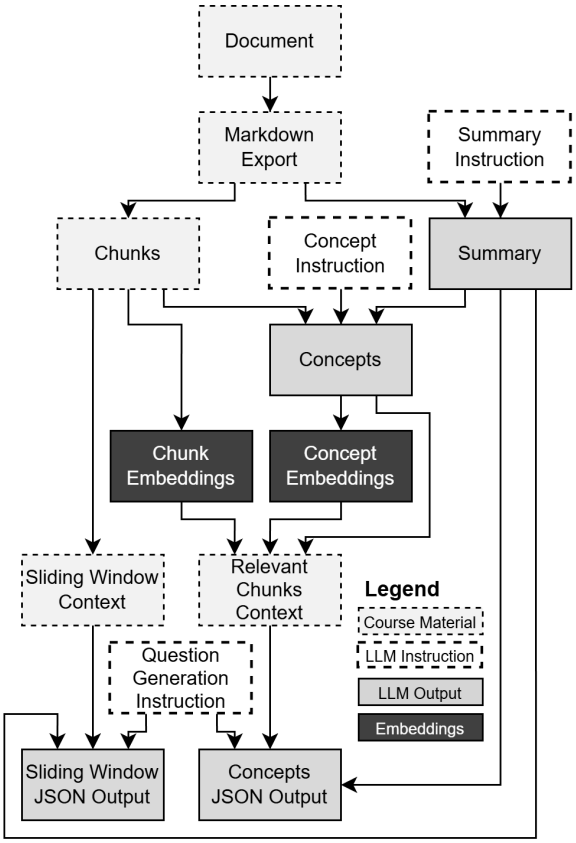

Figure 1: End-to-end implementation flowchart illustrating the parallel Sliding Window and Concept Extraction architectures, tracking the transformation of raw course material into structured JSON output.

Concept Extraction (RAG). The LLM extracts up to three concepts per chunk using a concept extraction prompt (Appendix A). It uses Qwen3-Embedding-0.6B model (Zhang et al., 2025) to calculate vector representations and retrieves the three most relevant chunks for each concept via cosine similarity: $\text{sim}(\mathbf{A}, \mathbf{B}) = (\mathbf{A} \cdot \mathbf{B})/(\|\mathbf{A}\|\|\mathbf{B}\|)$.

3.2 Cognitive Frameworks & Generation

We map Bloom and Webb levels to four difficulty tiers from 1 (simple) to 4 (complex), based on Hess' Cognitive Rigor Matrix (Hess et al., 2009), resulting in four configurations (Table 1).

	Bloom	**Webb**
Sliding Window	Pipeline 1	Pipeline 2
Concept Extraction	Pipeline 3	Pipeline 4

Table 1: Evaluated Pipeline Combinations

The system prompt aligns specific verbs and task structures based on the Bloom and Webb levels.

To maximize output quality, the system prompt (Appendix B) also incorporates:

Single-Pass Generation with Chain-of-Thought. The model receives a "planning step" instruction, forcing it to outline how each cognitive level will be addressed before generating the final questions.

Few-Shot Learning. Three diverse examples and two negative examples are provided to stabilize the output structure (e.g., JSON formats). Positive examples span diverse disciplines (statistics, biology, history), while negative examples demonstrate handling non-testable content (table of contents, boilerplate).

Didactic Constraints. Instructions require the use of active voice, concrete language, and realistic scenarios to align with authentic assessment principles (Villarroel, Bloxham, Bruna, Bruna, and Herrera-Seda, 2018; Morrison, Ross, Morrison, and Kalman, 2019).

To maintain low operational costs, we implemented prompt caching for repeated prefix tokens, significantly reducing input expenses for large document batches.

3.3 Evaluation Design

A mixed-methods approach was used to validate the pipelines. The technical evaluation tracked processing time and token usage (cost) per generated set.

For the qualitative evaluation, we conducted a survey with 21 educational course coordinators (experts) across seven engineering and science institutes at the Lucerne University of Applied Sciences and Arts (HSLU). Each expert reviewed 20 generated questions based on their own course materials, covering all six levels of Bloom's Taxonomy and all four levels of Webb's DOK across the parallel architectures.

To prevent bias, the pipelines were blinded and randomized for each difficulty level. Participants ranked the output from 1 (best suited) to 4 (least suited). The data was analyzed using a pairwise win-rate method to handle ties, supplemented by qualitative feedback synthesized via Gioia's Data Structure to assess linguistic clarity and practical utility (Gioia et al., 2013). Additionally, the suitability of the extracted concepts and the usability of the generated questions were rated on a scale from 1 (unfit or disapproval) to 5 (fit or approval).

4 Results

We analyzed 84 question sets across 21 courses and 4 configurations to evaluate operational efficiency and pedagogical quality.

4.1 Operational Efficiency: Time and Cost

The Sliding Window approach is significantly faster, completing a question set in an average of 232 seconds, compared to 399 seconds for Concept Extraction.

Generating a complete question set costs an average of $0.045 using the Sliding Window method and $0.053 with Concept Extraction. Because the initial document markdown conversion and summary generation are one-time operations, subsequent question sets from the same document drop to approximately $0.04. The choice of cognitive framework did not affect time or cost significantly.

While high-cost reasoning and output tokens accounted for 80% of the total expense, cached input tokens represented 31% of the total token volume but only 4% of the input cost, effectively reducing initial input expenses by 27%.

4.2 Quantitative Survey Results

The pairwise win-rate analysis of the 21 educational course coordinators' rankings identified preferences regarding both variables.

Cognitive Frameworks. Webb's DOK outperformed Bloom's Revised Taxonomy by a 6% margin overall. Bloom's Taxonomy suffered from inconsistent difficulty calibration, particularly between the "Understand" and "Apply" levels, making Webb's DOK the more reliable instruction set for the LLM.

Context Preparation. Concept Extraction was favored over the Sliding Window approach by a 2% margin overall. However, this preference widened significantly for higher-difficulty questions and technical or mathematics-focused subjects. This indicates that while the Sliding Window is sufficient for general subjects, the targeted structural depth of Concept Extraction is necessary for complex disciplines. Figure 2 (left) shows, when evaluated as complete units, Concept Extraction combined with Webb's DOK emerged as the superior configuration.

Usability. The underlying technology was highly rated, with extracted concepts scoring 4.18 out of 5 for suitability. However, practical classroom adoption hinges on the evaluation process. Lecturers

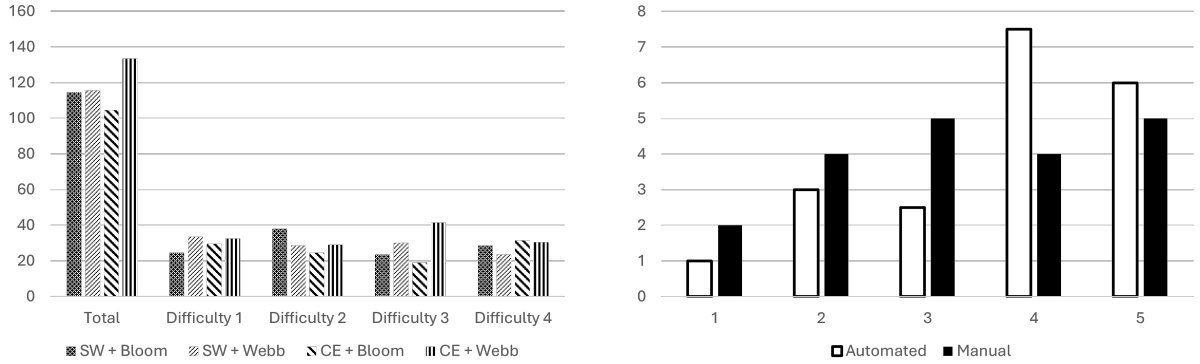

Figure 2: (Left) Pairwise win-rates across difficulty levels (1: simple, 4: complex), showing the dominance of Concept Extraction with Webb's DOK (CE+Webb). (Right) Lecturer usability ratings (1: unfit/disapproval, 5: fit/approval) for automated vs. manual evaluation.

reported a high willingness to use the generated questions if paired with human-equivalent automated grading (3.73/5). If manual grading is required, usability drops significantly (3.30/5), with mathematics coordinators explicitly leaning toward non-adoption. The histograms are shown in Figure 2 on the right.

4.3 Qualitative Insights

A thematic synthesis of coordinator feedback revealed four core constraints:

Content Alignment and Scope Accuracy. Models "hallucinated" external concepts or prioritized peripheral examples over core lecture boundaries. Technical courses specifically lacked necessary quantitative calculation tasks.

Linguistic and Structural Logic. Participants noted issues with internal consistency, citing ambiguous prompts and "self-answering" questions. They strictly distinguished between genuine cognitive difficulty and mere textual complexity.

Pedagogical Appropriateness and Difficulty Calibration. Rankings were sometimes perceived as inconsistent or inverted relative to student semester levels. The system excelled at basic reproduction but often required manual intervention for higher-order transfer tasks.

Practical Utility, Acceptance, and Integration. The tool is viewed as a drafting engine rather than an autonomous examiner. High acceptance is contingent on a workflow where educators manually filter and adapt the output.

5 Discussion and Conclusion

Our findings establish that Concept Extraction combined with Webb's Depth of Knowledge (Pipeline 4) yields the highest pedagogical quality, particularly for technical disciplines.

5.1 Didactic and Technical Implications

A major contribution of this work is the empirical comparison between cognitive frameworks. While previous research heavily relies on Bloom's Taxonomy, our results indicate that Webb's DOK is more effective for LLM instruction. Bloom's cognitive processes (e.g., "Understand" vs. "Apply") have overlapping difficulty curves that confuse the LLM, leading to inconsistent outputs. Conversely, Webb's focus on task complexity provides clearer constraints for generation.

Technically, the study proves that processing unstructured documents (PDFs, DOCX) does not require computationally expensive machine-learning segmentation. Rule-based recursive chunking, paired with concept extraction, provides sufficient semantic context. Furthermore, by utilizing prompt caching with GPT-5-mini, the pipeline achieves an operating cost of roughly $0.05 for the first question set.

5.2 Limitations and Domain Friction

Despite these successes, qualitative feedback revealed clear domain-specific limitations. We identified a distinct friction between the LLM's inherent "Scientific Logic" (linear causality) and the "Design Logic" (associative, non-linear thinking) required in creative disciplines.

Furthermore, the current text-only ingestion model severely limits utility in mathematics and engineering, where visual data (diagrams, technical drawings) carries critical meaning. Finally, while the LLM successfully generated complex questions,

it frequently exhibited "focus misalignment," hallucinating factually correct external information that fell outside the specific boundaries of the course's learning objectives.

Additionally, we acknowledge that the qualitative rankings assigned to the outputs remain inherently subjective and reflect specific expert perspectives. Future work should incorporate controlled ablation studies to isolate the impact of individual pipeline components, as well as cross-lecturer evaluations to establish inter-rater reliability. Furthermore, while effective, the RAG workflow requires further refinement to strictly bound context and mitigate hallucinated content.

5.3 Conclusion

Modern Large Language Models function as highly effective drafting engines, drastically reducing the time required to overcome "writer's block" when designing university assessments. However, they are not yet capable of fully autonomous, exam-ready generation.

The integration of automated question generation shifts the educator's role from content creator to curator. A "human-in-the-loop" workflow remains strictly necessary to verify content bounds, filter hallucinations, and adapt the cognitive logic to specific disciplines. Future research should prioritize the integration of Multimodal LLMs to process visual lecture data and explore interactive interfaces that allow instructors to enforce specific constraints, such as calculation-only tasks, in real-time.

Acknowledgements

The authors thank edisconet AG for providing the research question that initiated this study.

References

D. An, A. Paice, C. Brockes, A. Sigaroudi, and M. Brockes. 2025. Retrieval-Augmented Generation for Telemedicine: A Privacy-Preserving AI Assistant for Healthcare.

Lorin W. Anderson, editor. 2009. *A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives*, abridged ed., [nachdr.] edition. Longman, New York Munich.

K. Bantsevich, M. Kovalev, V. Tsishchanka, N. Malinovskaya, and A. Andrushevich. 2023. Integration of large language models with knowledge bases of intelligent systems. *Repository BSUIR: Home*.

Dhawaleswar Rao Ch and Sujan Kumar Saha. 2020. Automatic Multiple Choice Question Generation From Text: A Survey. *IEEE Transactions on Learning Technologies*, 13(1):14–25.

Dhawaleswar Rao Ch and Sujan Kumar Saha. 2023. Generation of Multiple-Choice Questions From Textbook Contents of School-Level Subjects. *IEEE Transactions on Learning Technologies*, 16(1):40–52.

Zhuyun Dai, Arun Tejasvi Chaganty, Vincent Y Zhao, Aida Amini, Qazi Mamunur Rashid, Mike Green, and Kelvin Guu. 2022. Dialog inpainting: Turning documents into dialogs. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4558–4586. PMLR.

Nghia Duong-Trung, Xia Wang, and Miloš Kravčík. 2024. BloomLLM: Large Language Models Based Question Generation Combining Supervised Fine-Tuning and Bloom's Taxonomy. In Rafael Ferreira Mello, Nikol Rummel, Ioana Jivet, Gerti Pishtari, and José A. Ruipérez Valiente, editors, *Technology Enhanced Learning for Inclusive and Equitable Quality Education*, volume 15160, pages 93–98. Springer Nature Switzerland, Cham. Series Title: Lecture Notes in Computer Science.

Dennis A. Gioia, Kevin G. Corley, and Aimee L. Hamilton. 2013. Seeking Qualitative Rigor in Inductive Research: Notes on the Gioia Methodology. *Organizational Research Methods*, 16(1):15–31.

Karin K. Hess, Ben S. Jones, Dennis Carlock, and 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*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Akhil Killawala, Igor Khokhlov, and Leon Reznik. 2018. Computational Intelligence Framework for Automatic Quiz Question Generation. In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8, Rio de Janeiro. IEEE.

Unggi Lee, Haewon Jung, Younghoon Jeon, Younghoon Sohn, Wonhee Hwang, Jewoong Moon, and Hyeoncheol Kim. 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.

Subhankar Maity, Aniket Deroy, and Sudeshna Sarkar. 2025. Can large language models meet the challenge of generating school-level questions? *Computers and Education: Artificial Intelligence*, 8:100370.

Gary R. Morrison, Steven M. Ross, Jennifer R. Morrison, and Howard K. Kalman. 2019. *Designing effective instruction*, eighth edition edition. Wiley, Hoboken, NJ.

Huy A. Nguyen, Shravya Bhat, Steven Moore, Norman Bier, and John Stamper. 2022. Towards Generalized Methods for Automatic Question Generation in Educational Domains. In Isabel Hilliger, Pedro J. Muñoz-Merino, Tinne De Laet, Alejandro Ortega-Arranz, and Tracie Farrell, editors, *Educating for a New Future: Making Sense of Technology-Enhanced Learning Adoption*, volume 13450, pages 272–284. Springer International Publishing, Cham. Series Title: Lecture Notes in Computer Science.

Kimia Noorbakhsh, Joseph Chandler, Pantea Karimi, Mohammad Alizadeh, and Hari Balakrishnan. 2025. Savaal: Scalable Concept-Driven Question Generation to Enhance Human Learning. *arXiv preprint*. Version Number: 2.

OpenAI. 2025. GPT-5 system card. Technical report, OpenAI. Accessed: 2026-02-20.

Nicy Scaria, Suma Dharani Chenna, and Deepak Subramani. 2024. Automated Educational Question Generation at Different Bloom's Skill Levels Using Large Language Models: Strategies and Evaluation. In Andrew M. Olney, Irene-Angelica Chounta, Zitao Liu, Olga C. Santos, and Ig Ibert Bittencourt, editors, *Artificial Intelligence in Education*, volume 14830, pages 165–179. Springer Nature Switzerland, Cham. Series Title: Lecture Notes in Computer Science.

Kathrin F. Stanger-Hall and Eric H. Chudler. 2012. Multiple-choice exams: An obstacle for higher-level thinking in introductory science classes. *CBE—Life Sciences Education*, 11(3):294–306.

U.S. Department of Education and Office of Educational Technology. 2023. Artificial Intelligence and Future of Teaching and Learning: Insights and Recommendations. Technical report, U.S. Department of Education, Office of Educational Technology, Washington, DC.

Verónica Villarroel, Susan Bloxham, Daniela Bruna, Carola Bruna, and Constanza Herrera-Seda. 2018. Authentic assessment: creating a blueprint for course design. *Assessment & Evaluation in Higher Education*, 43(5):840–854.

Junling Wang, Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, and Mrinmaya Sachan. 2024. Book2Dial: Generating teacher student interactions from textbooks for cost-effective development of educational chatbots. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9707–9731, Bangkok, Thailand. Association for Computational Linguistics.

Norman L. Webb. 2002. Depth-of-Knowledge Levels for Four Content Areas. Technical report, Wisconsin Center for Education Research, Madison, WI.

Yongan Yu, Alexandre Krantz, and Nikki G. Lobczowski. 2025. From Recall to Reasoning: Automated Question Generation for Deeper Math Learning Through Large Language Models. In Alexandra I. Cristea, Erin Walker, Yu Lu, Olga C. Santos, and Seiji Isotani, editors, *Artificial Intelligence in Education*, volume 15881, pages 414–422. Springer Nature Switzerland, Cham. Series Title: Lecture Notes in Computer Science.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

A Concept Extraction System Prompt

Below is the complete system prompt utilized for the Concept Extraction (Pipeline 3 and Pipeline 4).

System Prompt: Concept Extraction

Role: 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.

Disambiguation heuristic (when needed): 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"]

B Webb's DOK System Prompt

Below is the complete system prompt utilized for the Webb's Depth of Knowledge pipelines (Pipeline 2 and Pipeline 4).

System Prompt: Webb's DOK

Role: 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 & 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 & 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:

   ```
   {"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*.**

- Course 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: `{"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):**

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

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
1. 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.


```
{
    "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. Show your steps.",
            "answer": "p_hat_A=500/10,000=0.050; p_hat_B=560/10,000=0.056; Delta=0.056-0.050=0.006. SE(Delta)
                 =0.003168. 95% CI: 0.006 +/- 1.96*0.003168 => (-0.00021, 0.01221)."
        },
        "DOK 3": {
            "question": "A product manager asks whether B beats A at alpha=0.05. Choose the appropriate significance
                 test for H_0: p_A=p_B and justify your choice. Compute the test statistic and decision.",
            "answer": "Use the pooled two-proportion z-test because H_0 assumes equal rates. p_hat=0.053. SE_0
                 =0.0031683. z=0.006/0.0031683 ~ 1.89 => two-sided p ~ 0.058. Decision at alpha=0.05: fail to reject
                 H_0."
        },
        "DOK 4": {
            "question": "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.",
            "answer": "Plan: (1) Metric: risk difference Delta. (2) Peeking: fixed-horizon stop rule. (3) Target 1:1
                 allocation; run over full weekly cycles. (4) Estimate Delta and 95% CI; test H_0 using pooled p_hat;
                  report Delta, CI, p-value, and expected lift."
        }
    }
}
```


2. 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. 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 without changing V_{max}; noncompetitive lower V_{max} without changing K_m. Turnover number $k_{cat} = V_{max}/[E]_T$; 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.


```
{
    "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 uM/min and K_m = 30 uM. Compute the approximate rate
                 and identify the kinetic order for (a) [S] = 3 uM and (b) [S] = 300 uM.",
            "answer": "(a) [S] << K_m -> v ~ (V_max/K_m)[S] = (120/30)*3 = 12 uM/min; first-order. (b) [S] >> K_m ->
                 v ~ V_max = 120 uM/min; zero-order."
        },
        "DOK 3": {
            "question": "An enzyme has V_max=100 and K_m=20. A noncompetitive inhibitor halves V_max to 50 with K_m
                 unchanged. At [S]=200, which choice yields the largest increase in rate: (A) increase [S] fivefold,
                 (B) double [E]_T, or (C) add a competitive inhibitor? Justify.",
            "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. Increasing [E]_T restores V_max. Adding a competitive
                 inhibitor (C) raises 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.",
            "answer": "Plan: (1) Collect initial-rate data v at >=8 substrate concentrations. (2) Fit v = (V_max[S])
                 /(K_m + [S]) by nonlinear regression. (3) Compute k_cat = V_max/[E]_T. (4) Diagnose inhibitor type.
                 (5) Validate with residual plots."
        }
    }
}
```


3. Negative Examples (Filtering Unsuitable Content)

Input (Copyright/Legal): 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.


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


Input (Logistics): 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.


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

C Concept Extraction + Webb's DOK Example Output

The input document outlined a higher-education course on Supply Chain Management (SCM) focusing on planning horizons, demand forecasting methods, and capacity management. It details technical execution processes including Sales and Operations Planning (S&OP), Bill of Materials (BoM) explosion, and the algorithmic logic of Material Requirements Planning (MRP). Full input examples are left out due to length. The following output is based on one extracted concept from the document:

Example Output: CE + Webb's DOK

DOK 1 Define Sales & Operations Planning (S&OP) in one sentence. Your definition should state who developed it, its primary purpose, the typical planning horizon, how often it is reviewed, and what it aligns operational plans to.

DOK 2 Organize the following activities into the S&OP governance functions. The S&OP governance functions are: Demand (Sales & Marketing), Supply (Operations & SCM), New Activities (Product Management), Finance. For each activity indicate which function owns it:
1. Produce a 12-month rolling demand forecast in units.
2. Set monthly production quantities and plant-level output to meet the plan.
3. Prioritize and schedule resources required for a new product launch.
4. Translate the agreed volume plan into monthly revenue and budgeted costs in currency.

DOK 3 You are the S&OP lead at a mid-sized manufacturer. Sales proposes an optimistic 18-month ramp; Finance insists on a conservative volume for the budget; Operations warns that any capacity expansion requires a 12-month lead time. Using S&OP principles — it balances demand and supply, is an executive decision-making process, covers a mid-term horizon up to 24 months and is reviewed monthly, and aligns operational plans to financial plans (volumes in units and currency) — choose between these options and justify your recommendation:
1. Adopt Finance's conservative budget now and set supply to that lower volume.
2. Adopt Sales' optimistic forecast and initiate capacity expansion immediately.
3. Run a cross-functional reconciliation to produce a consensus S&OP plan, surface remaining trade-offs and risks to the executive S&OP meeting for a decision, and update monthly as data arrives.

Which option do you choose? Provide (1) your chosen option and rationale, (2) three concrete S&OP steps you will execute in the next month to reconcile the views, and (3) the key data you will bring to the executive meeting to enable a clear decision.

DOK 4 Design and justify a 12-month implementation plan to establish a formal S&OP process at a mid-size manufacturer that currently plans in functional silos. The S&OP must balance demand and supply, function as executive decision-making, and align operational plans (volumes in units) with financial plans (currency) over a mid-term horizon (up to 24 months). Your plan must specify:
1. Governance and roles (who owns S&OP, who attends the executive S&OP meeting).
2. A month-by-month rollout timeline with the main milestones for months 1–12.
3. The monthly meeting cadence (types of meetings, their purpose, typical inputs and outputs).
4. Minimum data and deliverables required each month to align volumes to financials.
5. Five KPIs to evaluate S&OP performance and how often you will review them.
6. Three likely implementation risks and your mitigation measures.

Justify how your design applies the core S&OP principles (executive decision-making, balance demand/supply, mid-term horizon, monthly review, alignment to financials).