LEVERAGING LARGE LANGUAGE MODELS FOR THE AUTOMATED GENERATION OF ASSESSMENT ITEMS BASED ON SOLO TAXONOMY

Uroš Petrašković[1],
[0009-0005-5193-5089]

Goran Savić[1]*,
[0000-0002-3917-5487]

Mihaela Osmajić[1],
[0009-0003-9712-1756]

Milan Segedinac[1],
[0000-0003-1743-9522]

Peter Steiner[2]
[0009-0001-5491-9464]

[1]University of Novi Sad,
Faculty of Technical Sciences,
Novi Sad, Serbia

[2]Pädagogischen Hochschule St.Gallen,
St.Gallen, Switzerland

Abstract:

Creating high-quality educational assessment items manually is a labor-intensive and cognitively demanding process for educators. This paper introduces an automated system for generating assessment items by integrating the Structure of Observed Learning Outcomes (SOLO) taxonomy with ontology-based knowledge representation. The system automatically extracts hierarchical course structures from learning materials. By leveraging Large Language Models (LLMs), it generates a comprehensive pool of questions that span various cognitive depths, from unistructural to extended abstract, as defined by the SOLO framework. The technical architecture of the system alongside a qualitative pilot evaluation of the generated questions.

Keywords:

Automatic Question Generation, SOLO Taxonomy, Ontology, LLM.

INTRODUCTION

Education has always played a fundamental role in society and as technology becomes an increasingly integral part of our lives, there is a growing need for tools that can assist teachers and students. One of the challenges teachers face is the creation assessment items. Developing effective assessment items, especially those designed to test different levels of student understanding, is a time-consuming process.

This contribution addresses this challenge by presenting an automated system developed to transform educational content, i.e. lecture notes in PDF format, into assessment items. Unlike generic item generators, this system categorizes items according to the Structure of Observed Learning Outcome (SOLO) taxonomy [1]. This framework allows for the generation of questions at varying cognitive levels, enabling a personalized learning experience where quizzes are adapted to a student's specific depth of understanding. To represent underlaying knowledge structure, the proposed system generates questions based on the underlying course structure, which is hierarchically organized into lessons, sections, and learning objects.

Correspondence:
Goran Savić

e-mail:
savicg@uns.ac.rs

DOI: 10.15308/Sinteza-2026-159-164

Specific educational material is linked to these learning objects and combined with the course structure to produce assessment items. To enable more efficient reasoning over this structure, the system stores information about the course hierarchy in the form of an ontology [2], which formally describes the relationships between the different parts of the course.

2. THEORETICAL BACKGROUND

The SOLO taxonomy, provides a framework for categorizing the increasing complexity of a learner's understanding. It consists of five distinct levels: (1) prestructural, where the student has not yet acquired understanding of the topic and responses are largely irrelevant or missed entirely; (2) unistructural, where they understand one aspect; (3) multistructural, where they understand several aspects but do not see how they connect; (4) relational, where they can integrate different pieces into a coherent whole; and (5) extended abstract, where they can generalize and apply the knowledge to new situations.

The field of automatic item generation has evolved significantly over the past two decades. One of the foundational works was by Mitkov and Ha (2003) [3], who developed a system that used natural language processing rules to transform declarative sentences into questions. Their approach could take a sentence like "Photosynthesis occurs in chloroplasts" and generate "Where does photosynthesis occur?" While effective for simple factual content, these rule-based systems struggled with complex material and often produced grammatically awkward questions.

More recently, Kasneci et al. (2023) [4] published a comprehensive review of using LLMs like ChatGPT in education. They highlighted both the potential and the risks, noting that while these models can generate diverse questions quickly, they also raise concerns about accuracy, bias, and the need for teacher oversight. Lister et al. (2006) [5] examined how the SOLO taxonomy could be applied to classify and evaluate programming questions. Their seminal work, "Not seeing the forest for the trees: novice programmers and the SOLO taxonomy" revealed a critical insight: many assessments that test programming skills focus too heavily on low-level skills (e.g., identifying syntax errors, recognizing code fragments) while neglecting higher-order thinking skills (e.g., integrating concepts, designing solutions). They demonstrated that assessment items could be systematically classified into SOLO levels, from uni-structural questions that test single concepts like "What does this variable store?" to extended abstract questions like "Design a program that solves this complex problem using multiple data structures." Their framework provides a structured way to ensure that items across all complexity levels are included.

One important aspect in automatic generation of multiple-choice items in particular is the quality of the provided wrong answers which are called distractors. The challenge of generating good distractors has been analyzed in Liang et al. (2018) [6], who proposed using a "learning to rank" approach where the system generates many candidate distractors and then ranks them by how plausible they are. Their key insight was that good distractors should be semantically related to the correct answer but clearly wrong upon careful reading. For example, if the correct answer is "mitochondria," good distractors might be "ribosomes" or "nucleus" rather than other less related phrases.

Inspired by the theory presented, a system was developed that utilizes an LLM to generate assessment items of varying cognitive complexity, conditioned on the specific scope of the course structure provided as LLM context during the generation process. In the following sections, this systems architecutre and implementation is detailed, describing the describing the technical solutions developed for this project. Furthermore the generated assessment items will be manually evaluated in the discussion section.

3. THE ASSESSMENT GENERATION SYSTEM

The system is implemented as a web application consisting of a three-tier monolithic architecture. Its primary function is the automated creation of educational assessments based on the SOLO taxonomy, achieved through a defined data processing pipeline that transforms raw materials into structured quizzes (Figure 1).

The process begins with the ingestion of course learning material in PDF format, after which the text is sent to the LLM model for parsing into logical components. While our architecture is model-agnostic, for the purpose of this research we used local Qwen 2.5 model with 14 billion parameters, this model can be run on the local machine without needing an internet connection or API keys. The model automatically extracts course structure from raw course material and stores it into a database as a hierarchical structure: a course contains lessons, lessons contain sections and sections contain specific learning objects. Once all learning objects are created, the system automatically establishes ontological relationships between them, which can then be exported as an OWL file.

The system iterates through the stored learning objects and ontological relationships to generate assessment items covering all four SOLO levels. Each level corresponds to a different type of item. Unistructural questions ask about a single fact. Multistructural items ask about multiple facts. Relational questions ask how concepts connect and extended abstract items ask students to apply what they learned in a new context. Each level is achieved by applying level-specific prompts, adapted to the required cognitive complexity. Every generated item is stored in the database along with its multiple-choice options, the correct answer and a detailed explanation.

The resulting item pool serves as the foundation for automated assessment construction. To generate an assessment, the educator selects the desired topics and specifies the required number of questions for each SOLO level (Figure 2).

Lessons in the upper part of Figure 2 were automatically generated from raw teaching materials on the Serbian language. Figure 3 illustrates a sample quiz as it appears within the system's end-user interface.

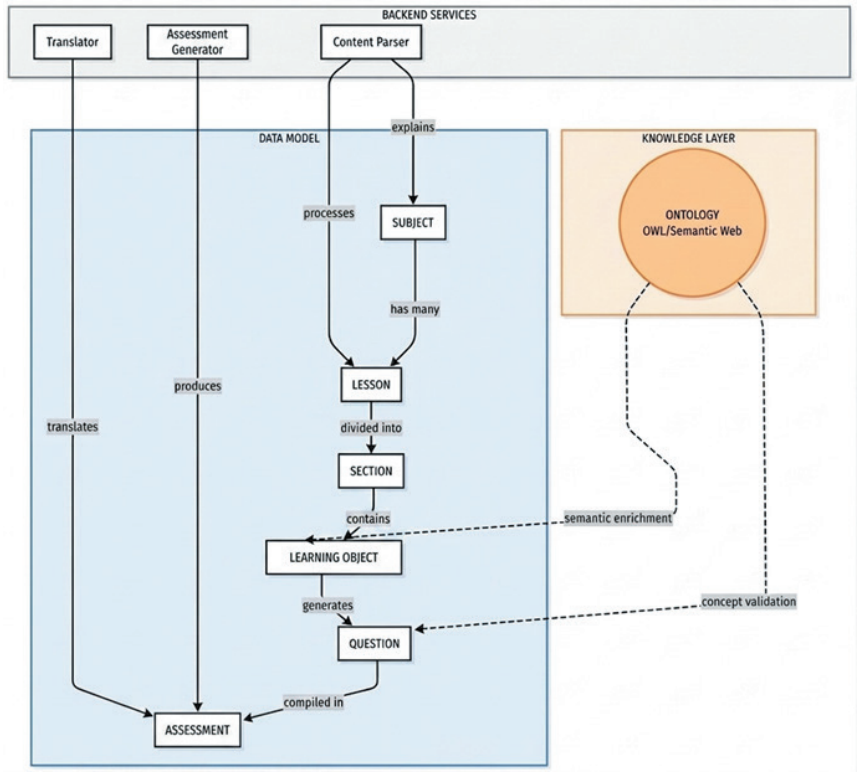

Figure 1. Workflow diagram

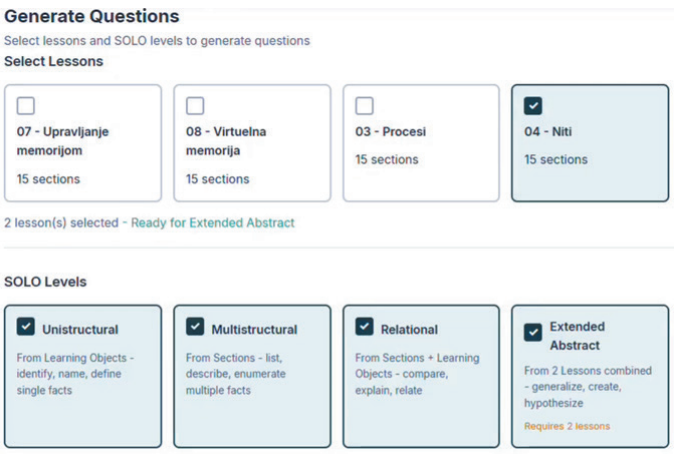

Figure 2. Setting parameters for an assessment

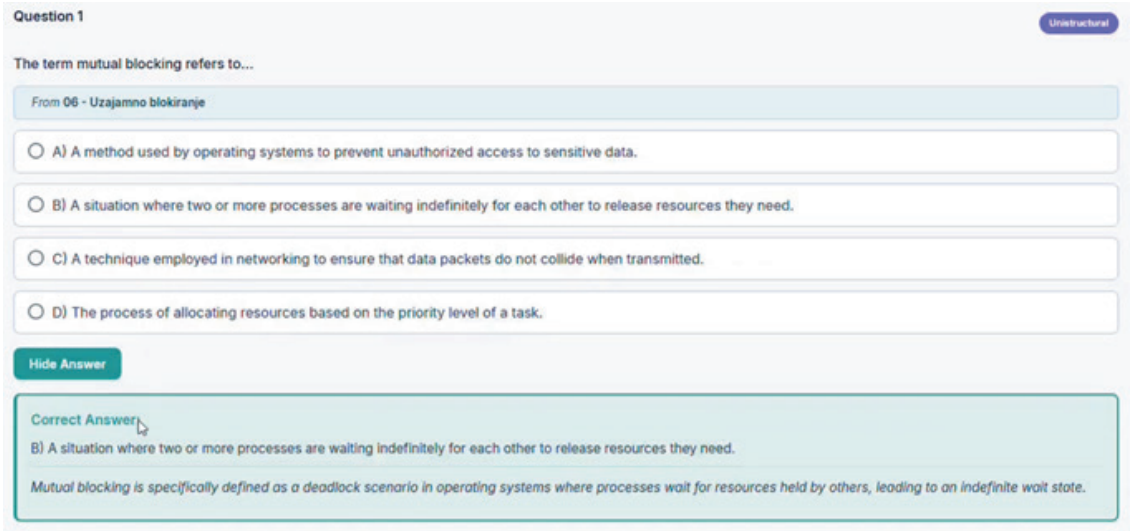

Figure 3. An educational assessment item in accordance with SOLO taxonomy

4. SYSTEM EVALUATION

In this section we discuss the quality of generated items by analysing one random example for each SOLO level. As a pilot study, learning material from two undergraduate courses was taken: operating systems, and software testing. Through the evaluation of the assessment generated automatically by the presented system first evidences regarding the following research questions was generated:

- RQ1: Do the automatically generated assessment items align with their assigned SOLO level?
- RQ2: Are the automatically generated distractors plausible?

4.1. UNISTRUCTURAL QUESTIONS

Items generated at the unistructural level generally succeeded in isolating single, fundamental concepts, which is the primary requirement for this taxonomic stage. To evaluate the generation quality for this category of questions, we examined a representative question: "The Program Counter (PC) holds the memory address of which type of entity? ". The correct answer is A) The next instruction to be executed.

The evaluation of this item's distractors revealed variable quality of our system. Option B (The current executing instruction) proved particularly effective by exploiting a temporal confusion between "next" and "current," accurately targeting students with an incomplete understanding of the fetch-execute cycle. Option C (All instructions in memory) addressed a common novice misconception regarding register scope, although it may be obvious for students aware that registers hold singular values. Option D (Data variables) was identified as the weakest distractor, as students with basic architectural knowledge could easily distinguish between instruction and data addresses, leading to its rapid elimination.

4.2. MULTISTRUCTURAL QUESTIONS

Items at the multistructural level require students to handle multiple independent aspects of a topic, typically through comparison or the identification of several correct characteristics. As a representative example, we analyse the question: "Which of the following statements correctly describe both User-Level Threads (ULT) and Kernel-Level Threads (KLT)?" with the correct answer being: A) ULT are managed by application, KLT by OS; ULT can provide finer control but may be less efficient than KLT.

The evaluation of this item's distractors showed a successful attempt at targeting complex misconceptions. Option B (Both ULT and KLT require explicit synchronization mechanisms to prevent race conditions) appeared highly plausible as synchronization is a genuinely relevant topic in threading. However, it incorrectly universalizes a concern that varies significantly by implementation. Option D (Kernel manages both ULT and KLT, providing equal efficiency and control) effectively targeted the misconception that kernel-level management automatically equalizes all thread characteristics. Option C (ULT provides more system resources allocation units compared to KLT) can be classified as

the weakest distractor. Its phrasing can be seen as too technical and sufficiently vague that students might eliminate it based on unclear wording rather than a true understanding of thread resource allorcation.

4.3. RELATIONAL QUESTIONS

Items at the relational level require students to have deeper understanding, to compare and connect concepts. As a representative example we will take the question "How does test automation contribute to the effectiveness of regression testing?" The generated options were: A) Test automation only improves the speed of functional testing without impacting regression testing, B) Test automation replaces manual testing entirely, making regression testing unnecessary, C) Test automation allows for faster execution and identification of changes in functionality, enhancing the thoroughness of regression tests, D) Test automation focuses on creating new test cases rather than running existing ones efficiently. The correct answer is C.

The question requires students to connect two concepts - understanding both what regression testing entails (re-executing existing tests to detect unintended changes) and how automation specifically addresses its challenges and can therefore be calssified as relational. With regard to distractors, option B captures an extreme misconception about automation replacing human judgment entirely, while option A creates an artificial boundary between functional and regression testing that might appeal to students with lesser understanding. However, regarding the option D, the claim that "test automation focuses on creating new test cases" doesn't align with any common misconception about automation - most students would recognize that automation executes tests rather than generates them. This makes the distractor too easily eliminated. Additionally, Option A's specificity about "only functional testing" may be overly obvious, as students familiar with basic automation concepts would recognize that automated tests can serve multiple purposes. Stronger distractors might address misconceptions about automation coverage limitations or the false belief that automated tests detect different types of defects than manual regression testing.

4.4. EXTENDED ABSTRACT QUESTIONS

Questions at the extended abstract level are designed to test the student's ability to generalize and apply learned concepts to new, more complex domains.

One representative example is the question "In a distributed computing environment, if a main process spawns multiple child processes across different nodes, what mechanism ensures that all child processes receive the correct system resources and do not interfere with each other?". The correct answer is A) Implementing inter-process communication (IPC) protocols to manage resource allocation and synchronization.

The evaluation of this question shows that it successfully takes the concept of process management, typically learned in a single-machine context and requires the student to extend it to a distributed environment. The distractors exhibited a mix of effectiveness in targeting reasoning errors. Option C (Allocating all child processes to a single node for centralized management of resources) effectively identifies a common misconception where students assume that centralization is a viable solution for management, despite it defeating the purpose of distribution. Option D (Using thread-level synchronization techniques within each individual node) correctly recognizes a partial truth, as thread synchronization addresses only intra-node concerns rather than the broader distributed problem. Option B (Increasing the priority of the parent process over its children to control their execution flow) was identified as too obviously incorrect. The idea of "increasing priority" does not logically connect to resource allocation or interference prevention across nodes, making it easily eliminated and reducing the overall challenge of the item.

4.5. DISCUSSION

The evaluation revealed several systematic limitations in the question generation process. A clear pattern emerged across cognitive levels: as the taxonomic complexity increased from unistructural to extended abstract, the item quality decreased. Lower-level questions (unistructural and multistructural) generally demonstrated stronger alignment with their intended cognitive levels. However, higher-level questions, particularly at the relational and extended abstract levels, exhibited significant validity concerns. The correct answers themselves sometimes contained technical inaccuracies or oversimplifications.

Distractor quality also declined at higher cognitive levels. While unistructural items typically featured distractors targeting genuine misconceptions, extended abstract items often included options that were either trivially eliminable (violating fundamental principles obvious to any competent student) or failed to reflect

realistic reasoning errors students might make when extending concepts to new domains. Additionally, distractor redundancy emerged as a recurring issue, with multiple options sometimes conveying essentially the same incorrect concept, reducing the discriminatory power of the assessment items.

5. CONCLUSION

This paper presented a system for the automated generation of educational assessments grounded in the SOLO taxonomy. By integrating Large Language Models (LLMs) with Semantic Web technologies and an intuitive user interface, the system enables educators to develop assessments automatically. Preliminary results indicate that the lower level questions (unistructural and multistructural) generally demonstrated stronger alignment with their intended cognitive levels, and they feature distractors sufficiently plausible to evaluate genuine conceptual understanding. While higher cognitive levels present challenges regarding question validity and distractor variety, the overall performance remains consistent. Nonetheless, certain limitations were identified, particularly concerning the nuance and variety of generated incorrect options.

Future work will primarily focus on a comprehensive evaluation of the system's quality. Such an evaluation would provide empirical insight into whether the system reliably produces pedagogically sound questions across all levels and diverse learning materials. A key goal within this evaluation effort is to identify the underlying causes of quality degradation observed at higher cognitive levels and to systematically improve the system's performance in those areas. Additionally, future work will refine distractor generation to ensure higher pedagogical utility, and will explore extending the system's capabilities beyond multiple-choice questions to include formats such as short-answer and essay prompts.

6. ACKNOWLEDGMENTS

This research has been supported by the Ministry of Science, Technological Development and Innovation (Contract No. 451-03-34/2026-03/200156) and the Faculty of Technical Sciences, University of Novi Sad through project “Scientific and Artistic Research Work of Researchers in Teaching and Associate Positions at the Faculty of Technical Sciences, University of Novi Sad 2026” (No. 01-3609/1).

REFERENCES

[1] C.C. Chan, M.S. Tsui, M.Y. Chan and J.H. Hong, "Applying the structure of the observed learning outcomes (SOLO) taxonomy on student's learning outcomes: An empirical study. Assessment & Evaluation in Higher Education", vol 27, no 6, pp. 511-527, 2002.

[2] B. Smith, "Ontology", 2012.

[3] R. Mitkov, "Computer-aided generation of multiple-choice tests," in Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing, pp. 17-22, 2003.

[4] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, D. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier and S. Krusche, "ChatGPT for good? On opportunities and challenges of large language models for education. Learning and individual differences", vol 103, p. 102274, 2023.

[5] R. Lister, B. Simon, E. Thompson, J.L. Whalley and C. Prasad, "Not seeing the forest for the trees: novice programmers and the SOLO taxonomy", in ACM SIGCSE Bulletin, vol. 38, no. 3, pp. 118-122, 2006.

[6] C. Liang, X. Yang, N. Dave, D. Wham, B. Pursel and C.L. Giles, "Distractor generation for multiple choice questions using learning to rank," in Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications, pp. 284-290, June 2018.