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 K-12 educational settings through structured prompting inspired by neural-symbolic principles. The system addresses educational disparities in Indonesia's Underdeveloped, Frontier, and Outermost (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 extends existing initiatives by implementing 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 requiring 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 *Proceedings of 2nd PromptEng Workshop at the ACM WebConf'25 (PromptEng'25).* ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
PromptEng'25, Sydney, NSW
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2025/04
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Recent advances in guardrail implementations for Large Language Models (LLMs) have demonstrated promising domain-specific control mechanisms [2, 3]. In resource-constrained educational environments, however, implementing effective guardrails requires balancing safety constraints with offline operational capabilities. We define guardrails as "structured prompt-based controls enforcing content boundaries while maintaining validation across edge deployments" through: (1) **Structured Prompting:** templates encoding safety constraints, (2) **Multi-stage Validation:** sequential prompt-based checks, and (3) **Edge Deployment Compatibility:** mechanisms for resource-constrained operation.

Drawing from neural-symbolic architectures [3] while operating within prompt engineering bounds, EdgePrompt implements pragmatic guardrails for K-12 assessment systems that require freeform answers, beyond multiple choice questions [4]. Our framework addresses core technical challenges: (1) maintaining content safety without persistent connectivity, (2) enabling sophisticated assessment in edge deployments, and (3) ensuring consistent validation across distributed components.

The implementation demonstrates offline-capable content filtering through: (1) cloud/edge generation with material-scoped guardrails, (2) edge evaluation supporting offline operation, and (3) atomic storage operations. Initial deployment targets Grade 5 language instruction, emphasizing practical validation in resource-limited settings.

Indonesia has more than 17,000 islands and most of its territory is seas, thus internet penetration is quite challenging, especially for accessing cloud-based LLM tools like ChatGPT. The Indonesian "3T" areas—Frontier, Outermost, and Disadvantaged—face economic, infrastructure, and human resource challenges, requiring targeted government development programs [1, 5]. Edge computing and LLMs can transform education in 3T areas by enabling offline AI-powered learning with minimal internet dependency. LLMs can provide localized, adaptive tutoring, while edge devices enhance access to digital resources, bridging educational gaps and fostering inclusive, remote learning opportunities in underserved regions.

2 Methodology

Our system implements a rigidly structured validation pipeline with strategically placed guardrails that leverage cloud and edge LLMs for distinct operational roles. The architecture enforces safety

through multi-stage template validation, explicit constraint propagation, and formalized evaluation protocols, as shown in Fig. 1. Due to the offline focus of the evaluation infrastructure, we are choosing edge-deployment-capable LLM. We're focusing on the prompt engineering strategy and comparison across different guardrails prompting techniques. The differences between LLM selections are out-of-scope.

2.1 Teacher-Driven Content Generation

(1) **Question Template Definition:** (a) domain-constrained content templates T_c, (b) answer space specification A_s with explicit boundaries, and (c) formal learning objective mapping $O : T_c \rightarrow L$ where L defines permissible learning outcomes.
(2) **Cloud/Edge LLM Assistance Pipeline:** (a) rubric formalization function $R(c_t, v_p)$ where c_t represents teacher criteria and v_p validation parameters, (b) transformation $S : R \rightarrow R'$ ensuring edge compatibility, and (c) grading template generation $G(R')$ with explicit validation constraints.

2.2 Student Answers Evaluation Infrastructure

(1) **Edge Validation Protocol:** (a) verification $V(q, a) \rightarrow \{0, 1\}$ for question-answer pairs alignment, (b) staged response validation sequence $\{v_1, ..., v_n\}$ against rubric R', and (c) boundary enforcement function $B(r) \rightarrow \{valid, invalid\}$ for responses r.
(2) **Evaluation Logic:** (a) application of R' through transformation $E(r, R')$, (b) calibrated scoring function $S(e)$ for evaluation e, and (c) constraint satisfaction verification $C(s, c_t)$ for score s.

2.3 Teacher Verification Protocol

(1) **Response Analysis:** (a) edge case detection function $D(r, \theta)$ with threshold θ, (b) review trigger $T(d) \rightarrow \{review, accept\}$, and (c) calibration state tracking $K(h)$ over evaluation history h.
(2) **System Adaptation:** (a) rubric adjustment $A : R' \rightarrow R''$, (b) criteria optimization function $O(K, \epsilon)$ with convergence parameter ϵ, and (c) template refinement process $P(T_c, h)$ based on performance history.

3 Implementation Strategy

Our technical implementation combines rigorous validation protocols with pragmatic deployment considerations:

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

Example prompts can be seen in the appendix.

3.2 Development Strategy

The deployment strategy would follow several prior similar activities that rely on physical package distribution by using commercial logistics vendors. The general deployment plan for EdgeLLM: (1) build compact all-in-on mini PC that is capable of running on device LLM like LLama 3.2 8B, (2) setup all necessary hardware and software, (3) package it properly and send it via a logistics vendor, and (4) gather online unboxing and hands-on tutorial with the recipient.

3.3 Validation Strategy

Technical validation emphasizes three key aspects:

(1) **Functional Metrics:**
  (a) guardrail effectiveness $E(g) = \frac{\text{valid\_responses}}{\text{total\_responses}}$,
  (b) offline stability $S(t) = \frac{\text{successful\_operations}}{\text{total\_operations}}$, and
  (c) teacher workflow integration rate
(2) **Performance Analysis:**
  (a) edge resource utilization $U(r) = \frac{\text{used\_resources}}{\text{available\_resources}}$,
  (b) response latency distribution $L(t)$, and
  (c) throughput scaling characteristics
(3) **System Insights:** deployment optimization, constraint analysis, and scaling considerations

Our evaluation framework integrates established guardrail effectiveness metrics [6] with edge-specific performance indicators, providing quantitative validation of the system's practical viability. These results will inform subsequent publications exploring comprehensive system evaluation and broader educational applications.

4 Resources

In the appendix, sample prompts can be seen in Fig. Fig. 2 and 3.
Repository: https://github.com/build-club-ai-indonesia/edge-prompt
Data source: https://buku.kemdikbud.go.id/

Acknowledgments

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

References

[1] 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.
[2] 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
[3] 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
[4] 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
[5] Kementerian Desa. 2025. Official Website of the Ministry of Villages. https://www.kemendesa.go.id
[6] 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

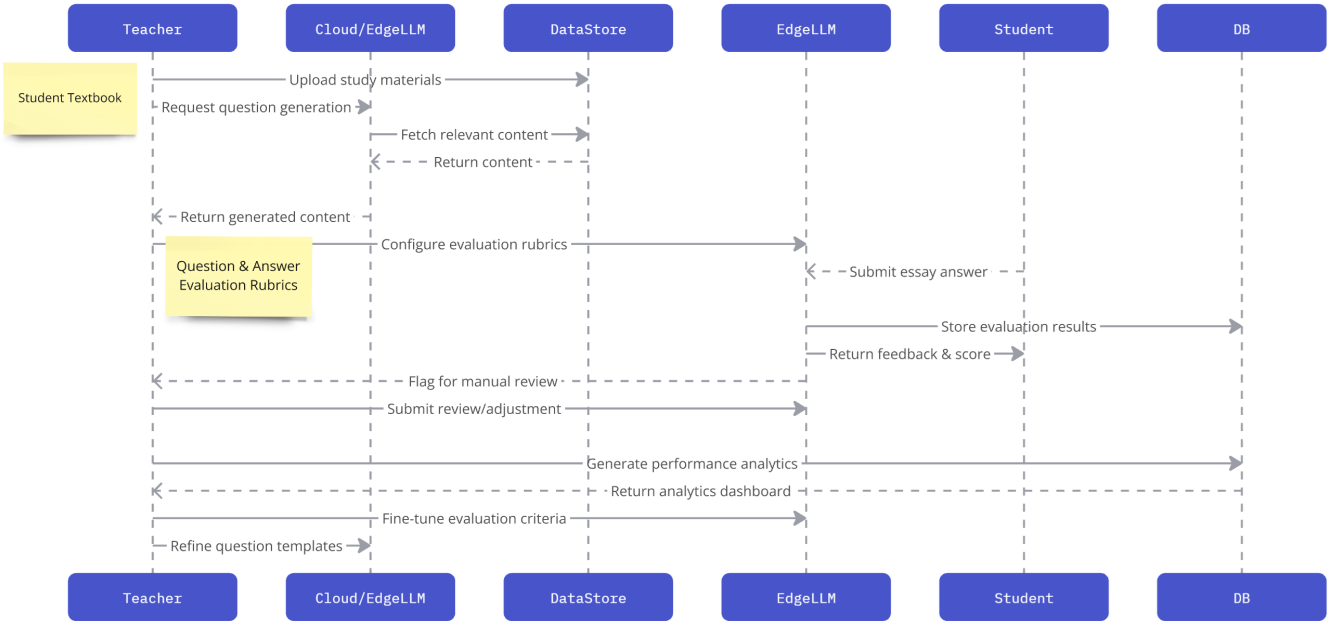

Figure 1: Sequence diagram of the system.

Subject	Topic	Question Type	Sample Prompt	Sample AI-Generated Response

Bahasa Indonesia	**Cerita Fabel**	Pilihan Ganda & Esai	`Buat 5 pertanyaan latihan tentang cerita fabel untuk kelas 5 SD`
			
			`Format: 3 pilihan ganda (4 opsi jawaban) dan 2 soal esai singkat`
			
			`Pertanyaan harus menguji pemahaman tokoh, alur cerita, dan pesan moral.`
				Pilihan Ganda:
				
				1. Siapakah tokoh utama dalam cerita fabel "Kancil dan Buaya"?
				
				• a) Kancil
				• b) Harimau
				• c) Ular
				• d) Ayam
				
				Jawaban:
				
				a) Kancil
				
				Esai: Apa pesan moral dari cerita "Kancil dan Buaya"? Jelaskan dalam 2-3 kalimat.

Figure 2: Sample prompt in Bahasa Indonesia

Subject	Topic	Question Type	Sample Prompt	Sample AI-Generated Response
English	**Simple Past Tense**	Multiple Choice	`Create 5 multiple-choice questions for Grade 5 students to practice the simple past tense. Each question must have 4 answer choices. Focus on using regular and irregular verbs`	**Multiple Choice:** 1. What is the correct past tense form of "go"? • a) goed • b) went • c) go • d) goes **Answer:** **b) went**

Figure 3: Sample prompt in English