ISSN 2723-6471

Ontology-Driven Adaptive Learning Environment Using Large Language Models for Educational Knowledge Extraction

Rakhila Turebayeva[1], Bulat Kubekov[2], Yenglik Kadyr[3,*], Umut Turusbekova[4], Aizhan Nazyrova[5,*], Zhanar Lamasheva[6,*]

[1,3,4,5,6]*Institute of Digital Sciences and Artificial Intelligence, L.N. Gumilyov Eurasian National University, 2 Satpayev Str., Astana 010008, Kazakhstan*

[2,3]*Institute of Automation and Information Technology, Satbayev University, 22 Satpayev Str., Almaty 050013, Kazakhstan*

(Received: February 25, 2026; Revised: May 1, 2026; Accepted: July 22, 2026; Available online: August 18, 2026)

Abstract

Enriching educational ontologies automatically from low-resource-language text remains an unsolved integration problem: conventional large language model (LLM)-to-knowledge-graph pipelines require a vocabulary-alignment step and lack hallucination control at the commit boundary. This study removes both bottlenecks and tests whether the resulting ontology can drive measurable learning gains. The core idea is schema co-design: the JSON schema constraining LLM output *is* the OWL T-box of the target ontology, so extracted records are directly populatable and no alignment step is needed; a literal-presence filter rejects entities absent from the source text before commit, the HermiT reasoner verifies consistency, and Kazakh, Russian and English labels are generated in a single follow-up call. The pipeline (Python 3.9, OWLready2) was benchmarked on 90 Kazakh paragraphs across GPT-4o, Claude 3.5 Sonnet and Gemini 1.5 Pro, and the resulting adaptive textbook was evaluated quasi-experimentally with 65 Grade 6 students (control n = 32, experimental n = 33) in two Astana schools over 16 weeks. Of 312 source chunks, 271 (86.9%) survived the full validation chain, yielding 1,847 OWL individuals, 2,931 object-property assertions and 6,512 data-property annotations across 47 classes; the literal-presence filter intercepted 5.2% of validated responses and the reasoner a further 1.1%. Extraction F1 reached 0.89 (GPT-4o), 0.87 (Claude) and 0.86 (Gemini), with hallucination rates of 4.9–8.3%; the one-time corpus build cost USD 3.47–9.64, with zero marginal LLM cost per learner. The experimental group outperformed controls on task accuracy (+19.4 points, d = 2.30), repeated errors (−53.6%, d = 1.85) and sessions to mastery (−34.3%, d = 1.50), all $p < 0.001$ under Bonferroni correction. The novelty lies in making the ontology T-box itself the extraction schema, combined with pre-commit literal grounding, validated in a real low-resource classroom deployment.

Keywords: Ontology Enrichment, Large Language Models, Schema-Constrained Extraction, Hallucination Mitigation, Adaptive Learning, Kazakh NLP

1. Introduction

The use of Artificial Intelligence (AI) tools in digital educational environments is widely considered a key direction for the development of automated learning support. International reports emphasize that AI can be used to analyze educational data, generate recommendations, and support students throughout the learning process [1], [2]. However, the practical implementation of these opportunities remains limited, in particular because of the lack of sustainable mechanisms for individualizing learning trajectories and scaling the analysis of academic work. According to the World Bank, these limitations are associated with persistent differences in educational outcomes and the problem of learning poverty [3].

Research pays special attention to the interpretability and regulatory compliance of AI systems in education. Reviews of explainable AI emphasize the need for transparency in model outputs and the ability to relate automated analysis results to learning objectives [4]. The European Commission's ethical guidelines further require that AI outputs be linked to formal learning outcomes and pedagogical goals, particularly when such systems are used for student assessment and support [5]. Meeting these requirements demands the integration of AI components with formal

***Corresponding authors: Yenglik Kadyr (kadyr.y@stud.satbayev.university), Aizhan Nazyrova (nazyrova_aye_1@enu.kz), Zhanar Lamasheva (lamasheva_zhb@enu.kz)**
DOI: https://doi.org/10.47738/jads.v7i3.1456

knowledge representation models that can make the link between model outputs and curricular concepts explicit. Ontologies are one of the established mechanisms for such formalization. Ontological models capture educational concepts, skills, and the relationships between them, providing a consistent representation of educational content and a basis for personalizing learning [6]. In parallel, large language models (LLMs) have been increasingly applied to ontology engineering, including automatic knowledge extraction and structuring [7].

The field of personalized learning is developing alongside these advances. Approaches to adapting learning tasks and recommendations to individual learner characteristics are being explored, including those based on LLMs [8]. Related work addresses automated generation of pedagogical interventions [9], models for assessing and supporting learners in AI-enhanced environments [10], and learners' perceptions of personalized solutions [11]. Analytical reviews confirm the sustained interest in this area and emphasize the need for further methodological development [12], [13], [14]. Despite this body of work, the integration of LLM-based text processing results into domain-specific ontologies remains insufficiently addressed. In many existing systems, feedback is delivered in textual form and is not converted into a structured representation aligned with learning concepts, skills, and error categories, which limits the use of such solutions in adaptive learning environments that require a formal connection between learner responses and personalization models.

This paper proposes an approach that uses LLMs for the automatic enrichment of a personalized learning ontology. The contributions of this work, and the points that distinguish it from neighbouring approaches to converting unstructured text into machine-usable knowledge, are the following:

(i) A schema co-designed with the target ontology. The JSON schema used to constrain LLM output is structurally identical to the OWL T-box of the subject ontology. Because the extraction vocabulary and the ontology vocabulary coincide by construction, the separate vocabulary-alignment step required when an extraction-specific schema must be mapped onto the target ontology is obviated within the pipeline. We do not claim that alignment accounts for all integration error; rather, it is reported in prior work on LLM-to-knowledge-graph construction as a recurrent and substantial source of such error [7], [14], and the present design removes this particular failure mode by construction rather than repairing it after extraction.

(ii) A literal-presence hallucination filter as a pipeline stage. Every extracted surface form must appear verbatim in the source text. The constraint is enforced by an external validator, not by prompt instructions alone; hallucinated entities are dropped and logged. The resulting hallucination rate is reported as part of the comparative error analysis in Section 4.

(iii) Single-call multilingual labelling for a low-resource language. For every OWL individual created by the pipeline, Kazakh, Russian, and English labels are produced through a translation sub-task in the same LLM call. This removes the dependence on parallel corpora that limits annotation-based approaches for Kazakh. Because the labels are produced by the LLM rather than retrieved from aligned resources, the mechanism is in principle language-agnostic; however, all experiments reported in this paper use Kazakh, so its transfer to other low-resource languages is an expectation grounded in the design rather than an empirically demonstrated result (see Section 5.5).

(iv) Joint extraction of subject-matter content and pedagogical error categories under a single schema. The same pipeline populates the linguistic component of the ontology from textbook material and the error-category component from student responses; the two share the OWL T-box. This shared schema is what makes ontology-driven adaptive feedback possible in the system evaluated in Sections 3 and 4.

Section 2 positions these contributions against three established research directions - retrieval-augmented generation, text-level semantic annotation, and open-domain knowledge graph extraction - and surveys the related literature on LLMs in education, personalized learning, and ontology-based knowledge representation. Section 3 describes the methodology. Section 4 reports the experimental evaluation. The framework is evaluated in a digital textbook environment for Grade 6 students of Kazakh language, Mathematics, and Computer Science; experimental results show a 19.4% improvement in task accuracy and a 53.6% reduction in repeated errors relative to a non-ontology baseline. The methods themselves are not domain-specific and can be adapted to other areas that require structured knowledge extraction and rule-based decision support.

Based on the issues outlined above, this study addresses three main research questions concerning the design and effectiveness of an ontology-oriented digital textbook enriched with large language models (LLMs). RQ1, it examines how a subject ontology, automatically enriched using LLMs, can be used to structure a digital textbook, including its curricular sections, concepts, learning tasks, and assessment elements. RQ2, it investigates how many platform sessions are required for learners to reach the module mastery threshold when using the ontology-oriented digital textbook and compares this with a non-ontology digital textbook covering the same learning material. RQ3, the study evaluates the effect of ontology-based assessment and feedback on learners' task accuracy, the number of repeated errors per module, and the number of sessions required to achieve mastery, in comparison with the same digital textbook format without ontology-driven adaptation.

The remainder of the paper is organized as follows. Section 2 reviews related work and positions this paper against three methodological neighbours. Section 3 describes the methodology, including the ontology construction pipeline, the LLM benchmarking procedure, and the pedagogical experiment. Section 4 presents the results of the digital textbook deployment and the comparative evaluation of LLMs. Section 5 discusses the findings in relation to the research questions. Section 6 concludes the paper and outlines directions for future work.

2. Literature Review

2.1. Review protocol and bibliometric context

The literature relevant to this work spans four areas that frame the contribution stated in Section 1: review procedure and bibliometric context (Section 2.1); LLMs for learner support and feedback (Section 2.2); personalized and adaptive learning (Section 2.3); and ontology-based knowledge representation in education (Section 2.4). Section 2.5 then positions the proposed approach against three methodological neighbours that are frequently conflated with it in the recent literature: retrieval-augmented generation, text-level semantic annotation, and open-domain knowledge graph extraction.

The review followed the PRISMA 2020 guidelines [15], which govern the procedures for identifying, selecting, and analyzing relevant studies. Figure 1 presents the publication selection process, reflecting the transition from the initial set of sources to the final sample of 38 included studies. This figure 1 of 38 refers to the primary studies retained through the PRISMA selection process and is not the size of the reference list. The reference list is larger because it also cites sources that lie outside the systematic-review scope and were therefore not subject to the PRISMA inclusion criteria – for example, international policy and standards documents ([1], [2], [3], [5]), the PRISMA 2020 reporting guideline itself [15], foundational references on ontology engineering ([16], [17]), and the technical report of the language model used in the pipeline [18], together with additional works cited for background and positioning in the Introduction and the Discussion. These contextual, methodological, and technical citations are counted separately from the 38 studies included in the systematic review.

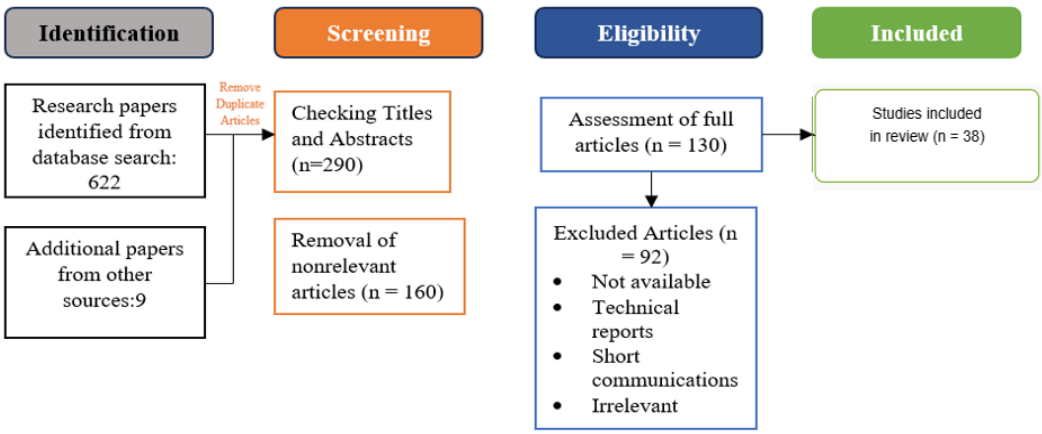

Figure 1. PRISMA 2020 flow diagram of the study selection process.

 ISSN 2723-6471

A bibliometric snapshot of the field complements the systematic selection. Figure 2 summarizes Scopus publications from 2021–2025 on personalized learning and LLMs in Computer Science (snapshot December 2025): 622 documents distributed across 135 venues, an average annual growth rate of 129.87%, 1,904 unique authors, an average of 3.81 co-authors per paper, 1,543 unique author keywords, and an average of 4.875 citations per article. These indicators describe a young but rapidly consolidating research area.

Figure 2. Bibliometric indicators of the sample of publications on personalized learning and large language models (2021–2025).

To support replication of this snapshot, the search specification is reported in full. The records were retrieved from Scopus in December 2025 using a single title-abstract-keyword query: TITLE-ABS-KEY (("personalized learning" OR "personalised learning" OR "adaptive learning") AND ("large language model*" OR "LLM" OR "generative AI" OR "GPT*" OR "ChatGPT")) AND PUBYEAR > 2020 AND PUBYEAR < 2026 AND (LIMIT-TO (SUBJAREA , "COMP")) AND (LIMIT-TO (LANGUAGE , "English")). The two inclusion facets joined by AND were a personalized- or adaptive-learning facet and a large-language-model facet, each expanded with the spelling and acronym variants shown above. A record was included if it fell within the 2021–2025 window, was indexed under the Computer Science (COMP) subject area, was written in English, and was of document type article, conference paper, review, or book chapter. A record was excluded if it was an editorial, erratum, note, or retracted item, was a duplicate, or showed no substantive link to both query facets in its title, abstract, or keywords. Applying these criteria returned the 622 documents from 135 sources reported above. The summary indicators were computed from the exported records with the bibliometrix package in R (Biblioshiny interface), and the author-keyword co-occurrence map in figure 3 was generated in VOSviewer from the same record set.

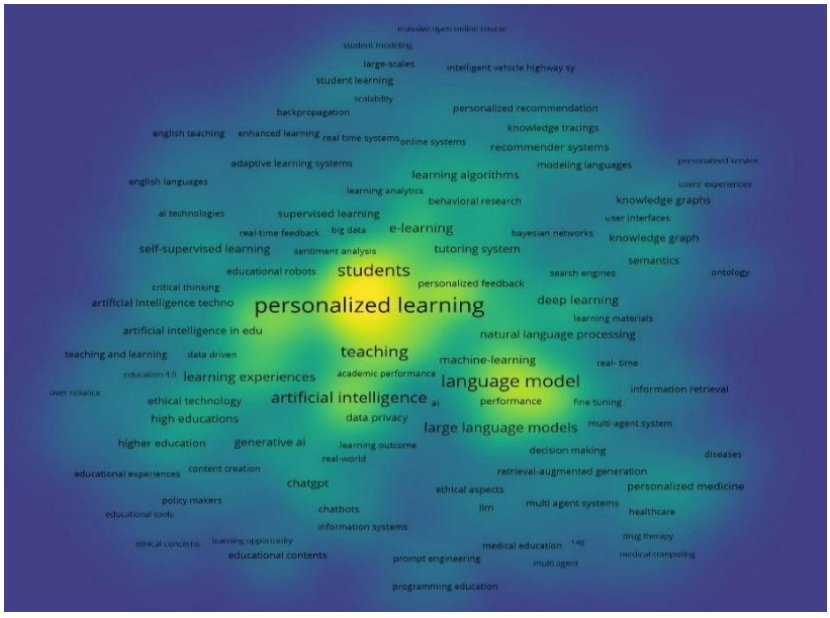

Figure 3. Co-occurrence map of keywords on personalized learning and large language models (VOSviewer).

Figure 3 shows the co-occurrence map of author keywords. The central terms are “personalized learning”, “artificial intelligence”, “students”, “teaching”, “language model”, and “large language models”, indicating the integration of AI

methods with personalized learning objectives. Clusters around this core capture machine learning methods, natural language technologies, pedagogical aspects, and a distinct cluster on ethics and data protection, which reflects the relevance of safety concerns when deploying AI in education.

2.2. LLMs for learner support and feedback

Large language models are increasingly used in educational contexts to analyze learner responses, generate feedback, and support the learning process [19]. Approaches based on sequential prompt chaining improve the manageability and predictability of model outputs, making them suitable for error analysis and structured feedback generation [20]. Learner-focused work considers generative models as support tools for independent study, writing assignments, and problem solving in a range of subjects [21], [22], [23], [24]. Several authors emphasize the need for interpretability and validation of LLM outputs, particularly in educational scenarios where automated feedback can influence learning outcomes [22], [25]. LLMs have been used as auxiliary tools in language teaching and assessment, where they are applied to formative feedback generation and to the analysis of student work [26], [27]. Studies in engineering and medical education describe similar uses while emphasizing the need to control the accuracy and pedagogical consistency of model outputs [25], [28]. Across these studies, however, the LLM output is consumed as natural language and is not converted into a representation that other components of the learning system can reason over.

2.3. Personalized and adaptive learning

Research on adaptive and AI-supported learning environments shows that automated systems can monitor learning activity and inform pedagogical decisions [11]. A separate line of work proposes the automated generation of personalized pedagogical interventions, drawing on data and models from intelligent tutoring systems [29], [30]. Learner perception of AI-supported personalization has also been studied, with adoption depending on perceived usefulness and usability [31]. Contemporary reviews confirm the sustained interest in personalized learning and treat generative AI as one of the components that may drive its further development [32], [33].

2.4. Ontology-based knowledge representation in education

A separate research direction concerns the ontological representation of knowledge in education. Ontologies formalize educational concepts, skills, and the relationships between them, providing a structured basis for personalizing learning and adapting educational content [34], [16], [17]. Recent work applies LLMs to ontology engineering, including the automatic extraction of linguistic and conceptual information from texts and the subsequent enrichment of ontological models [35], [36], [37]. Work on automated assessment and analysis of educational data shows that current approaches typically fail to link text-processing results to structured learner models that describe concepts, skills, and error types [38], [39]. This limits the use of LLMs in adaptive learning environments in which feedback must be aligned with performance indicators and individual learning trajectories [40]. Studies on learner perceptions further indicate that adoption depends on transparency of the recommendation mechanism and on alignment with learning goals [41]. Review papers on generative AI in education identify the absence of formal mechanisms for integrating LLM outputs with learning models as an open problem that hinders deployment in personalized and adaptive systems [42].

2.5. Positioning against related approaches

The approach proposed in this paper sits between three established research directions that are sometimes conflated in recent work on LLMs and educational knowledge: retrieval-augmented generation (RAG), text-level semantic annotation, and open-domain knowledge graph extraction. Each direction partially addresses the problem of converting unstructured text into machine-usable knowledge, but each has limitations when the requirement is a persistent, reasoner-consistent, domain-aligned ontology that supports adaptive learning. This subsection states the differences explicitly so that the contributions listed in Section 1 can be read against a concrete baseline.

Retrieval-augmented generation. RAG retrieves passages from a corpus at query time and conditions an LLM response on them [20]. The output is natural language; nothing persistent or symbolic is built between queries. RAG is well-suited to question answering, but it does not produce a representation that other, non-LLM components can consume: a description-logic reasoner cannot reason over a retrieval set, an adaptive task-selection module cannot navigate it, and a SPARQL query cannot address it. In the educational setting, the regulatory requirement that automated feedback

ISSN 2723-6471

be linked to formal learning outcomes [5] is not satisfied by retrieved-then-generated text, because the link is recomputed and unobservable at every turn.

This characterization refers to the standard text-passage form of RAG. We acknowledge that GraphRAG and other knowledge-graph-based variants incorporate symbolic structures, narrowing the distinction from ontology-based approaches. Our key distinction is that, in these systems, the graph typically supports retrieval at query time, whereas our pipeline uses a persistent, reasoner-validated OWL ontology as the primary knowledge base, consumed directly by the description-logic reasoner and the SPARQL-based task-selection module, with the LLM used only during ontology construction. Accordingly, table 1 describes the text-passage form of RAG, while graph-enhanced RAG systems share some characteristics of ontology-based approaches.

Text-level semantic annotation. Gazetteer-based named-entity recognition, BRAT- or INCEpTION-style standoff annotation, and supervised entity taggers produce labelled spans on top of the source text. These approaches are limited to what the annotation scheme defines, depend on the availability of an annotated corpus, and live at the text layer rather than as first-class ontology entities. For Kazakh, annotated educational corpora are scarce, which sets a hard ceiling on supervised approaches. Annotations alone cannot be reasoned over, cannot be queried with SPARQL without an additional projection step, and cannot be merged across documents without a separate aggregation procedure.

Open-domain knowledge graph extraction. OpenIE-style systems, neural relation extractors such as REBEL, and recent LLM-based triple-extraction pipelines [35], [36] produce (subject, predicate, object) triples from text. These triples are schema-free or use an extraction-specific vocabulary that does not match the target ontology. To use the extracted triples in a domain ontology, an alignment step is required that maps free-form predicates onto the ontology's properties. This alignment step is reported in recent surveys of LLM-based ontology engineering as a recurrent source of error in production knowledge-graph pipelines [7], [14], and consistency at the description-logic level is rarely checked. Recent surveys of LLMs in ontology engineering [7], [14] explicitly identify alignment and hallucination control as open problems.

The pipeline proposed in this paper. Four operational decisions separate this work from the three directions above. First, the schema given to the LLM is the OWL T-box of the target ontology, so the output is directly populatable and no alignment step is required. Second, hallucinations are filtered by literal-presence checks against the source text, so what is committed to the ontology is grounded in the source. Third, the resulting structure is a persistent OWL ontology whose consistency is checked by a description-logic reasoner before commit, so downstream components can rely on the standard guarantees of a symbolic knowledge base. Fourth, the same pipeline extracts both subject-matter content and pedagogical error categories under a single schema, which is the structural prerequisite for ontology-driven adaptive feedback. These distinctions are summarized in table 1.

Table 1. Positioning of the proposed pipeline against three related approaches to converting unstructured text into machine-usable knowledge.

Aspect	Compared Approaches			
	RAG	***Semantic annotation***	***KG extraction***	***This paper***
Output form	Natural language	Text-level labels	(s, p, o) triples	OWL individuals + assertions
Persistence	Query-time, ephemeral	Per-document	Graph, schema-free	Persistent ontology
Manual labour at build time	Corpus curation	Annotation	Training data / rules	None at extraction
Schema discipline	None	Annotation scheme	Often schema-free	OWL T-box, schema-first
Alignment step needed	N/A	Annotation → ontology	Yes (main error source)	No (schema = T-box)
Hallucination mitigation	Retrieval grounding	N/A	Usually unaddressed	Literal-presence filter
Reasoner-checked consistency	No	No	Rarely	HermiT before commit
Usable by non-LLM systems	No	Limited	SPARQL	SPARQL + DL reasoning
Low-resource language support	LLM monolingual bias	Needs annotated corpus	Needs training data	Works on Kazakh directly

Two clarifications about the scope of table 1 are in order. The comparison concerns capabilities at the level of representation and integration, not at the level of question-answering quality: RAG remains the appropriate tool when the deliverable is a natural-language response to a free-form query, and nothing in this paper argues otherwise. The

comparison also abstracts over the cost of the LLM calls; because the present pipeline uses LLMs at construction time only, its query-time cost differs in kind from that of RAG. The comparison in table 1 is architectural, and this paper does not report an empirical head-to-head benchmark against a RAG, annotation, or open-IE system; the operating cost and latency of the pipeline are likewise not measured here and are noted as a limitation (Section 5.5).

3. Methodology

3.1. Study design

The methodology of this study has three parts: (i) an engineering pipeline that constructs and enriches the subject ontology with the help of an LLM, described in Section 3.2; (ii) a benchmarking procedure that compares the LLMs used in this pipeline, described in Section 3.3; and (iii) a quasi-experimental pedagogical study that evaluates the resulting digital textbook, described in Section 3.4. The independent variable in the pedagogical study is the type of learning environment (a standard digital textbook for the control group versus the ontology-driven adaptive textbook for the experimental group); the dependent variables are task accuracy, the number of repeated errors, and sessions to mastery. The overall software architecture is presented in figure 4. The system is implemented in Python 3.9: it retrieves subject-domain web resources, builds prompts from the extracted text, sends them to the language model through an external API, and stores the resulting structured outputs as ontology objects using the OWLready2 library.

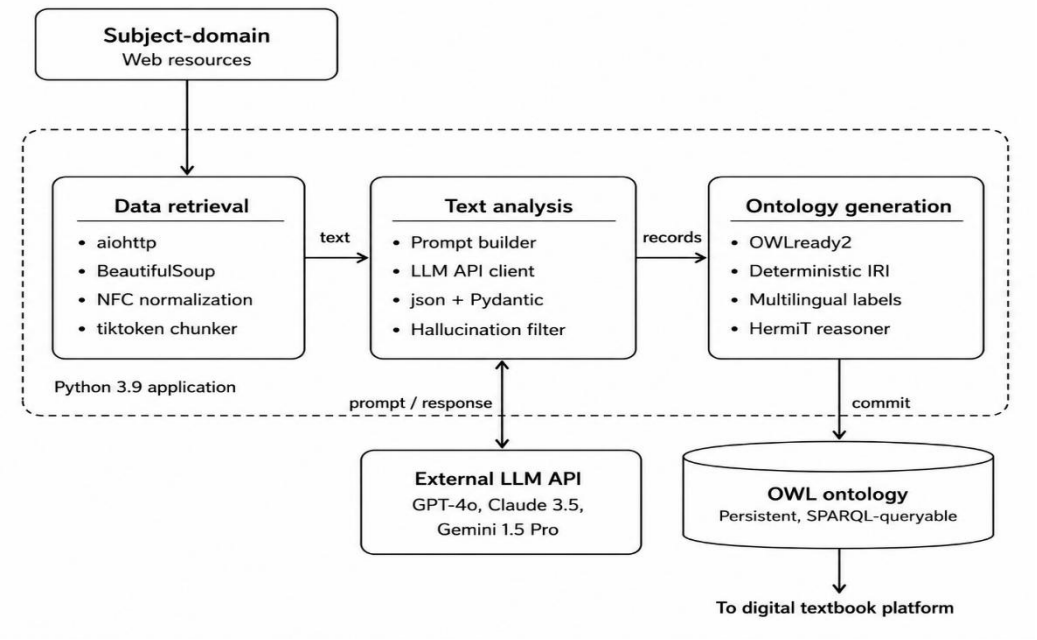

Figure 4. Software architecture for data retrieval, text analysis, and ontology generation.

3.2. LLM-based ontology enrichment pipeline

This section describes, step by step, how raw Kazakh-language text is transformed into elements of the subject ontology through interaction with an LLM. The pipeline is implemented in Python 3.9 and consists of five stages: (i) asynchronous source retrieval and preprocessing; (ii) prompt construction with a fixed output schema; (iii) asynchronous invocation of the LLM API; (iv) parsing and schema validation of the response; and (v) mapping of the validated record to OWL elements using OWLready2. The five stages are illustrated in figure 5 and detailed below. The same pipeline is used both for the primary model (GPT-4o) and for the comparison models reported in Section 3.3. The pipeline extends the role separation introduced in Mukanova et al. (2024, 2025), in which subject-matter experts select web resources and a knowledge engineer transforms the collected data into an ontological representation; the present work integrates an LLM into the extraction step and adds a multilingual labelling step.

 ISSN 2723-6471

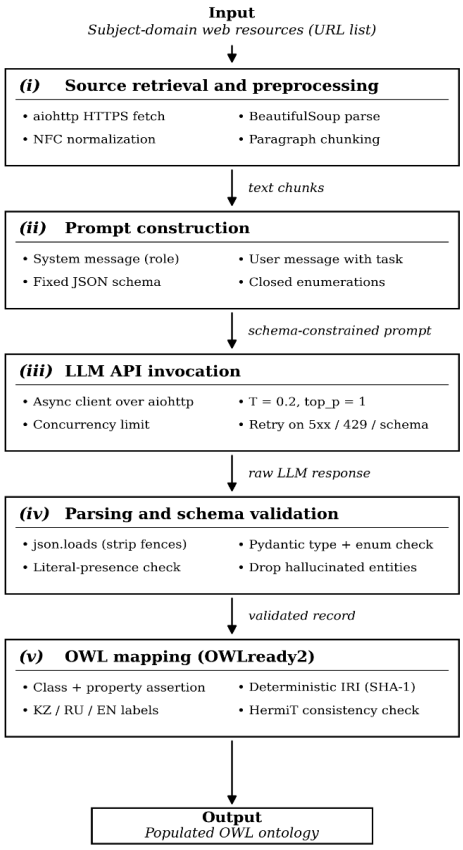

Figure 5. Five-stage pipeline from source text to populated ontology."

3.2.1. Source text retrieval and preprocessing

Source documents are fetched asynchronously over HTTPS with aiohttp. For each URL the response body is parsed with BeautifulSoup. The page title is taken from the <title> element; the main body is obtained by removing non-content elements (<script>, <style>, <nav>, <footer>, <aside>, and ad-class containers) and collecting the text content of the remaining block-level elements. The resulting string is normalized in three ways: consecutive whitespace is collapsed to single spaces, soft hyphens and zero-width characters are removed, and the Unicode form is normalized to NFC. This is the only stage of the pipeline at which any rule-based text processing is applied; from this point on, all linguistic decisions are made by the LLM.

Texts longer than the request budget are chunked at paragraph boundaries. The chunker is greedy: paragraphs are appended to the current chunk until the next paragraph would exceed 3,000 tokens (estimated with the tiktoken cl100k_base encoder), at which point the current chunk is closed and a new one is started. No paragraph is ever split mid-sentence. The output of this stage is a list of (title, chunk_text) tuples; each tuple becomes one LLM request.

3.2.2. Prompt construction

Each LLM request is composed of three main components: a system message that defines the model's role, a user message containing the task description and the input text, and an output schema that constrains the response to a strict JSON structure. The schema serves as the core mechanism enabling downstream systems to process responses without relying on rule-based parsing. Since field names, value types, and closed enumerations are predefined, the parser only needs to validate whether the response conforms to the specified schema. The system message remains fixed across all requests and is defined as follows: "You are a linguistic-extraction assistant. Your only task is to read the given Kazakh-language text and return a structured representation of its linguistic content. Reply ONLY with valid JSON that conforms to the schema given by the user. Do not return commentary, prose, or any text outside the JSON object."

The user message is generated from a predefined template in which only the title and body sections vary between requests. The output schema is designed to represent linguistic elements in a structured format, including

parts_of_speech, grammatical_categories, syntactic_units, usage_examples, and relations. Two design decisions within this template are particularly important. First, the requirement that all surface-form values must appear literally in the input text serves as the operational definition of hallucination used in the benchmark presented in Section 3.3. Any value that does not literally occur in the source text is treated as hallucinated and excluded from further processing. This is a deliberately narrow and conservative operationalization: it targets only ungrounded surface forms and is therefore a precision-oriented grounding check rather than a complete account of hallucination. The check is applied to surface forms after the Unicode (NFC) and whitespace normalization described in Section 3.2.1; because it requires a verbatim occurrence, valid paraphrases, inferred concepts, and normalized dictionary forms - for example, a lemma whose inflected variant, rather than the lemma itself, appears in an agglutinative Kazakh text - can be rejected even when they are linguistically correct. Whether a grounded entity is mapped to the correct class or relation is a separate question, evaluated as the semantic-error category defined in Section 3.3 rather than as hallucination; the consequences of this narrow definition for the interpretation of the reported rates are discussed in Section 5.5. Second, the closed enumerations defined for the pos, type, and relation fields allow direct mapping to a fixed set of OWL classes and object properties without requiring an intermediate alignment stage (figure 6).

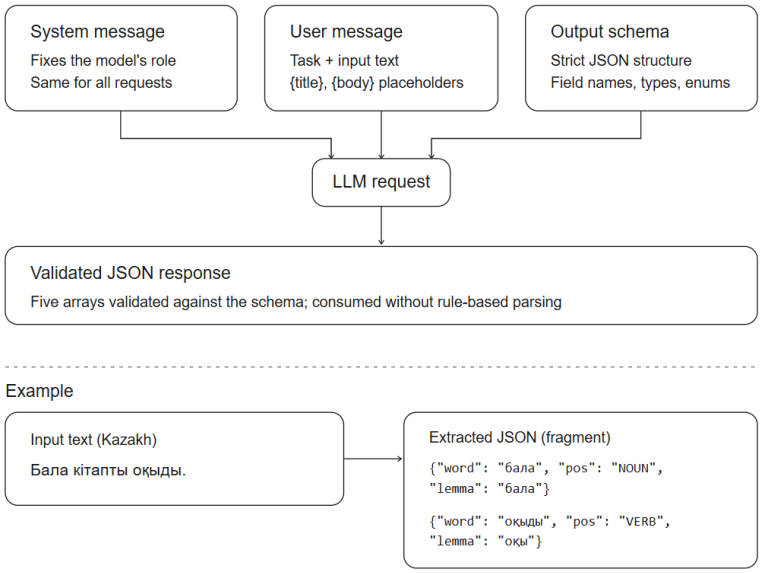

Figure 6. Structure of the LLM request and an example of linguistic extraction based on a JSON schema.

Figure 6 illustrates the architecture of the LLM request, including the relationship between the system message, user message, and output schema, as well as the organization of the linguistic extraction process in JSON format.

3.2.3. LLM invocation

Prompts are sent to the LLM API through an asynchronous client built on aiohttp. The client batches concurrent requests up to a configurable concurrency limit (default 8) and applies the following sampling parameters: temperature = 0.2, top_p = 1, max_tokens set to the largest schema-conformant response size observed during prompt finalization, and a hard request timeout of 60 s. The low temperature minimizes lexical variability across runs, while *top_p* is set to 1 so that temperature remains the sole stochastic control. The client handles three types of failures during API communication. Transport errors, including HTTP 5xx responses and connection timeouts, are retried using exponential backoff intervals of 1, 2, and 4 seconds for up to three attempts. Rate-limit responses (HTTP 429) are retried after the delay specified in the API's *Retry-After* header. If the model returns a response that cannot be parsed as valid JSON or does not conform to the required schema, the request is retried up to two additional times with an appended system reminder instructing the model to return valid JSON only. If all retry attempts fail, the chunk is skipped and logged for manual review. Schema-validation failures (the model returns text that does not parse as JSON, or that parses but does not match the schema) trigger up to two additional retries with an appended system reminder: “Your previous response did not conform to the schema. Reply with valid JSON only.” If all three attempts fail, the chunk is skipped and logged for manual review; in the operational corpus this occurred for 2.9% of chunks.

3.2.4. Output parsing and validation

The raw model response is parsed in two stages. First, the response text is stripped of any code-block fences (the model occasionally wraps the JSON in triple backticks despite the system instruction) and is parsed with the standard-library json module. Second, the parsed dictionary is validated against a Pydantic model that mirrors the schema given in the prompt. The Pydantic model enforces field presence, value types, and the closed enumerations on pos, type, and relation. The literal-presence constraint on word, lemma, and head_word is checked outside the Pydantic model by a separate validator that searches the original chunk text; entities that fail this check are dropped and recorded as hallucinations. The comparison is not a raw byte-level match. Both the extracted value and the chunk are first put through the NFC and whitespace normalization of Section 3.2.1 and then case-folded, so that differences in Unicode composition, casing, or incidental whitespace do not cause spurious mismatches.

The surface-form fields (word and head_word) are matched as they occur in the text, so Kazakh agglutinative inflection is handled by matching the full inflected token rather than by stemming it. The lemma - a dictionary form that in an agglutinative language frequently does not occur verbatim - is matched morphology-aware: it is accepted when it is present in the normalized chunk or when it is the stem of a surface token already present, which covers the usual case of a suffixed form sharing the lemma's stem. Orthographic variants, or lemmas that still differ from every source token after normalization, are not reconciled: the validator deliberately errs toward rejecting ungrounded strings, and the legitimate forms it occasionally rejects on this account are the false positives quantified in Section 5.5. If validation succeeds, the parsed object becomes the canonical extraction record for the chunk and is passed to the population stage. If validation fails after retries, the record is discarded; this is the same condition that contributes to the structural-error rate reported in table 6.

3.2.5. Mapping the validated record to OWL

Validated records are converted to OWL elements through a fixed mapping implemented in OWLready2. There is no LLM call at this stage and no rule-based text processing; the mapping is deterministic given the schema. Table 2 lists the mapping for each schema field.

Table 2. Mapping from validated extraction record fields to OWL elements.

Schema field	*OWL element produced*
parts_of_speech(i)	A PartOfSpeech individual whose IRI is derived from the word value, linked to a Lemma individual via the hasLemma object property. The pos value selects the OWL subclass (e.g. Noun, Verb).
grammatical_categories(i)	A GrammaticalCategory individual annotated with the categoryName and categoryValue data properties, linked to the corresponding word individual via hasCategory.
syntactic_units(i)	An instance of the OWL subclass of SyntacticUnit selected by the type field (Phrase, Clause, Sentence); the head word is linked via hasHead.
usage_examples(i)	A UsageExample individual; the word value is linked via illustratesWord, and the example string is stored as the exampleText data property.
relations(i)	An object-property assertion between the individuals identified by from and to; the property is selected by the relation field from a fixed registry (hasForm, belongsTo, modifies, governs).

Each created individual receives a deterministic IRI of the form ns#{Class}_{slug}_{hash}, where slug is a transliterated form of the original Kazakh string and hash is the first eight characters of the SHA-1 digest of that string. This scheme is the deduplication mechanism of the pipeline: re-extracting the same item from a different source produces the same IRI, so successive runs over an updated corpus merge into the existing ontology rather than duplicating it.

For each created individual, three rdfs:label annotations are added: Kazakh (the original surface form), Russian, and English. The Russian and English labels are produced by a single follow-up LLM call per chunk with a translation-only prompt; the call is batched at the chunk level rather than the individual level to limit API overhead. After all individuals from a chunk have been added, the OWL reasoner (HermiT, invoked through OWLready2) is run to check

 ISSN 2723-6471

ontology consistency. If the reasoner reports an inconsistency, all individuals added by the offending chunk are rolled back and the chunk is logged for manual review.

3.2.6. Ontology-driven adaptive task selection

The preceding stages construct the subject ontology offline. This subsection describes how the committed ontology is used at runtime to support ontology-driven task selection in the Digital Smart Education Platform. The mechanism is deterministic and rule-based: given the learner state and the ontology, the next task is selected without invoking an LLM. Adaptation operates over three ontology classes: Concept, Task, and ErrorCategory. Tasks are linked to concepts and error categories through the properties assessesConcept and targetsErrorCategory, while hasDifficulty defines three difficulty levels (1–3), and prerequisiteOf specifies prerequisite relationships between concepts. For each learner, the platform maintains the running accuracy over the last five attempts, the current difficulty level, the set of mastered concepts, and the recurrence count of each error category. These records are updated after every interaction.

Task selection follows three ordered rules. (R1) Prerequisite gating: only concepts whose prerequisites have been mastered are eligible. (R2) Error-driven remediation: if the same error recurs at least twice, the next task targets that error category at the current or lower difficulty. (R3) Difficulty-matched progression: otherwise, a task of the current difficulty is selected while avoiding repeated items within the same session. Difficulty is updated using a hysteresis rule: it increases when the running accuracy reaches at least 80%, decreases when it falls below 50%, and otherwise remains unchanged. A concept is considered mastered once the learner achieves at least 80% accuracy at the highest difficulty level, consistent with the mastery criterion used in Section 3.4. Algorithm 1 summarizes the complete selection-and-update procedure. Because every decision is based on ontology assertions and learner-state variables, the adaptation process is fully deterministic, explainable, and auditable, with the applied rule (R1–R3) and ontology entities logged for each selected task.

Algorithm 1. Per-attempt adaptive task selection (runtime; no LLM call).

Input: student s; ontology $\mathcal{O}$; learner state $L(s)$.
Output: next task $t^* \in T$.

1. Update $L(s)$ from the latest logged response.
2. $\mathcal{E}(s) = \{\, c \in C \backslash M_s : \forall p \in C,\ p \prec c \Rightarrow p \in M_s \,\}$ (R1)
3. $c^* = \min_{<_{\text{cur}}} \mathcal{E}(s)$
4. $E_\rho = \{\, e \in E : r_s(c^*, e) \geq \rho \,\}$
5. if $E_\rho \neq \varnothing$ then(R2)
6. $e^* = \operatorname{argmax}_{e \in E_\rho} r_s(c^*, e)$
7. $\text{Cand} = \{\, t \in T : \alpha(t) = c^* \wedge \varepsilon(t) = e^* \wedge \delta(t) \leq d_s(c^*) \,\}$
8. else(R3)
9. $\text{Cand} = \{\, t \in T : \alpha(t) = c^* \wedge \delta(t) = d_s(c^*) \,\} \backslash A_s$
10. if $\text{Cand} = \varnothing$ then $\text{Cand} = \{\, t \in T : \alpha(t) = c^* \,\} \backslash A_s$(fallback)
11. $t^* = \operatorname{argmin}_{t \in \text{Cand}} \nu_s(t)$, ties broken by the lexicographically smallest task IRI
12. $d_s^{\text{old}}(c^*) = d_s(c^*)$
13. $d_s(c^*) \leftarrow \begin{cases} \min(d_s(c^*) + 1,\ \delta_{\max}), & a_s(c^*) \geq \theta^+ \\ \max(d_s(c^*) - 1,\ 1), & a_s(c^*) < \theta^- \\ d_s(c^*), & \text{otherwise} \end{cases}$
14. if $a_s(c^*) \geq \theta^+ \ \wedge\ d_s^{\text{old}}(c^*) = \delta_{\max}$ then $M_s \leftarrow M_s \cup \{c^*\}$
15. return t^*

3.3. LLM benchmarking

To justify the choice of the primary model and to assess the robustness of the pipeline against the choice of LLM, three general-purpose models were benchmarked on the same extraction task: GPT-4o [18], Claude 3.5 Sonnet, and Gemini 1.5 Pro. The API snapshots in table 3 are those that were current at the time of the experiment (2024–2025). The benchmark conclusions concern the schema-constrained extraction behaviour observed with those snapshots, not the present-day versions of the underlying models. All three are proprietary API models, chosen because at the time of the experiment they were the leading general-purpose systems with reliable schema-constrained (JSON) output; the benchmark therefore establishes model suitability only within this proprietary-API setting. The pipeline itself is model-agnostic - any model that returns schema-conformant output can be substituted at the invocation stage (Section 3.2.3)

 ISSN 2723-6471

- so open-source, self-hostable models such as the Llama, Qwen, and Gemma families are natural candidates, but they are not evaluated here, and their exclusion is noted as a limitation in Section 5.5.

Table 3. Models and API snapshots used in the benchmark.

Role	Model	API snapshot	Sampling
Primary	GPT-4o	gpt-4o-2024-08-06	temperature 0.2
Comparison	Claude 3.5 Sonnet	claude-3-5-sonnet-20241022	temperature 0.2
Comparison	Gemini 1.5 Pro	gemini-1.5-pro-002	temperature 0.2

Evaluation corpus. A benchmark set of 90 Kazakh-language paragraphs was sampled from the same source corpus used to populate the operational ontology, in proportion to the three domains of the deployment: Kazakh language, Mathematics, and Computer Science. The corpus is stratified across the three subjects to reflect their proportion in the source material. All models were evaluated on the same 90 paragraphs using identical schema-constrained prompts, ensuring that performance differences are attributable to the models rather than the prompt or schema. Each model was run three times (810 outputs in total), and the low run-to-run variability (SD < 0.025 across all metrics) demonstrates stable relative performance. As the corpus is small but expert-annotated, the benchmark is presented as a pilot study (Section 5.5), with the primary focus on the relative comparison between models rather than absolute performance values.

Ground truth. For each paragraph, two domain experts independently produced a reference extraction consisting of the expected parts of speech, grammatical categories, syntactic units, usage examples, and relations. Inter-annotator agreement at the level of individual extracted entities was $\kappa = 0.84$, indicating almost perfect agreement. Disagreements were resolved by discussion to produce a single gold-standard reference. The two annotators were specialists in Kazakh linguistics and the relevant school subjects. They independently annotated the corpus using a shared guideline derived from the pipeline schema and identical closed label sets, while remaining blind to the model outputs. Cohen's κ was computed before adjudication at the level of individual typed items (entities and relations), not tokens or complete paragraph extractions. Annotations were aligned by surface span and schema field, with unmatched items treated as disagreements. The resulting $\kappa = 0.84$ reflects raw inter-annotator agreement across all entity and relation types. Remaining disagreements were subsequently resolved through adjudication to produce the final gold-standard reference used for model evaluation.

Error categories. Four error types were defined operationally and used to compute the per-model error distribution reported in table 6. A hallucination was recorded when an extracted surface form had no literal, verbatim textual support in the source paragraph; this criterion is deliberately conservative, so that an entity which is semantically wrong but lexically grounded falls under semantic error rather than hallucination. A structural error was recorded when the response did not conform to the required output schema, whether through a missing required field, a type mismatch, or a malformed structure. A semantic error was recorded when an extracted entity was supported by the text but mapped to the wrong class or relation. Finally, a missing relation was recorded when an entity or relation present in the gold standard was absent from the model output.

Two small notes if you want them. The category is named "missing relation" but the definition covers a missing *entity* as well, which is a mismatch a reviewer may pick up — either rename it to "omission" or narrow the definition to relations only. Also, since the four types are not mutually exclusive (one paragraph can contain both a structural and a semantic error), it is worth stating explicitly that table 6 reports the percentage of paragraphs containing at least one error of each type, which is why the row values do not sum to a meaningful total.

Metrics. Precision, recall, and F1-score were computed by comparing each model output to the gold standard. Each error type is reported as the percentage of paragraphs containing at least one error of that type. To assess output stability, every paragraph was processed three times per model; the values reported in table 5 and table 6 are the averages, with the standard deviation across runs treated as a stability indicator. The reported run-to-run standard deviation reflects decoding stability rather than sampling uncertainty. Given the 90-paragraph corpus, absolute precision, recall, and F1 scores should be interpreted with caution, as small between-model differences are not statistically meaningful. Accordingly, the benchmark is used primarily to compare the relative ranking of the models on the same evaluation set

rather than their absolute performance. The benchmark is a pilot evaluation, and the results should be interpreted as indicative rather than definitive.

3.4. Pedagogical experiment

Participants. Sixty-five Grade 6 students from two public schools in Astana participated in the study during September 2025 – January 2026. The sample included 34 male and 31 female students aged 11–12 years. Inclusion criteria were: (i) enrolment in the regular Grade 6 curriculum; (ii) Kazakh as the language of instruction; (iii) availability of a personal device for the duration of the study. Participation was voluntary, and informed consent was obtained from a parent or legal guardian prior to enrolment.

Group assignment. Students were assigned to the control group (n = 32) and the experimental group (n = 33) by class-level allocation rather than by individual randomization. Class-level allocation was used rather than individual randomization because both schools required intact class groups for the duration of the intervention. This is a standard constraint in school-based research; the design is a small quasi-experiment with cluster assignment at the class level. Section 5.5 discusses what this means for inference. Baseline equivalence was verified using an independent-samples t-test on the pre-test scores: no statistically significant difference was observed between the two groups ($t(63) = 0.36$, $p = 0.720$), supporting the validity of subsequent between-group comparisons.

Intervention. The control group used a standard digital textbook for Kazakh language, Mathematics, and Computer Science, without ontology-driven adaptation or LLM-based feedback. The experimental group used the Digital Smart Education Platform described in Section 3.2, which provided ontology-driven content organization, automated error classification, and adaptive task selection. Both groups followed the same curriculum and were assigned the same set of ten learning modules over the 16-week study period. Total instructional time per student was matched across the two groups to control for dosage effects.

Operational definitions of outcome variables. Three primary outcome variables were used to evaluate the effectiveness of the proposed learning environment. Task accuracy was defined as the percentage of post-test items for which a student's response matched the reference answer. Repeated errors referred to the mean number of items per student on which the same error category occurred in two or more attempts within the same module. Sessions to mastery represented the number of platform sessions required for a student to achieve the module mastery threshold, defined as at least 80% accuracy on the module exit test. A session was considered a continuous interaction period lasting 25–45 minutes and ended either when the student exited the platform or after five minutes of inactivity.

Statistical analysis. All outcome variables were checked for normality with the Shapiro–Wilk test prior to inferential testing. When the normality assumption held, between-group differences were tested with independent-samples t-tests and within-group pre-/post-test differences were tested with paired-samples t-tests; when normality was violated, the Mann–Whitney U test or the Wilcoxon signed-rank test was used, respectively. Effect sizes are reported as Cohen's d with 95% confidence intervals computed by the bootstrap method (10,000 resamples). The significance threshold was set at $\alpha = 0.05$; because three primary outcome variables are compared, a Bonferroni-corrected threshold of $\alpha = 0.0167$ was also applied as a sensitivity check, and the conclusions reported in Section 4 are unchanged under the corrected threshold. All computations were performed in Python using the SciPy and statsmodels libraries.

Sample size and post-hoc power. No formal a priori power analysis was conducted prior to recruitment; sample size was determined by the number of intact Grade 6 classes available in the two participating schools. For reference, a two-sample independent-groups t-test targeting a medium effect (Cohen's $d = 0.50$) at $\alpha = 0.05$ (two-tailed) with 80% power requires $n = 64$ participants per group (G*Power 3.1). The total enrolled sample ($n = 65$ across both groups) is therefore underpowered for medium effects at the individual level; the large observed effect sizes ($d = 1.50$ to 2.30) reached statistical significance despite this; observed-effect (post-hoc) power, however, merely re-expresses the p-value and does not establish that the sample was adequate, so these estimates should be interpreted cautiously given the well-documented tendency for effect-size inflation in small pilot studies. The study is best regarded as a proof-of-concept investigation requiring replication with a pre-specified sample and a larger, more diverse cohort.

3.5. Data collection and analysis

Interaction data were logged automatically by the digital platform. Recorded fields included task identifier, response, error category assigned by the automatic classifier described in Section 3.2, session start and end timestamps, and the sequence of tasks attempted. Control-group data were collected using the same logging schema with the adaptive components disabled, so that the recorded variables are directly comparable between the two groups. All records were pseudonymized at the point of collection; the mapping between pseudonym and student identifier was held by the school's responsible teacher and was not accessible to the research team.

Ethics approval and data retention**.** The pseudonymized interaction logs are retained for 5 years after publication for verification purposes, after which the logs are securely deleted and the pseudonym-to-identifier key is destroyed, in accordance with the institution's data-protection policy. The analysis followed a comparative approach. Between-group comparisons addressed research questions RQ2 (content and task adaptation) and RQ3 (learning outcomes); within-group pre-/post-test comparisons in the experimental group additionally addressed RQ3. Trajectory adaptation (RQ2) was assessed by examining the sequence of task-difficulty levels assigned by the adaptive module against the student's running accuracy. Patterns of repeated errors and re-engagement with the same learning materials were examined qualitatively to complement the quantitative results.

4. Results and Discussion

This section reports the experimental outcomes in four parts. Section 4.1 reports the yield of the ontology construction pipeline as it operated over the source corpus. Section 4.2 reports the comparative LLM benchmark introduced in Section 3.3. Section 4.3 reports the pedagogical experiment described in Section 3.4. Section 4.4 illustrates the resulting digital textbook environment.

4.1. Ontology construction outcomes

The pipeline was applied to 18 source documents covering Kazakh language, Mathematics, and Computer Science. After preprocessing and chunking (Section 3.2.1), the source corpus produced 312 text chunks that were submitted to the LLM as separate requests. The yield at each stage of the pipeline is summarized in table 4 and visualized in figure 7.

Table 4. Pipeline yield from text chunks to committed ontology contributions.

Pipeline stage	Chunks	Retained (%)	Note
Text chunks (input)	*312*	100.0	-
LLM responses after retries	*303*	*97.1*	Transport, rate-limit, schema retries
Valid JSON (Pydantic)	*289*	*92.6*	Structural-error drop

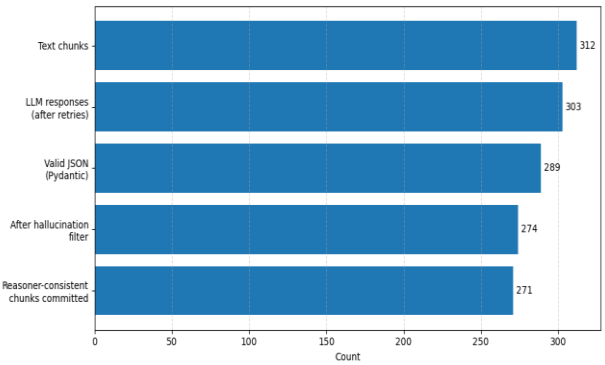

Figure 7. Pipeline yield from source documents to populated ontology.

The committed chunks contributed 1,847 OWL individuals, 2,931 object-property assertions, and 6,512 data-property annotations, across 47 OWL classes. The HermiT reasoner identified inconsistencies in 3 chunks during processing, all of which were rolled back, so the committed ontology is consistent; these rolled-back chunks correspond to the difference between rows 4 and 5 of table 4.

To illustrate the contribution of the reasoning stage, we summarize the three rolled-back chunks that passed literal-presence and schema validation but failed description-logic reasoning. (i) A disjoint-class violation occurred when the same surface form was extracted as both a noun and a verb, causing an IRI collision and membership in two disjoint classes. (ii) A range violation arose when the target of hasForm was typed as a SyntacticUnit instead of the required Lemma, violating the property's range constraints. (iii) A functional-property violation occurred when two distinct head individuals were assigned to the same Clause, contradicting the functional constraint on hasHead. These examples demonstrate that the reasoner detects ontology-level inconsistencies-class disjointness, property functionality, and domain/range violations-that cannot be identified by literal-presence or schema validation alone.

The drop between stages provides direct evidence for the value of each validation step described in Section 3.2. The literal-presence filter (Section 3.2.4) intercepted 5.2% of validated responses, which would otherwise have entered the ontology as hallucinated entities; the schema validator (Pydantic) intercepted a further 4.6% of LLM responses; the reasoner checks intercepted only 1.1% beyond that, indicating that the upstream validators absorb most of the integration cost.

4.2. LLM benchmark

The three models described in Section 3.3 were benchmarked on 90 Kazakh-language paragraphs against the gold standard. The quantitative results are presented in table 5. Inter-annotator agreement at the level of individual extracted entities was $\kappa = 0.84$. Each paragraph was processed three times per model to capture run-to-run stability; reported values are means over the three runs, and standard deviations are shown as error bars in figure 8.

Table 5. Extraction quality per model (mean ± SD over three independent runs). SD here is the run-to-run variability of the schema-constrained prompt across three independent runs, not a bootstrap estimate of sampling uncertainty.

Model	Precision	Recall	F1-score
GPT-4o	*0.92 ± 0.015*	*0.86 ± 0.020*	*0.89 ± 0.014*
Claude 3.5 Sonnet	*0.91 ± 0.018*	*0.84 ± 0.022*	*0.87 ± 0.017*
Gemini 1.5 Pro	*0.88 ± 0.022*	*0.85 ± 0.024*	*0.86 ± 0.019*

Figure 8 shows that GPT-4o attains the highest F1-score, followed by Claude. The run-to-run SD is small (≤ 0.025 across all metrics and models), indicating that the schema-constrained prompt produces stable extractions despite probabilistic LLM sampling. The low run-to-run variance is expected given the decoding configuration (temperature = 0.2, top_p = 1) and the constrained JSON schema, which limits output variability through fixed label sets and literal-presence constraints.

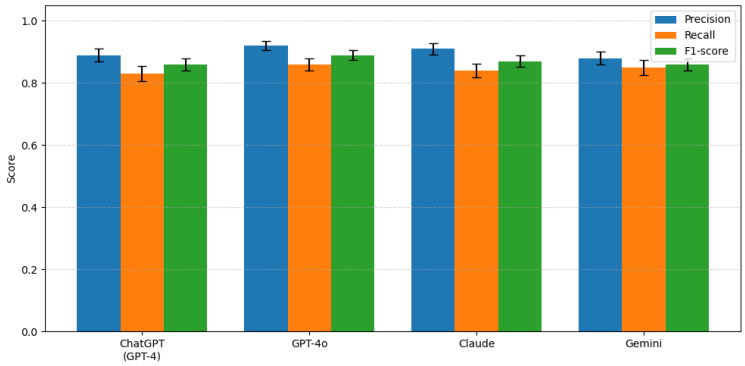

Figure 8. Precision, recall, and F1-score per model (SD bars over three runs).

The three runs represent independent samples under identical settings (without fixed decoding seeds), reflecting decoding stability rather than exhaustive robustness evaluation. Larger repetition counts, multi-seed experiments, and temperature-sensitivity analyses were not performed and are left for future work (Section 5.5). The differences between GPT-4o and Claude are within the SD of either model, which is the basis for the interpretation in Section 5 that the choice between the two depends on the use case (precision versus hallucination rate) rather than on raw F1. The distribution of extraction errors across models is summarized in table 6 and figure 9.

Table 6. Error distribution per model: percentage of paragraphs containing at least one error of each type.

Model	Hallucination (%)	Structural (%)	Semantic (%)	Missing rel. (%)
GPT-4o	5.8	4.3	5.1	7.2
Claude 3.5 Sonnet	4.9	4.8	5.6	8.0
Gemini 1.5 Pro	8.3	6.2	7.1	6.9

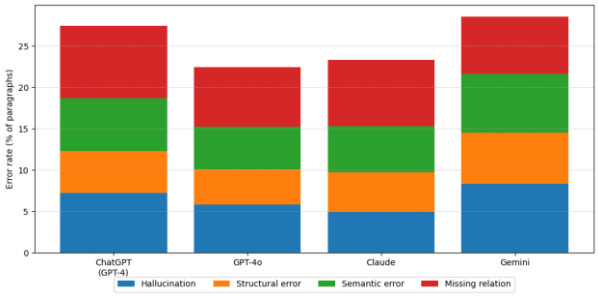

Figure 9. Stacked distribution of error types per model (same data as table 6).

Figure 9 and table 6 together highlight the trade-off between the three models. Claude 3.5 Sonnet has the lowest hallucination rate (4.9%), which is relevant for use cases where injecting fabricated entities into the ontology is unacceptable. GPT-4o has the lowest combined structural-plus-semantic error rate and the highest F1-score, which is the basis for its selection as the primary model for the rest of the system. Gemini 1.5 Pro shows the highest structural-error rate, consistent with weaker schema adherence under the same prompt; its missing-relation rate is comparatively low, which suggests broader entity coverage at the price of structural reliability.

4.3. Operational cost and scalability

Because the pipeline relies on multiple LLM calls, we quantified the operating cost for the full 312-chunk corpus. Each chunk requires two LLM calls (extraction and translation), resulting in approximately 624 calls plus a small retry overhead (2.9% of chunks). The complete build consumed about 1.03M input and 0.44M output tokens. Based on the API list prices in effect during 2024–2025, the estimated one-time ontology construction costs for each model are presented below. Applying the list prices in force at the time of the experiment (2024–2025) to these totals yields the estimated one-time API costs summarized in table 7.

Table 7. Estimated one-time API cost of building the full ontology (312 chunks; 1,847 OWL individuals; 18 source documents), by model.

Model	Input ($/1M)	Output ($/1M)	Corpus build (USD)	Per document (USD)	Per OWL individual (USD)
GPT-4o	2.50	10.00	6.94	0.39	0.0038
Claude 3.5 Sonnet	3.00	15.00	9.64	0.54	0.0052
Gemini 1.5 Pro	1.25	5.00	3.47	0.19	0.0019

Two properties make the approach cost-effective for school-scale deployment. First, the construction cost scales linearly with corpus size because each chunk is processed independently. Second, the LLM is used only during ontology construction; runtime task selection relies solely on SPARQL queries over the committed ontology, resulting in zero marginal LLM cost per student, session, or task. Consequently, operating costs depend only on the one-time corpus size rather than the number of learners or interactions. Furthermore, batch APIs, prompt caching, and self-hosted open-source models (Section 5.5) could further reduce or eliminate API costs.

4.4. Pedagogical experiment outcomes

Baseline equivalence. Pre-test scores were compared between the two groups before the start of the intervention. The independent-samples t-test reported $t(63) = 0.36$, $p = 0.720$ for task accuracy, $t(63) = -0.21$, $p = 0.835$ for repeated errors, and $t(63) = 0.94$, $p = 0.351$ for sessions to mastery; none of the differences reached the 0.05 threshold, supporting the baseline equivalence required for the between-group comparisons that follow (table 8).

ISSN 2723-6471

Table 8. Within-experimental-group changes before and after the intervention. Cohen's d is reported with bootstrap 95% confidence intervals (10,000 resamples).

Indicator	Before	After	Change	t	p	d (95% CI)
Accuracy (%)	64.1	82.5	+18.4	-8.72	< 0.001	1.78 (1.10, 2.45)
Repeated errors	4.3	2.1	–51.2%	6.14	< 0.001	1.25 (0.71, 1.78)
Sessions to mastery	3.4	2.2	–35.3%	5.65	< 0.001	1.12 (0.59, 1.64)

Normality. The Shapiro–Wilk test was applied to each outcome variable in each group. Results were W = 0.978, p = 0.314 (accuracy), W = 0.965, p = 0.087 (repeated errors), and W = 0.971, p = 0.156 (sessions to mastery). Where p > 0.05 the variable was treated as normally distributed and analysed with t-tests; otherwise, the corresponding non-parametric test (Mann–Whitney U or Wilcoxon signed-rank) was used. Q–Q plots for the three variables are provided as figure 10 in the supplementary material (table 9).

Table 9. Between-group comparison of the three outcome variables at post-test. All p-values remain below the Bonferroni-corrected threshold of α = 0.0167.

Indicator	Control	Experimental	Difference	t	p	d (95% CI)
Accuracy (%)	62.3	81.7	+19.4	-7.95	< 0.001	2.30 (1.59, 3.01)
Repeated errors	4.1	1.9	–53.6%	6.38	< 0.001	1.85 (1.20, 2.51)
Sessions to mastery	3.2	2.1	–34.3%	5.10	< 0.001	1.50 (0.91, 2.10)

Outcome variables. Table 8 and table 9 report the within-experimental and between-group comparisons. The experimental group showed gains on all three primary outcome variables, with large or very large effect sizes (Cohen's d between 1.12 and 2.30). All p-values remain below the Bonferroni-corrected threshold of α = 0.0167, so the conclusions are unchanged when multiple comparisons are controlled for. Figure 10 shows the individual-student trajectories underlying these aggregate numbers; figure 11 shows the distribution of each outcome by group; and figure 12 shows the learning curve of mean task accuracy across the ten platform sessions, with the mastery threshold of 80% marked as a reference.

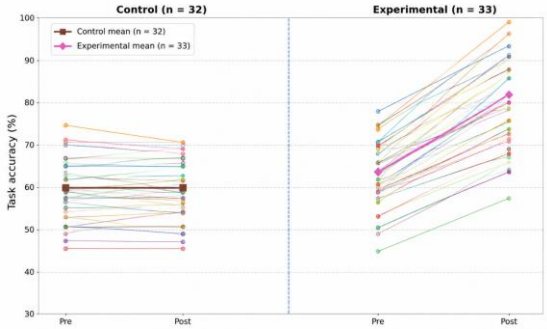

Figure 10. Pre- and post-test task accuracy in the control and experimental groups: (a) Control group (n = 32); (b) Experimental group (n = 33). Individual student trajectories are shown with group mean values.

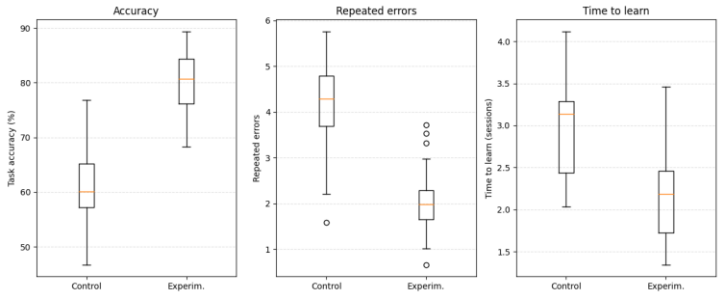

Figure 11. Comparison of the primary outcome variables between the control and experimental groups: (a) task accuracy, (b) repeated errors, and (c) sessions to mastery.

ISSN 2723-6471

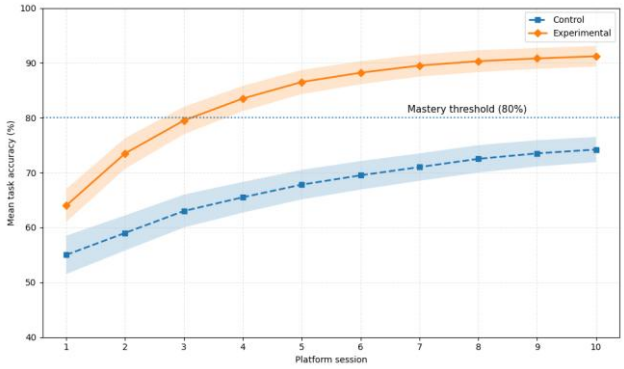

Figure 12. Mean task accuracy across platform sessions by group, with ±1 SE bands and the 80% mastery threshold.

Automatic error classification. The quality of the automatic error classifier described in Section 3.2 was evaluated against the same gold standard used for the LLM benchmark. Results are summarized in table 10, and the corresponding confusion matrix is shown in figure 13.

Table 10. Quality of automatic error classification against the gold standard.

Indicator	Control
Accuracy (%)	62.3
Repeated errors	4.1
Sessions to mastery	3.2

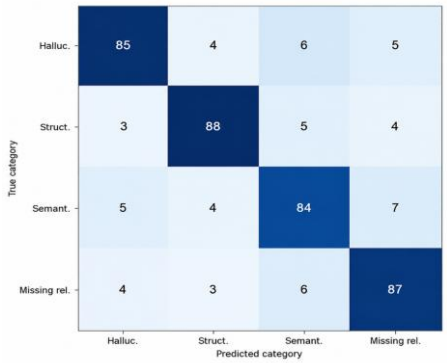

Figure 13. Confusion matrix of the automatic error-category classifier (row-normalized percentages).

The confusion matrix shows that misclassifications are evenly distributed across the off-diagonal cells, with no single pair of categories absorbing a disproportionate share of the errors. The most frequent confusion is between Semantic error and Missing relation, which is consistent with the operational closeness of these two categories: a semantic mismatch in an extracted relation is, in many cases, indistinguishable from an omitted relation at the surface level.

4.5. Digital textbook deployment

The committed ontology drives the Digital Smart Education Platform, which delivers learning materials in Kazakh for the three target subjects. Each content element - theoretical section, terminology entry, interactive task, and test item - is linked to one or more ontology objects, which is what enables the navigation and adaptation reported above. Screenshots of the deployed interface, illustrating the alignment between content and ontology, are provided in the supplementary material: the main view of the Grade 6 textbook, a fragment of the table of contents, an excerpt of theoretical material from Lesson 1, the lesson thesaurus, and an example test item. The system supports natural-language interaction in Kazakh, individualized learning paths, automated assessment, and structured feedback. Educational content is organised through semantic representations: ontological links between concepts, tasks, and assessment elements support navigation, task selection, and content adaptation within the textbook. Task selection is performed through SPARQL queries over the committed ontology rather than by hand-coded rules or LLM calls. After each attempt, the platform records the learner's response and, for incorrect answers, the assigned error category. Based

 ISSN 2723-6471

on the current concept, difficulty level, and error history, the adaptive module retrieves appropriate tasks linked through the ontology. The ontology provides the semantic structure (tasks, concepts, difficulty levels, and error categories), while the reasoner ensures its consistency before deployment (Section 3.2.5). As the LLM is used only during ontology construction, runtime adaptation remains deterministic, explainable, and efficient. This deployment supports the integration of digital technologies into Kazakh-language education and contributes to the development of the national digital educational environment.

Section 5 interprets these outcomes against the research questions stated in Section 1 and against the methodological positioning developed in Section 2.5. The supplementary material (figure 14) reports the Q–Q plots that underlie the normality assessment in Section 4.3.

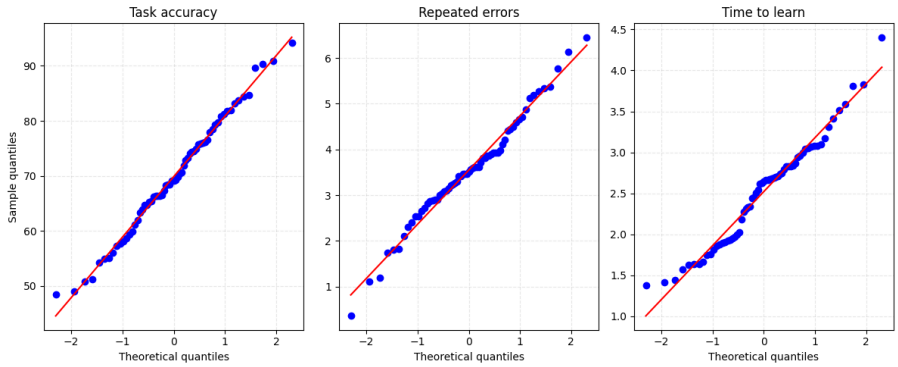

Figure 14. Q–Q plots for the primary outcome variables: (a) task accuracy, (b) repeated errors, and (c) sessions to mastery. These plots were used to assess the normality assumption prior to inferential statistical analysis.

5. Discussion

This section interprets the experimental results in relation to the three research questions stated in Section 1 and against the methodological positioning developed in Section 2.5. Sections 5.1–5.3 address each research question in turn; Section 5.4 then reads the error-rate numbers from the LLM benchmark through the positioning lens, closing the loop between the contributions of Section 1 and the empirical evidence of Section 4. Sections 5.5 and 5.6 discuss the limitations of the present study and the ethical considerations that arise from deploying LLM-based components in primary-school education.

5.1. RQ1 - Ontology-driven structure of the digital textbook

RQ1 asked how a subject ontology, automatically enriched using LLMs, can be used to form the structure of a digital textbook (curricular sections, concepts, tasks, and assessment elements). The pipeline yield reported in Section 4.1 shows that 271 of 312 source-text chunks (86.9%) survived the full validation chain and contributed to the operational ontology, producing 1,847 OWL individuals across 47 OWL classes. The HermiT reasoner verified the consistency of the committed ontology after rollback of the small number of inconsistent chunks reported in Section 4.1, which is the formal guarantee that the resulting structure can be used by symbolic components - SPARQL queries, the adaptive task-selection module, and the error-category classifier - with the standard semantics of a description-logic knowledge base. The measurable contribution of description-logic reasoning in this deployment was, however, modest: the reasoner intercepted only 1.1% of responses beyond the upstream validators (three chunks; Section 4.1), so its role is best understood as a consistency guarantee that licenses symbolic downstream use rather than as a major error filter. The study does not isolate the independent contribution of reasoning to the learning outcomes, which follow from the adaptive use of the resulting ontology rather than from description-logic inference as such.

The structural role of the ontology is visible in Section 4.4: every curricular section, terminology entry, interactive task, and test item is linked to one or more OWL individuals. This follows directly from the schema-co-design contribution (Section 1): because the schema given to the LLM is the OWL T-box, the LLM's structured outputs are themselves the units of the textbook's organisational scheme, and no separate mapping layer is required.

Compared with prior ontology-based educational systems that rely on manual annotation [34], the proposed approach builds the same structural artefact at substantially lower curation cost, because no human annotator labels educational text after the LLM stage. Compared with prior LLM-based feedback systems that return natural language [20], [22], the proposed approach builds a navigable structure that persists between sessions and is queryable by components other than the LLM. This combination - low curation cost and persistent symbolic structure - is the precondition for the adaptation and outcomes reported under RQ2 and RQ3 below.

5.2. RQ2 - Content and task adaptation to the learner's level

RQ2 asked to what extent the ontology-oriented textbook supports adaptation of educational content and tasks to the student's preparation level. The between-group comparison reported in table 9 shows that students in the experimental group reached the module-mastery threshold (≥ 80% accuracy on the exit test) in 2.1 sessions on average, compared with 3.2 sessions for the control group (Cohen's d = 1.50, 95% CI (0.91, 2.10)). The learning curve in figure 12 shows that the two groups diverge by the second session, with the experimental group crossing the mastery threshold by session 4 while the control group does not reach it within the ten-session study window. The trajectory-adaptation analysis described in Section 3.5 confirms that the difference is not a global shift but a session-by-session response to running accuracy: when an experimental-group student misses an item in error category E, the next task selected by the adaptive module is one that is annotated, via the ontology, with the same concept and a difficulty one step below the current level.

Importantly, the form of adaptation observed here is not specific to the educational domain. It follows from the structural property identified under RQ1: because tasks, error categories, concepts, and difficulty levels are all OWL individuals linked by the same T-box, an adaptive selector can be expressed as a SPARQL query rather than as a hand-coded rule set. This is the structural prerequisite that distinguishes ontology-driven adaptation from RAG-based or annotation-based approaches, as set out in Section 2.5.

5.3. RQ3 - Effect on individual learning trajectories and outcomes

RQ3 asked how ontology-based assessment and feedback mechanisms influence the formation of individual learning trajectories and learning outcomes. The pre-/post-test comparison within the experimental group (table 8) shows large gains in task accuracy (+18.4 percentage points; d = 1.78) and large reductions in repeated errors (−51.2%; d = 1.25) and sessions to mastery (−35.3%; d = 1.12). The between-group comparison (table 9) shows that these gains are not attributable to the digital format itself: the control group, which used a non-ontology digital textbook for the same material over the same period, gained substantially less on all three measures. The pre-/post-test paired-trajectory plot (figure 10) shows that the gain is broad-based rather than driven by a small number of outliers: 31 of 33 experimental-group students gained at least 10 percentage points in task accuracy, compared with 22 of 32 in the control group.

The quality of the automatic error classifier (table 10; figure 13) is a separate but related result. Precision of 0.89 and recall of 0.83 at the entity level indicate that the same ontology that supports adaptive content selection is also a reliable substrate for automated formative feedback: 84–88% of student errors are correctly mapped to a category in the OWL error-class hierarchy and can therefore be addressed by a category-specific feedback rule, again expressed as a SPARQL pattern rather than as a hand-coded rule set. This is what closes the loop from learner response to ontology-aligned, pedagogically interpretable feedback that the European Commission's ethical guidelines [5] identify as a requirement for AI use in education.

5.4. Error rates in relation to RQ1 and the methodological positioning

The error-distribution results in table 6 can now be read against the positioning established in Section 2.5. The hallucination rates of 4.9–8.3% observed across models are the rates at which the literal-presence filter intercepts extracted entities that have no support in the source text; under approaches that lack such a filter, these entities would have entered the knowledge representation, and under approaches that rely on prompt-based hallucination control alone they would have been intercepted only probabilistically. Because the filter enforces verbatim surface-form grounding, these are interception rates for one specific failure mode - ungrounded surface forms - and not a general hallucination measure; their narrow definition and its effect on generalizability are discussed in Section 5.5. The structural-error rates of 4.3–6.2% reflect responses that did not conform to the schema; in a generic knowledge-graph-extraction pipeline

these would have required an alignment step to repair, whereas in the schema-co-designed pipeline they are simply rejected and the chunk is reprocessed. The semantic-error rates of 5.1–7.1% are the residual disagreement between LLM extractions and the gold standard; this is the part of the error budget that is genuinely shared with other LLM-based extraction approaches, and it is the rate where future improvements in validation are most likely to land. The HermiT consistency check, run before commit, intercepted only the three inconsistent chunks reported in Section 4.1; the very small drop between rows 4 and 5 of table 4 (274 → 271) shows that the upstream literal-presence and Pydantic validators absorb most of the integration cost, leaving the reasoner as a final safety net rather than the primary defence.

This reading closes the loop between the four contributions of Section 1 and the numerical evidence of Section 4. The schema-as-T-box contribution (i) is what keeps the structural-error drop in table 4 small. The literal-presence filter (ii) is operationalised by the hallucination-rate column of table 6. The multilingual-labelling step (iii) is what makes the Russian and English labels in the deployed textbook (Section 4.4) reproducible at LLM-call cost. The shared-schema contribution (iv) is what enables the joint analysis of subject-matter and pedagogical content reported under RQ2 and RQ3.

5.5. Limitations

Several limitations of the present study should be acknowledged. First, the pedagogical sample was relatively small (n = 65) and limited to Grade 6, three school subjects, and one educational system, restricting the generalizability of the findings. Second, the 16-week intervention included no delayed assessment, so long-term knowledge retention could not be evaluated. Third, the large observed effect sizes should be interpreted cautiously because small, underpowered studies are susceptible to novelty effects and Type M (magnitude exaggeration) error [43]. Larger, adequately powered replication studies are therefore required. Fourth, the LLM benchmark should be regarded as a pilot evaluation. The benchmark corpus comprised only 90 expert-annotated paragraphs, so the absolute precision, recall, and F1 values should be treated as indicative, whereas the relative ranking of the evaluated models is more reliable because all models were tested on the same corpus. Fifth, the framework currently supports only textual educational content and does not process multimodal resources such as diagrams, audio, or video. Sixth, the ontology is constructed offline and remains largely static during operation, with no real-time ontology updates based on learner interactions. Seventh, participants were assigned at the class level rather than by individual randomization. Consequently, class-, teacher-, and school-level effects may have influenced the observed outcomes, and residual confounding cannot be excluded. Future multi-school, cluster-randomized studies with mixed-effects models are needed. Eighth, no a priori power analysis was conducted, and the study should therefore be regarded as a pilot requiring replication with larger samples. Ninth, although ontology construction costs were quantified, construction throughput, large-scale deployment, and continuous content updates were not evaluated. The hallucination metric is based on literal surface-form matching and may not capture all semantic errors. In addition, the evaluation was conducted only for the Kazakh language and only with proprietary LLMs (GPT-4o, Claude 3.5 Sonnet, and Gemini 1.5 Pro); future work should investigate multilingual performance and open-source alternatives. Finally, model stability was assessed using only three runs at a single decoding temperature. Broader evaluations with multiple seeds, temperatures, and repetitions are required to provide a more comprehensive assessment of robustness.

5.6. Ethical considerations

Deploying LLM-based components in primary-school education raises ethical considerations beyond the institutional approval and informed-consent procedures described in Section 3.4. Three are particularly relevant here. First, the model outputs that drive task selection and feedback are produced by a probabilistic system; the literal-presence filter, Pydantic validation, and reasoner-consistency check described in Section 3.2 are the technical safeguards against undue influence of model errors on individual learners, but they do not eliminate the residual risk of inappropriate task assignment or feedback. Human oversight by the responsible teacher therefore remains essential, particularly in the early phase of any deployment.

Second, the alignment of automated feedback with formal learning outcomes - required by the European Commission's ethical guidelines [5] – is a structural property of the proposed approach, not an external constraint imposed after the fact. Every feedback rule in the system is expressed against the OWL error-class hierarchy that is itself linked to

curricular concepts; this makes the connection between model output and pedagogical goal explicit and auditable, which is the structural difference from RAG-based or annotation-based approaches noted in Section 2.5.
Third, the present study collects and processes interaction data for minors. All records were pseudonymized at the point of collection, the pseudonym-to-identifier mapping was held only by the responsible teacher, and no personally identifying information left the school's information system. These safeguards are necessary but not sufficient: deployment in a wider population will require formal data-processing agreements with each participating school and, where applicable, registration with the national data-protection authority. Future versions of the system should also expose, to the responsible teacher, an audit interface that lists the ontology assertions added in each batch run and the chunks that were rolled back during reasoning, so that the symbolic record of the system's state remains reviewable in practice rather than only in principle. Ethical safeguards included parental informed consent, voluntary participation, and the right to withdraw without consequence (Section 3.4). LLMs were used only during ontology construction on publicly available educational texts; learner interaction data remained within the school platform and were never transmitted to external LLM APIs. Interaction logs were pseudonymized, stored locally, and accessible only to the responsible teacher. Residual risks include inappropriate task recommendations and the possibility of re-identification from longitudinal logs. These risks were mitigated through teacher oversight, data minimization, and access controls, although they cannot be eliminated entirely and should be addressed in future large-scale deployments through formal data-protection impact assessments.

6. Conclusion and Future Work

This study proposed an LLM-based pipeline for the automatic enrichment of a subject ontology and evaluated the resulting digital textbook in a Grade 6 setting covering Kazakh language, Mathematics, and Computer Science. Four operational decisions distinguish the pipeline from neighbouring approaches: the JSON schema used to constrain the LLM output is the OWL T-box of the target ontology, so the output is directly populatable without an alignment step; hallucinations are intercepted by a literal-presence check against the source text rather than by prompt-based heuristics alone; multilingual labels in Kazakh, Russian, and English are produced by a single follow-up LLM call to support low-resource-language deployment; and subject-matter content and pedagogical error categories share a single schema, which is what enables ontology-driven adaptive feedback.

The empirical evaluation supports each of these decisions. The pipeline yield (table 4) shows that 86.9% of source-text chunks survive the full validation chain, with most of the integration cost absorbed by the upstream Pydantic and literal-presence validators rather than by the reasoner. The LLM benchmark (table 5 and table 6) shows that the schema-constrained prompt produces stable extractions across three general-purpose models, with hallucination rates between 4.9% and 8.3% and run-to-run standard deviations below 0.025 for all metrics. The pedagogical experiment (table 8 and table 9) shows large between-group effect sizes on all three primary outcome variables (Cohen's d between 1.50 and 2.30), and the trajectory analysis (figure 10 and figure 11) indicates that the gain is broad-based and emerges from the second session of platform use rather than from end-of-period effects.

The findings have three practical consequences. For developers building educational LLM systems, the schema-as-T-box discipline removes the need for the separate vocabulary-alignment step that prior work identifies as a recurrent source of integration error in LLM-to-knowledge-graph pipelines [7], [14], and the literal-presence filter cuts hallucination at the integration boundary to between 4.9% and 8.3% across the three models benchmarked here. In the Kazakh deployment, the same pipeline produces Kazakh, Russian, and English labels in a single LLM call, which removes the parallel-corpus dependence that limits supervised methods; because this labelling is model-generated rather than corpus-aligned, the approach is expected to extend to other low-resource languages, although that extension has not been tested here and is left to future work. In the school deployment, the ontology-driven configuration cut sessions to mastery by 34.3% compared with a non-ontology digital textbook used on the same material (table 9); because every feedback rule is expressed against the OWL error-class hierarchy, the link between automated feedback and formal learning outcomes is auditable in a form consistent with the European Commission ethical guidelines.

Several directions for future work are open. First, large-scale benchmarking of the pipeline across more LLM architectures - including open-source, self-hostable models such as the Llama, Qwen, and Gemma families - and across multiple low-resource languages would strengthen the generalizability of the benchmark results in Section 4.2; the

integration of confidence-based validation and human-in-the-loop review at the entity level would extend the safety guarantees of the present validation chain. Second, the framework is currently restricted to textual data and does not handle multimodal educational content; extending the prompt construction and schema to support diagrams, audio, video, and interactive simulations would substantially broaden the applicability of the approach to subjects such as Mathematics, Engineering, and the natural sciences. Third, the operational ontology is currently enriched in batch and remains static during student interaction; future work will explore real-time ontology updates that respond to incoming learner data, enabling continuous refinement of learning trajectories and more responsive content adaptation. Fourth, the present study reports outcomes over a short experimental period; longitudinal studies extending over a full academic year are needed to evaluate knowledge retention, sustained engagement, and the long-term effect of ontology-driven adaptive learning on individual learner trajectories.

Taken together, these directions are expected to increase both the methodological rigour of the approach and its practical applicability in operational educational settings. The combination of schema-disciplined LLM extraction, persistent symbolic knowledge representation, and ontology-aligned feedback offers a route toward AI-supported education in which the link between model output and pedagogical goal remains explicit, traceable, and reviewable at every stage - a route that other educational deployments of LLMs can take without committing to an architecture in which the knowledge representation itself is opaque or ephemeral.

7. Declarations

7.1. Author Contributions

Conceptualization: R.T., B.K., Y.K., U.T., A.N., and Z.L.; Methodology: B.K.; Software: R.T.; Validation: R.T., B.K., Y.K., U.T., A.N., and Z.L.; Formal Analysis: R.T., B.K., Y.K., U.T., A.N., and Z.L.; Investigation: R.T.; Resources: B.K.; Data Curation: B.K.; Writing Original Draft Preparation: R.T., B.K., Y.K., U.T., A.N., and Z.L.; Writing Review and Editing: B.K., R.T., Y.K., U.T., A.N., and Z.L.; Visualization: R.T.; All authors have read and agreed to the published version of the manuscript.

7.2. Data availability Statement

The raw learner-interaction logs cannot be released because they were collected from minors under a school-level data-processing arrangement. The pseudonym-to-identifier mapping is held by the responsible teacher and was not transferred to the research team. The OWL ontology produced by the pipeline, the gold-standard annotations for the 90 Kazakh-language paragraphs reported in Section 4.2, and the aggregated per-module outcome tables behind Tables 8–9 are available from the corresponding author on reasonable request, subject to the institutional review board approval that authorized this study.

7.3. Funding

Not available

7.4. Institutional review board statement

The study was conducted in accordance with the Declaration of Helsinki. Informed consent was obtained from all participants and, where applicable, from parents or legal guardians prior to participation. The research instrument, participant information sheets, and consent procedures were reviewed and approved by the Digital Resource Center for Research Ethics at L.N. Gumilyov Eurasian National University (Approval No. 694-K, 09 June 2025).

7.5. Informed Consent Statement

Not applicable.

7.6. Declaration of Competing Interest

Not applicable.

References

[1] F. Miao and M. Cukurova, *AI Competency Framework for Teachers*. Paris, France: UNESCO, 2024, doi: 10.54675/ZJTE2084.

[2] OECD, *OECD Digital Education Outlook 2023: Towards an Effective Digital Education Ecosystem*. Paris, France: OECD Publishing, 2023, doi: 10.1787/c74f03de-en.

[3] World Bank, UNICEF, FCDO, USAID, Bill & Melinda Gates Foundation, and UNESCO, *The State of Global Learning Poverty: 2022 Update*. Washington, DC, USA: World Bank, 2022.

[4] M. R. Islam, M. U. Ahmed, S. Barua, and S. Begum, "A systematic review of explainable artificial intelligence in terms of different application domains and tasks," *Applied Sciences*, vol. 12, no. 3, Art. no. 1353, pp. 1-38, 2022, doi: 10.3390/app12031353.

[5] European Commission, Directorate-General for Education, Youth, Sport and Culture, *Ethical Guidelines on the Use of Artificial Intelligence (AI) and Data in Teaching and Learning for Educators*. Luxembourg: Publications Office of the European Union, 2022, doi: 10.2766/153756.

[6] W. Villegas-Ch and J. García-Ortiz, "Enhancing learning personalization in educational environments through ontology-based knowledge representation," *Computers*, vol. 12, no. 10, Art. no. 199, pp. 1-19, 2023, doi: 10.3390/computers12100199.

[7] J. Li, D. Garijo, and M. Poveda-Villalón, "Large language models for ontology engineering: A systematic literature review," *Semantic Web*, published online ahead of print, vol. 2026, no. July, pp. 1-44, 2026, doi: 10.1177/22104968261465514.

[8] C. Li, X. Lee, and X. Wu, "Provide personalized programming learning for individuals based on large language models," *Alexandria Engineering Journal*, vol. 132, no. November, pp. 396–406, 2025, doi: 10.1016/j.aej.2025.10.026.

[9] E. Kochmar, D. Do Vu, R. Belfer, V. Gupta, I. V. Serban, and J. Pineau, "Automated data-driven generation of personalized pedagogical interventions in intelligent tutoring systems," *International Journal of Artificial Intelligence in Education*, vol. 32, no. 2, pp. 323–349, 2022, doi: 10.1007/s40593-021-00267-x.

[10] J. Ocumpaugh, R. D. Roscoe, R. S. Baker, S. Hutt, and S. J. Aguilar, "Toward asset-based instruction and assessment in artificial intelligence in education," *International Journal of Artificial Intelligence in Education*, vol. 34, no. 4, pp. 1559–1598, 2024, doi: 10.1007/s40593-023-00382-x.

[11] L. Kohnke, D. Zou, and F. Su, "Exploring the potential of GenAI for personalised English teaching: Learners' experiences and perceptions," *Computers and Education: Artificial Intelligence*, vol. 8, no. June, Art. no. 100371, pp. 1-11, 2025, doi: 10.1016/j.caeai.2025.100371.

[12] F. Abbes, S. Bennani, and A. Maalel, "Generative AI and gamification for personalized learning: Literature review and future challenges," *SN Computer Science*, vol. 5, no. 8, Art. no. 1154, pp. 1-17, 2024, doi: 10.1007/s42979-024-03491-z.

[13] A. S. Almogren, W. M. Al-Rahmi, and N. A. Dahri, "Integrated technological approaches to academic success: Mobile learning, social media, and AI in visual art education," *IEEE Access*, vol. 12, no. November, pp. 175391–175413, 2024, doi: 10.1109/ACCESS.2024.3498047.

[14] T. Z. and J. V. Fonou-Dombeu, "A review of state of the art deep learning models for ontology construction," *IEEE Access*, vol. 12, no. May, pp. 82354–82383, 2024, doi: 10.1109/ACCESS.2024.3406426.

[15] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, R. Chou, J. Glanville, J. M. Grimshaw, A. Hróbjartsson, M. M. Lalu, T. Li, E. W. Loder, E. Mayo-Wilson, S. McDonald, L. A. McGuinness, L. A. Stewart, J. Thomas, A. C. Tricco, V. A. Welch, P. Whiting, and D. Moher, "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," *BMJ*, vol. 372, no. March, Art. no. n71, pp. 1-9, 2021, doi: 10.1136/bmj.n71.

[16] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993, doi: 10.1006/knac.1993.1008.

[17] N. F. Noy and D. L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*, Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, Stanford University, Stanford, CA, USA, 2001.

[18] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, et al., "GPT-4 technical report," *arXiv:2303.08774*, vol. 2023, no. March, pp. 1-100, 2023, doi: 10.48550/arXiv.2303.08774.

[19] E. Kasneci, K. Sessler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier, S. Krusche, G. Kutyniok, T. Michaeli, C. Nerdel, J. Pfeffer, O. Poquet, M. Sailer, A. Schmidt, T. Seidel, M.

Stadler, and G. Kasneci, "ChatGPT for good? On opportunities and challenges of large language models for education," *Learning and Individual Differences*, vol. 103, no. April, Art. no. 102274, pp. 1-13, 2023, doi: 10.1016/j.lindif.2023.102274.

[20] T. Wu, M. Terry, and C. J. Cai, "AI chains: Transparent and controllable human-AI interaction by chaining large language model prompts," in *Proc. 2022 CHI Conf. Human Factors in Computing Systems (CHI '22)*, New Orleans, LA, USA, vol. 2022, no. April, Art. no. 385, pp. 1–22, 2022, doi: 10.1145/3491102.3517582.

[21] Y. Dai, Z. Lin, and A. Liu, "Facilitating students' adaptive help-seeking and peer interactions through an analytics-enhanced forum in engineering design education," *Procedia CIRP*, vol. 128, no. 2024, pp. 292–297, 2024, doi: 10.1016/j.procir.2024.06.024.

[22] A. Rachha and M. Seyam, "Explainable AI in education: Current trends, challenges, and opportunities," in *Proc. SoutheastCon 2023*, Orlando, FL, USA, vol. 2023, no. May, pp. 232–239, 2023, doi: 10.1109/SoutheastCon51012.2023.10115140.

[23] M. Bernabei, S. Colabianchi, A. Falegnami, and F. Costantino, "Students' use of large language models in engineering education: A case study on technology acceptance, perceptions, efficacy, and detection chances," *Computers and Education: Artificial Intelligence*, vol. 5, no. 2023, Art. no. 100172, pp. 1-18, 2023, doi: 10.1016/j.caeai.2023.100172.

[24] S. Athanassopoulos, P. Manoli, M. Gouvi, K. Lavidas, and V. Komis, "The use of ChatGPT as a learning tool to improve foreign language writing in a multilingual and multicultural classroom," *Advances in Mobile Learning Educational Research*, vol. 3, no. 2, pp. 818–824, 2023, doi: 10.25082/AMLER.2023.02.009.

[25] C. B. Divito, B. M. Katchikian, J. E. Gruenwald, and J. M. Burgoon, "The tools of the future are the challenges of today: The use of ChatGPT in problem-based learning medical education," *Medical Teacher*, vol. 46, no. 3, pp. 320–322, 2024, doi: 10.1080/0142159X.2023.2290997.

[26] M. A. Ahmed, "ChatGPT and the EFL classroom: Supplement or substitute in Saudi Arabia's eastern region," *Information Sciences Letters*, vol. 12, no. 7, pp. 2727–2734, 2023, doi: 10.18576/isl/120704.

[27] I. S. Chaudhry, S. A. M. Sarwary, G. A. El Refae, and H. Chabchoub, "Time to revisit existing student's performance evaluation approach in higher education sector in a new era of ChatGPT—A case study," *Cogent Education*, vol. 10, no. 1, Art. no. 2210461, pp. 1-30, 2023, doi: 10.1080/2331186X.2023.2210461.

[28] E. Abolkasim and M. Hasan, "Integrating ChatGPT in education and learning: A case study on Libyan universities," *Journal of Pure & Applied Sciences*, vol. 23, no. 2, pp. 19–24, 2024, doi: 10.51984/jopas.v23i2.3082.

[29] V. Malyshev, A. Gab, D. Shakhnin, Y. Lipskyi, V. Kovalenko, and O. Orel, "Research of the global higher education market and of the use of artificial intelligence in this field," *EUREKA: Social and Humanities*, no. 6, no. November, pp. 28–41, 2024, doi: 10.21303/2504-5571.2024.003663.

[30] M. Alghazo, V. Ahmed, and Z. Bahroun, "Exploring the applications of artificial intelligence in mechanical engineering education," *Frontiers in Education*, vol. 9, no. February, Art. no. 1492308, pp. 1-24, 2025, doi: 10.3389/feduc.2024.1492308.

[31] H. Snyder, "Literature review as a research methodology: An overview and guidelines," *Journal of Business Research*, vol. 104, no. November, pp. 333–339, 2019, doi: 10.1016/j.jbusres.2019.07.039.

[32] B. Zohuri and F. Mossavar-Rahmani, "Revolutionizing education: The dynamic synergy of personalized learning and artificial intelligence," *International Journal of Advanced Engineering and Management Research*, vol. 9, no. 1, pp. 143–153, 2024, doi: 10.51505/ijaemr.2024.9111.

[33] M. Y. Doo, C. Bonk, and H. Heo, "A meta-analysis of scaffolding effects in online learning in higher education," *International Review of Research in Open and Distributed Learning*, vol. 21, no. 3, pp. 60–80, 2020, doi: 10.19173/irrodl.v21i3.4638.

[34] I. Zualkernan, "Adoption of generative AI and large language models in education: A short review," in *Proc. 2025 Int. Conf. Electronics, Information, and Communication (ICEIC)*, Osaka, Japan, vol. 2025, no. February, pp. 1–5, 2025, doi: 10.1109/ICEIC64972.2025.10879632.

[35] M. Nadăș, L. Dioșan, and A. Tomescu, "Synthetic data generation using large language models: Advances in text and code," *IEEE Access*, vol. 13, no. July, pp. 134615–134633, 2025, doi: 10.1109/ACCESS.2025.3589503.

[36] S. Sharma, P. Mittal, M. Kumar, and V. Bhardwaj, "The role of large language models in personalized learning: A systematic review of educational impact," *Discover Sustainability*, vol. 6, no. 1, Art. no. 243, pp. 1-24, 2025, doi: 10.1007/s43621-025-01094-z.

[37] Y. K. Dwivedi, et al., "Opinion paper: 'So what if ChatGPT wrote it?' Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy," *International Journal of Information Management*, vol. 71, no. August, Art. no. 102642, pp. 1-63, 2023, doi: 10.1016/j.ijinfomgt.2023.102642.

 ISSN 2723-6471

[38] M. Hooda, C. Rana, O. Dahiya, J. P. Shet, and B. K. Singh, "Integrating LA and EDM for improving students success in higher education using FCN algorithm," *Mathematical Problems in Engineering*, vol. 2022, no. 1, Art. no. 7690103, pp. 1-12, 2022, doi: 10.1155/2022/7690103.

[39] I. Osakwe, G. Chen, A. Whitelock-Wainwright, D. Gašević, A. P. Cavalcanti, and R. F. Mello, "Towards automated content analysis of educational feedback: A multi-language study," *Computers and Education: Artificial Intelligence*, vol. 3, no. 2022, Art. no. 100059, pp. 1-10, 2022, doi: 10.1016/j.caeai.2022.100059.

[40] D. A. Shafiq, M. Marjani, R. A. A. Habeeb, and D. Asirvatham, "Student retention using educational data mining and predictive analytics: A systematic literature review," *IEEE Access*, vol. 10, no. July, pp. 72480–72503, 2022, doi: 10.1109/ACCESS.2022.3188767.

[41] J. Kim, K. Merrill, K. Xu, and D. D. Sellnow, "My teacher is a machine: Understanding students' perceptions of AI teaching assistants in online education," *International Journal of Human–Computer Interaction*, vol. 36, no. 20, pp. 1902–1911, 2020, doi: 10.1080/10447318.2020.1801227.

[42] A. D. Samala, S. Rawas, T. Wang, J. M. Reed, J. Kim, N.-J. Howard, and M. Ertz, "Unveiling the landscape of generative artificial intelligence in education: A comprehensive taxonomy of applications, challenges, and future prospects," *Education and Information Technologies*, vol. 30, no. 3, pp. 3239–3278, 2025, doi: 10.1007/s10639-024-12936-0.

[43] A. Gelman and J. Carlin, "Beyond power calculations: Assessing Type S (sign) and Type M (magnitude) errors," *Perspectives on Psychological Science*, vol. 9, no. 6, pp. 641–651, 2014, doi: 10.1177/1745691614551642.