EdgePrompt: Engineering Guardrail Techniques for Offline LLMs in K-12 Educational Settings

Riza Alaudin Syah*
alaudinsyah@graduate.utm.my
Universiti Teknologi Malaysia
Johor Bahru, Malaysia

Christoforus Yoga Haryanto*
cyharyanto@zipthought.com.au
ZipThought
Melbourne, VIC, Australia

Emily Lomempow
ZipThought
Melbourne, VIC, Australia

Krishna Malik
Independent Researcher
Jakarta, Indonesia

Irvan Putra
Independent Researcher
Jakarta, Indonesia

Abstract

EdgePrompt is a prompt engineering framework that implements pragmatic guardrails for Large Language Models (LLMs) in the K-12 educational settings through structured prompting inspired by neural-symbolic principles. The system addresses educational disparities in Indonesia's Frontier, Outermost, Underdeveloped (3T) regions by enabling offline-capable content safety controls. It combines: (1) content generation with structured constraint templates, (2) assessment processing with layered validation, and (3) lightweight storage for content and result management. The framework implements a multi-stage verification workflow that maintains safety boundaries while preserving model capabilities in connectivity-constrained environments. Initial deployment targets Grade 5 language instruction, demonstrating effective guardrails through structured prompt engineering without formal symbolic reasoning components.

CCS Concepts

• **Social and professional topics** → **K-12 education**; • **Applied computing** → **Computer-assisted instruction**; • **Computing methodologies** → *Natural language generation.*

Keywords

Large Language Models, Edge Computing, K-12 Education, AI Safety, Prompt Engineering, Content Filtering, Offline AI, Educational Technology, Guardrails

ACM Reference Format:
Riza Alaudin Syah, Christoforus Yoga Haryanto, Emily Lomempow, Krishna Malik, and Irvan Putra. 2025. EdgePrompt: Engineering Guardrail Techniques for Offline LLMs in K-12 Educational Settings. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3701716.3717810

*Both authors contributed equally to this research.

WWW Companion '25, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1331-6/2025/04
https://doi.org/10.1145/3701716.3717810

1 Introduction

In Indonesia's remote 3T regions ("Terdepan, Terluar, Tertinggal" – Frontier, Outermost, Underdeveloped), mostly unreliable internet connectivity makes cloud-based solutions impractical for classroom activities. EdgePrompt addresses this by enabling teachers to generate and evaluate educational content locally, while keeping cloud services optional for complex tasks.

LLMs show promise in educational applications, but research indicates that technical capabilities alone do not ensure classroom adoption: instructors need control, transparency, and workflow integration [11]. EdgePrompt bridges this gap through structured prompts and automated safeguards, allowing educators to leverage LLMs without deep technical expertise.

Recent advances in LLM guardrails have demonstrated feasible domain-specific control mechanisms [3, 4]. However, these implementations typically assume access to the cloud infrastructure, making them unsuitable for offline settings. Inspired by neural-symbolic architectures [4], we define guardrail techniques as:

(1) **Structured Prompting:** templates embedding the safety constraints with formal validation rules,
(2) **Multi-stage Validation:** sequential prompt-based checks with explicit boundary conditions, and
(3) **Edge Deployment Compatibility:** optimized mechanisms for operation in low-resource environments.

Our framework addresses three key challenges in K-12 education: (1) ensuring robust content safety in offline settings, (2) enabling accurate assessment with edge-based validation, and (3) maintaining consistency in distributed evaluation processes.

The initial deployment of EdgePrompt targets Grade 5 language instruction in Indonesia's 3T regions [6, 12], where internet limitations [1] require local LLM evaluation while selectively leveraging cloud-based resources for content generation. By integrating structured prompt engineering with multi-stage validation, EdgePrompt ensures that teachers can safely and effectively apply AI-driven evaluation methods without requiring extensive technical expertise.

2 Methodology

EdgePrompt has two methodologies: (1) **Prompt Development** to ensure the prompts are suitable for the objectives, and (2) **Framework Development** to ensure the multi-stage prompting can achieve the expected goals.

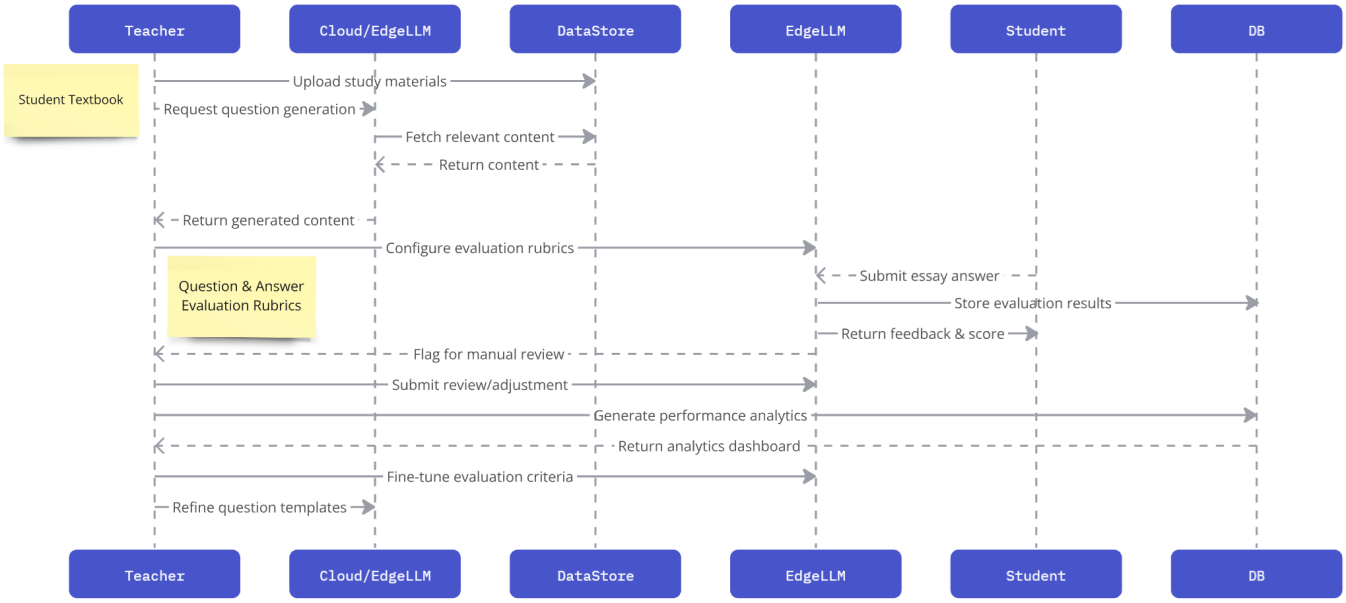

Figure 1: Sequence diagram of the system.

2.1 Prompt Development

To evaluate various prompting approaches for offline LLMs in K-12 educational settings, we propose an evaluation framework. Inspired but distinct from the SPADE methodology which uses fine-tuning to optimize production models [7], our approach relies solely on prompt engineering without modifying the base model parameters.

2.1.1 Evaluation Metrics. We measure prompt effectiveness using:

(1) **Content Validity:** Generated responses align with educational objectives and source materials.
(2) **Safety and Appropriateness:** Adherence to age-appropriate and safe content guidelines.
(3) **Efficiency:** Response latency and computational resource usage on edge devices.
(4) **Teacher Satisfaction:** Qualitative feedback regarding the clarity and usability.
(5) **Robustness:** Consistency of the performance across varied input conditions including variations of teacher requests and student responses.

2.1.2 Experimental Protocol. The evaluation process consists of:

(1) **Candidate Selection:** Identify a diverse set of prompting strategies tailored to our domain requirements.
(2) **Functional Testing:** Deploy candidate prompt in educational tasks to ensure usability.
(3) **Assessment:** Collect both quantitative data (using metrics above) and qualitative feedback from educators.
(4) **Statistical Comparison:** Apply paired statistical analyses to identify significant differences among strategies.
(5) **Iterative Refinement:** Refine prompt designs based on evaluation insights.

2.2 Framework Development

We design a rigidly structured question generation and validation pipeline leveraging cloud and edge LLMs for distinct operational roles. Previous research focused on generating multiple choice questions (MCQs) from educational text [2, 5], creating essay questions [10], and combining NLP and machine learning for structured validation pipelines [9]. Furthering their work, we develop an architecture to generate essay questions and evaluate students' answers while enforcing safety through multi-stage template validation, explicit constraint propagation, and formalized evaluation protocols, as shown in Fig. 1. The core components implement template processing, staged validation, and lightweight integration.

2.2.1 Teacher-Driven Content Generation.

(1) **Question Template Definition:**
   (a) Domain-constrained content templates T_c, e.g. "Write a descriptive paragraph about [topic]",
   (b) Answer space specification A_s with explicit boundaries, e.g. "Response must be 50-100 words, school-appropriate vocabulary", and
   (c) Formal learning objective mapping $O : T_c \rightarrow L$ where L defines permissible learning outcomes, e.g. "Student demonstrates ability to use descriptive language"
(2) **Cloud/Edge LLM Pipeline:**
   (a) Rubric formalization function $R(c_t, v_p)$ where c_t represents teacher criteria and v_p validation parameters, e.g. "4 points: proper length, grade-level vocabulary",
   (b) Transformation $S : R \rightarrow R'$ to ensure edge compatibility, e.g. "Scoring criteria is [condensed rubrics]", and
   (c) Grading template generation $G(R')$ with explicit validation constraints, e.g. "Check: word count 50-100, formal vocabulary"

2.2.2 *Student Answer Evaluation.*

(1) **Edge Validation:**
   (a) Question-answer verification $V(q, a) \rightarrow \{0, 1\}$, e.g. "Does answer describe requested topic?",
   (b) Staged response validation sequence $\{v_1, \ldots, v_n\}$ against the rubric R', e.g. "Length → vocabulary → content → scoring", and
   (c) Boundary enforcement function $B(r) \rightarrow \{valid, invalid\}$ for responses r, e.g. "Filter inappropriate content, off-topic responses"

(2) **Evaluation Logic:**
   (a) Apply rubric R' through transformation $E(r, R')$, e.g. "Count sensory details, check length requirements",
   (b) Calibrated scoring function $S(e)$ for evaluation e, e.g. "3/4 points - meets length, 2 sensory details", and
   (c) Verify the constraints satisfaction $C(s, c_t)$ for score s, e.g. "Response meets safety and topic requirements"

2.2.3 *Teacher Review System.*

(1) **Response Analysis:**
   (a) Edge case detection function $D(r, \theta)$ with threshold θ, e.g. "80% confidence threshold for automated scoring",
   (b) Review triggers $T(d) \rightarrow \{review, accept\}$, e.g. "Borderline scores, unusual patterns", and
   (c) Track patterns $K(h)$ over the evaluation history h, e.g. "Common vocabulary errors, length issues"

(2) **System Adaptation:**
   (a) Rubric adjustment $A : R' \rightarrow R''$, e.g. "Add specific examples of sensory language",
   (b) Criteria optimization function $O(K, \epsilon)$ with convergence parameter ϵ, e.g. "Update scoring based on review history", and
   (c) Template refinement process $P(T_c, h)$ based on performance history, e.g. "Clarify instructions from common mistakes"

3 Implementation Details

We will implement the entire framework as an edge-deployable application, considering pragmatic constraints of limited edge computing capacity.

3.1 Core Components

(1) **Template Processing:**
   (a) Prompt template definition $T(p, c)$ encoding patterns p and constraints c,
   (b) Validation rule formalization $V(r)$ for rubric r, and
   (c) Edge-compatible transformation protocols

(2) **Validation Framework:**
   (a) Constraint checking $C(i, r)$ for input i,
   (b) Staged response validation $\{v_1, ..., v_n\}$, and
   (c) Boundary enforcement $B(r) \rightarrow \{valid, invalid\}$

(3) **Integration Architecture:**
   (a) State synchronization,
   (b) Atomic storage, and
   (c) Failure recovery

Ongoing implementation with the documentation and example prompts can be seen in the project accessible at https://github.com/build-club-ai-indonesia/edge-prompt GitHub repository.

Teaching materials are taken from https://buku.kemdikbud.go.id Indonesia government education department website.

3.2 Deployment Architecture

The EdgeLLM deployment architecture implements:

(1) Optimized edge runtime for Llama 3.2 3B or similar LLM with minimal resource footprint,
(2) Environment ensuring consistent model behavior,
(3) Storage system for offline operation, and
(4) Validation protocol maintaining the safety constraints.

3.3 Validation Approach

(1) **Functional Metrics:** Evaluating guardrail effectiveness such as valid response ratios, offline stability, operational reliability, and teacher workflow integration.
(2) **Performance Analysis:** Assessing edge deployment capabilities via resource utilization, response latency profiles, and scalability characteristics under varying loads.
(3) **System Validation:** Building on SPADE's guardrail metrics [7], we evaluate edge-specific indicators while maintaining a pure prompt engineering approach without fine-tuning.

4 Discussion

We evaluate other guardrails methodologies such as constrained decoding methods [8] which enforce output restrictions by limiting the model's token choices. However, they require continuous updates to safe/unsafe token lists and can inadvertently restrict creative expression. In contrast, our framework relies on structured prompt engineering and multi-stage validation.

It is important to note that the final prompt design is still under development, and future work will involve an empirical comparison of multiple prompting strategies to determine the most effective approach. Eventually, these may inform subsequent publications.

5 Conclusion and Future Work

In this expression of interest, we presented EdgePrompt, a framework that combines structured prompt engineering with multi-stage validation to enable offline LLM applications in K-12 education. Designed for Indonesia's remote 3T regions, our proposal addresses the twin challenges of unreliable connectivity and limited technical expertise among educators.

While our system has not yet been empirically validated, EdgePrompt lays a strong conceptual foundation. Our next steps include iterative refinement, implementation, pilot deployments, gathering teacher feedback to refine our approach, and optimizing edge performance. Ultimately, our vision is to bridge the gap between advanced LLM capabilities and real-world educational needs, paving the way for scalable, teacher-friendly AI solutions in resource-constrained environments.

Acknowledgments

BuildClub.ai as the training campus for AI learners, experts, builders.

References

[1] Livia Kristianti Raka Adji. 2024. Indonesia's internet penetration hits 79.5 percent, trend continues. *Antara News* (Jan. 2024). https://en.antaranews.com/news/304593/indonesias-internet-penetration-hits-795-percent-trend-continues

[2] Ayan Kumar Bhowmick, Ashish Jagmohan, Aditya Vempaty, Prasenjit Dey, Leigh Hall, Jeremy Hartman, Ravi Kokku, and Hema Maheshwari. 2023. Automating Question Generation From Educational Text. In *Artificial Intelligence XL*, Max Bramer and Frederic Stahl (Eds.). Vol. 14381. Springer Nature Switzerland, Cham, 437–450. doi:10.1007/978-3-031-47994-6_38 Series Title: Lecture Notes in Computer Science.

[3] Yi Dong, Ronghui Mu, Gaojie Jin, Yi Qi, Jinwei Hu, Xingyu Zhao, Jie Meng, Wenjie Ruan, and Xiaowei Huang. 2024. Building Guardrails for Large Language Models. doi:10.48550/ARXIV.2402.01822

[4] Yi Dong, Ronghui Mu, Yanghao Zhang, Siqi Sun, Tianle Zhang, Changshun Wu, Gaojie Jin, Yi Qi, Jinwei Hu, Jie Meng, Saddek Bensalem, and Xiaowei Huang. 2024. Safeguarding Large Language Models: A Survey. doi:10.48550/ARXIV.2406.02622

[5] Ching Nam Hang, Chee Wei Tan, and Pei-Duo Yu. 2024. MCQGen: A Large Language Model-Driven MCQ Generator for Personalized Learning. *IEEE Access* 12 (2024), 102261–102273. doi:10.1109/ACCESS.2024.3420709

[6] Kementerian Desa. 2025. Official Website of the Ministry of Villages. https://www.kemendesa.go.id

[7] Mohammad Niknazar, Paul V Haley, Latha Ramanan, Sang T. Truong, Yedendra Shrinivasan, Ayan Kumar Bhowmick, Prasenjit Dey, Ashish Jagmohan, Hema Maheshwari, Shom Ponoth, Robert Smith, Aditya Vempaty, Nick Haber, Sanmi Koyejo, and Sharad Sundararajan. 2024. Building a Domain-specific Guardrail Model in Production. doi:10.48550/ARXIV.2408.01452

[8] Gabriel Poesia, Oleksandr Polozov, Vu Le, Ashish Tiwari, Gustavo Soares, Christopher Meek, and Sumit Gulwani. 2022. Synchromesh: Reliable code generation from pre-trained language models. doi:10.48550/ARXIV.2201.11227 Version Number: 1.

[9] Lala Septem Riza, Yahya Firdaus, Rosa Ariani Sukamto, Wahyudin, and Khyrina Airin Fariza Abu Samah. 2023. Automatic generation of short-answer questions in reading comprehension using NLP and KNN. *Multimedia Tools and Applications* 82, 27 (Nov. 2023), 41913–41940. doi:10.1007/s11042-023-15191-6

[10] Nicy Scaria, Suma Dharani Chenna, and Deepak Subramani. 2024. Automated Educational Question Generation at Different Bloom's Skill Levels Using Large Language Models: Strategies and Evaluation. In *Artificial Intelligence in Education*, Andrew M. Olney, Irene-Angelica Chounta, Zitao Liu, Olga C. Santos, and Ig Ibert Bittencourt (Eds.). Vol. 14830. Springer Nature Switzerland, Cham, 165–179. doi:10.1007/978-3-031-64299-9_12 Series Title: Lecture Notes in Computer Science.

[11] Xu Wang, Simin Fan, Jessica Houghton, and Lu Wang. 2022. Towards Process-Oriented, Modular, and Versatile Question Generation that Meets Educational Needs. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 291–302. doi:10.18653/v1/2022.naacl-main.22

[12] Joko Widodo. 2020. Peraturan Presiden (PERPRES) Nomor 63 Tahun 2020 Penetapan Daerah Tertinggal Tahun 2020-2024. https://peraturan.bpk.go.id/Details/136563/perpres-no-63-tahun-2020 Publication Title: Database Peraturan | JDIH BPK.