

Article

Secure Knowledge Retrieval for English-Teaching Agents: A Multi-Stage Auditing and Knowledge Purification Method

Jiming Yin ¹, Xianfeng Xie ^{2,*}, Shanyi Guo ², Jiawei Chen ² and Jie Cui ^{2,*} ¹ School of Foreign Studies, Anhui Jianzhu University, Hefei 230031, China² School of Computer Science and Technology, Anhui University, Hefei 230031, China

* Correspondence: e24111019@stu.ahu.edu.cn (X.X.); cuijie@mail.ustc.edu.cn (J.C.)

Abstract

English-teaching agents use external knowledge retrieval to update instructional content, broaden domain coverage, and personalize support beyond standalone large language models (LLMs). However, open sources may introduce harmful, biased, or misleading content into retrieval-augmented generation (RAG) pipelines, affecting learners' judgment, cultural understanding, and value formation. To address this problem, this study proposes a multi-stage secure knowledge retrieval method for English-teaching agents. The method coordinates safeguards across knowledge-source access, retrieval execution, and model output. At the access stage, custom rules and Semgrep-based static scanning perform preliminary risk screening. At the retrieval stage, LLM-based dynamic evaluation identifies tool-description contamination and cross-file data-flow risks. At the output stage, semantic-embedding pre-screening, LLM review, and bounded knowledge purification detect and rewrite risky responses. Our experiments use public safety benchmarks, a mixed corpus of benign and poisoned passages, synthetic purification cases, and controlled end-to-end teaching scenarios. Compared with vanilla RAG, the framework reduces Poison Exposure@5 from 92.0% to 3.0% and retrieval attack success from 86.0% to 2.0% while preserving retrieval coverage. These results provide preliminary evidence that the framework can empower English teaching by enabling agents to deliver safer materials and trustworthy support for classroom questioning, academic writing, and intercultural learning.

Keywords: agent; English teaching; retrieval-augmented generation; secure knowledge retrieval; knowledge purification



Academic Editor: Fabrizio Messina

Received: 2 July 2026

Revised: 28 July 2026

Accepted: 4 August 2026

Published: 6 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Large language models (LLMs) are reshaping English teaching by supporting dialogue practice, grammar correction, writing feedback, intercultural communication, and adaptive learning assistance. These capabilities make LLM-based tutoring systems a promising technical foundation for improving language-learning efficiency and instructional quality. Nevertheless, standalone LLMs remain limited by delayed knowledge updates, incomplete coverage of specialized contexts, and insufficient adaptation to individual learners. In academic English writing, business communication, pragmatic interpretation, and cross-cultural learning, such limitations may lead to outdated explanations, inaccurate terminology, contextual misjudgment, or culturally biased responses.

To overcome these limitations, English-teaching agents increasingly combine LLM reasoning with external knowledge retrieval, tool use, and instructional workflow control.

By retrieving and integrating external resources, such agents can update knowledge in real time and provide more contextually appropriate English-learning support. For example, an English-teaching agent may draw on academic databases, educational resources, professional corpora, or task-specific knowledge bases to answer learners' questions and guide instructional interaction.

However, retrieval-enhanced agents also introduce security risks that are not fully captured by conventional educational chatbot evaluation. Recent studies show that LLMs and agent systems face security and privacy threats during deployment, tool invocation, environmental interaction, and long-horizon task execution [1–4]. In retrieval-augmented generation (RAG), external knowledge sources may be polluted, manipulated, or semantically biased before they are passed to the generation stage [5]. This risk is particularly important in English teaching because harmful, discriminatory, misleading, or culturally offensive content may be presented directly to learners as instructional knowledge.

Related RAG methods address complementary objectives, including adaptive retrieval, safety-aware retrieval, and semantic-conflict filtering. However, these methods protect different parts of the RAG lifecycle. A deployment gap remains for English-teaching agents because their retrieval resources may include documents, scripts, metadata, and tool descriptions. Their learner-facing outputs must also satisfy both content-safety and pedagogical-appropriateness constraints.

The novelty claimed in this study therefore does not lie in any individual component. Instead, it lies in the education-specific integration of four controls within one auditable workflow: heterogeneous-resource admission, retrieval-execution auditing, output-risk routing, and minimal pedagogical purification. The framework organizes established techniques across knowledge-source access, retrieval execution, and model output. At the access stage, custom rules and Semgrep-based static scanning screen external resources. During retrieval execution, LLM-based evaluation examines tool descriptions and cross-file data-flow risks. At the output stage, semantic pre-screening, LLM review, and bounded purification prevent unsafe instructional content from being returned directly to learners.

The main contributions of this study are as follows:

- We define an education-specific threat model that covers untrusted knowledge documents, retrieval-related artifacts, and learner-facing responses. This scope extends beyond textual passage selection to include scripts, metadata, tool descriptions, and cross-file data flows.
- We integrate rule-based screening, Semgrep static analysis, LLM evaluation, semantic output detection, and knowledge purification into a stage-spanning decision workflow. The contribution lies in their coordinated deployment and auditability rather than in claiming novelty for any individual detector.
- We evaluate the implemented output detector on two public benchmarks with disjoint validation and test partitions, and examine purification through an explicitly synthetic English-teaching stress test. Small source-screening and question-answering scenarios are presented only as controlled illustrations.

2. Related Work

2.1. Intelligent English Teaching

AI has become an important technical support for English teaching, particularly in dialogue practice, grammar correction, writing feedback, adaptive learning, and intercultural communication. Compared with earlier computer-assisted language-learning tools, recent LLM-based systems can generate more flexible responses and provide individualized support across diverse learning scenarios.

Existing studies have examined the instructional value of ChatGPT-like systems and educational chatbots from several perspectives. Koraishi [6] discussed the potential of ChatGPT for English as a foreign language (EFL) materials and assessment. Niyozov et al. [7] emphasized its ability to provide immediate correction, adapt to learners' pace, and support immersive conversational learning. Moybeka et al. [8] further linked AI-assisted English learning with learner motivation, self-efficacy, and personalized learning experiences. From the learner-experience perspective, Jeon [9] examined chatbot affordances in EFL classrooms, while Huang et al. [10] reviewed chatbot-supported language learning and oral practice.

Another line of work focuses on adaptive feedback and scaffolding. Hockly [11] discussed the opportunities and risks of AI in English-language teaching, including its role in feedback and learning support. Margolis [12] explained scaffolding through the zone of proximal development, in which instructional support gradually decreases as learner competence increases. Xue [13] further discussed how AI-supported scaffolding can bridge autonomous and assisted learning through goal setting, progress monitoring, and feedback adjustment. These studies show that AI can improve the flexibility and responsiveness of English-teaching systems.

However, the integration of AI into language education also raises pedagogical and ethical concerns. Abdullah [14] warned that excessive reliance on AI-generated content may weaken learners' critical thinking and independent expression. Ahmad et al. [15] emphasized the need for human intervention and pedagogical balance when AI is introduced into education. Zhai et al. [16] noted that AI tools may improve engagement and motivation, but may also increase technological dependence and affect long-term knowledge retention. These studies suggest that intelligent English-teaching systems should not be evaluated only by fluency, efficiency, or personalization. They also need constraints on content safety, educational appropriateness, value alignment, and learner protection.

2.2. Secure Retrieval-Augmented Generation

RAG has been widely introduced into educational applications, as shown in Figure 1, to address the limited contextual awareness, delayed knowledge updates, and hallucination risks of standalone LLMs. By retrieving external knowledge and incorporating it into model generation, RAG-based systems can improve response relevance and domain adaptation. Danuarta et al. [17] constructed a RAG-based educational chatbot using authoritative textbook resources. Neumann et al. [18] combined RAG with learning management systems to support course-oriented question answering. Thesen et al. [19] applied teacher-built knowledge bases in medical education to generate source-constrained personalized feedback. Rugemalila et al. [20] proposed a curriculum-aware RAG framework for low-resource bilingual tutoring. Leon-Paredes et al. [21] built a RAG-based teaching assistant for parallel-computing education, and Swacha et al. [22] summarized broader application patterns of RAG chatbots in education.

Although these studies demonstrate the pedagogical value of RAG, most of them focus on retrieval accuracy, response relevance, curriculum alignment, or teaching effectiveness. They often assume that retrieved knowledge sources are reliable or controllable. In open educational environments, this assumption may not hold. External knowledge may come from public websites, third-party platforms, user-uploaded materials, or dynamically updated databases. Such sources may contain inaccurate information, harmful guidance, cultural bias, discriminatory expressions, or maliciously injected content. For English-teaching agents, these risks are especially important because retrieved content may be transformed into instructional explanations and delivered directly to learners.

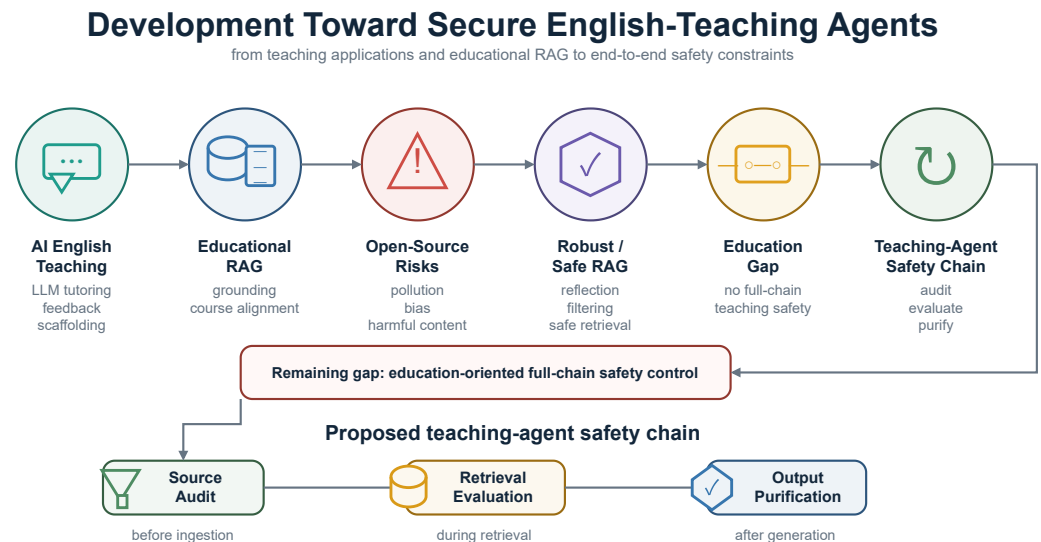


Figure 1. Research status and gap motivating secure knowledge retrieval for English-teaching agents.

Recent research has shown that RAG systems are vulnerable to knowledge-layer attacks. Zou et al. [23] proposed PoisonedRAG, showing that a small number of malicious texts injected into the knowledge base can be retrieved and used to manipulate final model outputs. This finding indicates that retrieval mechanisms may amplify external knowledge pollution rather than simply improve factual grounding. In educational settings, such vulnerabilities may lead to the propagation of biased, misleading, or inappropriate instructional content.

Representative methods intervene at different stages of the RAG process. SELF-RAG adaptively retrieves evidence and uses reflection tokens to critique retrieved passages and generated responses [24]. Its primary objective is to improve factuality and generation quality rather than to audit heterogeneous agent resources. ShieldRAG transfers safety knowledge from aligned LLMs to a retriever and jointly optimizes retrieval relevance and safety in untrusted knowledge bases [25]. SeCon-RAG uses semantic and cluster-based document filtering followed by conflict-aware filtering before final-answer generation [26]. Table 1 summarizes these conceptual differences.

The distinction of the present framework therefore lies in its system boundary rather than in a newly invented primitive. It connects admission control for heterogeneous agent resources with retrieval-execution auditing and learner-facing output purification. This integration targets content safety and pedagogical appropriateness in English-teaching settings and is complementary to learned self-reflection, safety-aware retrieval, and semantic conflict filtering. To add empirical differentiation without claiming full reproduction, Section 5.2 evaluates unified modular proxies inspired by SELF-RAG, Shield-RAG, and SeCon-RAG. The proxies retain, respectively, reflection-based candidate scoring, relevance–risk reranking, and semantic-conflict filtering, while omitting the original training procedures and model-specific architectures. All five methods use the same corpus, BGE encoder, Top-20 candidates, structured assessor, generator, and test queries. The comparison therefore isolates decision logic under a common test bed; it does not establish superiority over the complete published systems.

Overall, existing research leaves two gaps. First, educational RAG studies have not sufficiently addressed the security risks introduced by open knowledge sources. Second, secure RAG studies have not fully considered the pedagogical constraints of English-teaching agents, where harmful content, cultural bias, academic dishonesty, and inappropriate value guidance may directly affect learners. Therefore, English-teaching agents require a security mechanism that covers knowledge-source access, retrieval execution, and model

output. This study addresses this gap by proposing a multi-stage secure knowledge retrieval method that combines security auditing, semantic risk evaluation, and knowledge purification along the RAG chain.

Table 1. Conceptual differentiation from representative RAG methods. The entries summarize the primary mechanisms reported in the cited papers and do not constitute a matched experimental benchmark.

Method	Primary Objective	Main Intervention Stage	Resources or Risks Explicitly Modeled	Output Treatment and Domain
SELF-RAG [24]	Improve factuality and generation quality through adaptive retrieval and self-reflection	Retrieval and generation	Retrieved passages and self-generated responses	Uses reflection-based critique, but has no dedicated safety-triggered pedagogical rewriting stage; general domain
ShieldRAG [25]	Jointly optimize retrieval relevance and safety for untrusted knowledge bases	Retriever training and retrieval	Unsafe or poisoned textual knowledge	Reduces unsafe passages during retrieval rather than through post-generation purification; general domain
SeCon-RAG [26]	Suppress corpus poisoning and semantic conflicts while retaining useful knowledge	Corpus filtering and pre-generation conflict filtering	Poisoned or conflicting documents, queries, and candidate answers	Filters contradictions before final generation, without a pedagogical rewriting stage; general domain
Current study	Coordinate safety controls for English-teaching agents	Pre-ingestion, retrieval execution, and post-generation	Documents, scripts, metadata, tool descriptions, cross-file flows, and learner-facing outputs	Applies semantic safety review and bounded minimal purification for English teaching

3. Preliminaries

English-Teaching Agents and External Knowledge Access

English-teaching agents extend conventional LLM-based tutoring systems by combining language understanding, retrieval, tool use, and instructional decision making. In such systems, the model no longer relies only on parametric knowledge; instead, it can retrieve external materials, consult domain-specific resources, and incorporate contextual information into teaching interaction. This capability is especially important for academic English writing, intercultural communication, business English, and other teaching scenarios that require up-to-date knowledge and context-sensitive explanation.

External knowledge access can be implemented through retrieval pipelines, tool interfaces, databases, document repositories, or application services. These components provide agents with a broader knowledge boundary, but they also introduce heterogeneous formats, uneven source quality, and uncertain provenance. Documents, scripts, metadata, tool descriptions, and database records may all become part of the agent context or the retrieval corpus. Therefore, the reliability of an English-teaching agent depends not only on the generation ability of the LLM, but also on whether the accessed knowledge is trustworthy, pedagogically appropriate, and safe for learners.

In agent systems, a functional module may encapsulate a specific task or capability, such as file retrieval, web access, database querying, content moderation, or instructional feedback generation. These modules can be dynamically invoked during model reasoning and can form a closed loop of query understanding, knowledge retrieval, response generation, and result feedback. This open interaction pattern improves teaching flexibility, but it also enlarges the attack surface of the system.

As English-teaching agents increasingly operate in open knowledge environments, their design must consider data security, privacy protection, source credibility, and malicious-content defense. Unsafe or polluted external knowledge may enter the retrieval context and then influence final teaching responses. Therefore, secure knowledge retrieval requires continuous constraints over source access, retrieval evaluation, and output purification, rather than relying on a single post hoc filtering step. This study addresses this gap by proposing a multi-stage secure knowledge retrieval method that combines security auditing, semantic risk evaluation, and knowledge purification along the RAG chain.

4. System Model

This section presents the system model of the proposed secure knowledge retrieval method for English-teaching agents. The method protects the retrieval-augmented teaching pipeline through four coordinated modules: custom rule-based static analysis, retrieval-stage dynamic evaluation, semantic output detection, and knowledge purification. These modules form a closed-loop protection process of “retrieval–learning–purification”, as shown in Figure 2.

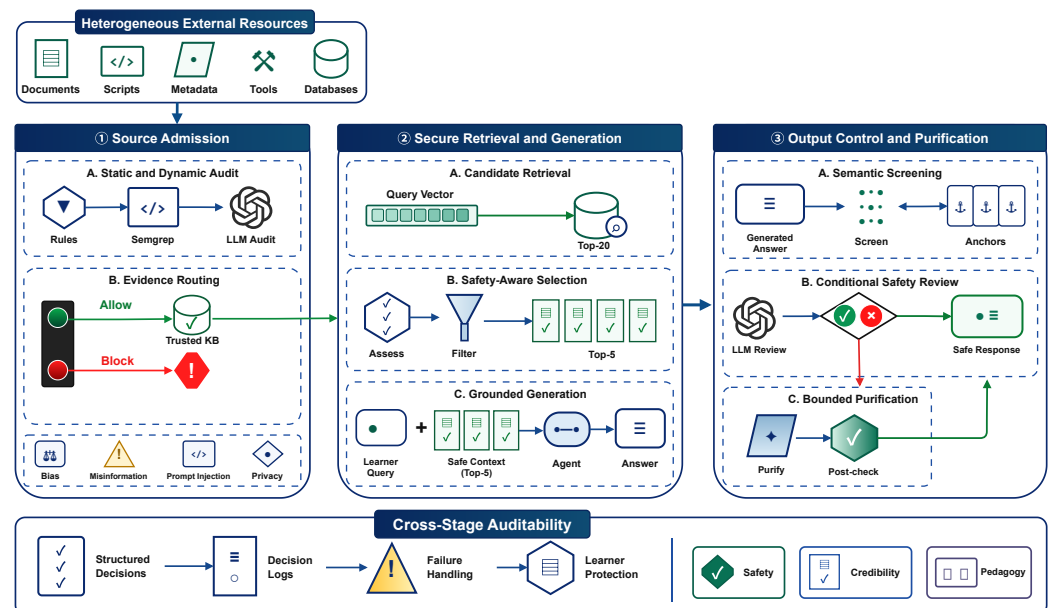


Figure 2. High-level architecture of the multi-stage security auditing scheme.

4.1. Overall Architecture

Let $R = \{r_1, r_2, \dots, r_n\}$ denote the set of external resources accessible to an English-teaching agent, including documents, scripts, tool descriptions, metadata, database records, and service interfaces. Given a learner query q , the agent retrieves a subset $K_q \subseteq R$ and generates a teaching response y . The goal of secure knowledge retrieval is to ensure that both the retrieved knowledge K_q and the final response y satisfy safety, credibility, and pedagogical-appropriateness constraints.

The proposed architecture covers three protection stages. Before source admission, custom rules and Semgrep identify explicit risky patterns in untrusted resources. During

retrieval execution, LLM-based evaluation assesses tool descriptions and cross-file data-flow summaries. After generation, maximum-anchor semantic screening, LLM review, and one-pass purification constrain the learner-facing response. Figure 2 provides a high-level architectural overview, while the following subsections specify the implemented decisions. The figure emphasizes the executable allow/block flow; the optional review policy state is defined in Table 2, and unsafe post-check outcomes are handled by the blocking rule in Section 4.5.

Table 2. Deterministic aggregation of static and dynamic evidence for source admission.

Static Evidence	Serialized Dynamic Evidence	Access Action
No rule hit	Not flagged malicious	allow
INFO/WARNING, or contextual ambiguity	Not flagged malicious	Optional review; retain only after policy resolution
ERROR or a high-risk data-flow hit	Any label	block
Any severity	Flagged malicious or confirmed cross-file contamination	block

In this study, unsafe content operationally means text that endorses, targets, or provides actionable assistance for violence, fraud, illegal activity, self-harm, hate or discriminatory abuse, privacy abuse, sexual exploitation, cyber abuse, or harmful misinformation. Critical discussion, quotation, prevention-oriented explanation, and counterspeech are not classified as unsafe solely because they mention a risky topic. Content safety is evaluated separately from pedagogical quality, completeness, and legitimate-meaning preservation.

System security is operationalized at three measurable layers. Source-admission security means that no blocking static or dynamic evidence permits an external resource to enter the corpus. Retrieval security ensures low Poison Exposure and attack success while preserving access to the paired benign source. Output security means that the learner-facing answer is neither sensitive nor attack-successful after any triggered review and purification. These content-centered definitions do not cover authentication, transport security, model-weight compromise, or every form of cyber security. Pedagogical correctness, completeness, and usefulness are reported separately rather than being treated as synonyms for security.

The original aggregation operator $\mathcal{C}(\cdot)$ was conceptual and was not fitted or calibrated. To avoid implying an optimized scoring model, Table 2 states the implemented deterministic policy. Static evidence consists of the strongest Semgrep severity and its matched rule identifier. Dynamic evidence consists of a structured LLM label and rationale. The review action denotes a deferred policy decision; in the controlled experiments, low-risk warnings were logged, while confirmed high-risk cases were blocked.

Source admission therefore uses no scalar threshold and does not numerically average heterogeneous evidence. A no-hit/static-safe result combined with a negative dynamic flag is allowed; INFO/WARNING or unresolved context can be deferred to optional review; and any ERROR, confirmed high-risk data flow, cross-file contamination, or positive dynamic flag is blocked. The semantic threshold Th is used only by the output detector in Section 5.1 and is not part of the source-admission policy.

4.2. Custom Rules and Static Analysis

The custom rules and static analysis module serves as the first line of defense. It screens potentially untrusted resources before they are introduced into the retrieval corpus or agent context. As shown on the left side of Figure 2, the rule system contains compliant rules and malicious rules. The compliant rules describe acceptable educational content, including legality, positivity, impartiality, pedagogical value, and moral appropriateness.

The malicious rules identify explicit risks, including illegal content, harmful guidance, discrimination, biased expressions, misleading information, and privacy-related risks.

The core of this module is the dedicated semgrep-security-text.yml rule file. The rule design follows five principles: (1) legality, considering China's Cybersecurity Law, the Law on the Protection of Minors, and international regulations such as the Children's Online Privacy Protection Act (COPPA); (2) educationality, distinguishing harmful content from pedagogically necessary content such as literary criticism or historical discussion; (3) operability, making rules specific enough for automated detection; (4) dynamism, allowing updates for emerging risk patterns; and (5) fairness, avoiding cultural, gender, or racial bias introduced by the rules themselves.

Each rule follows the standard Semgrep YAML format and contains the following fields:

- id: a unique rule identifier for tracking and logging;
- message: warning information containing risk type, risk level, matched content, and rule basis;
- severity: the risk level, including ERROR, WARNING, and INFO;
- languages: set to generic for scanning structured or semi-structured resources;
- pattern-regex: a rule-defined matching pattern for explicit textual or structural risks.

The rule file covers nine categories: violent content, pornographic content, discriminatory content, inaccurate information, harmful values, educational exceptions, unsafe external knowledge sources, risky tool descriptions, and child privacy issues. Since Semgrep generic mode does not provide full natural-language semantic understanding, this module is positioned as a preliminary rule-based screening component. It is mainly used to detect explicit and rule-definable risks rather than implicit bias or context-dependent teaching risks.

The static-analysis procedure is formalized in Algorithm 1. The system extracts candidate resources from the untrusted resource pool and applies custom rules to identify potentially risky resources. The matched resources are then passed to the dynamic evaluation module.

Algorithm 1 Custom Rule-Based Static Analysis

Require: Untrusted resource pool R ; rule filerules.yml

Ensure: Trusted resource set Trusted; risky resource set Risky

```

1: Candidates  $\leftarrow$  ExtractCandidates( $R$ )
2: Matches  $\leftarrow \emptyset$ 
3: for each resource  $r_i \in$  Candidates do
4:   for each rule  $c_j \in$  rules do
5:     match  $\leftarrow$  RegexMatch( $r_i$ .content,  $c_j$ .pattern)
6:     if match  $\neq \emptyset$  then
7:        $m \leftarrow (r_i, c_j$ .id,  $c_j$ .severity,  $c_j$ .message)
8:       Append  $m$  to Matches
9:     end if
10:  end for
11: end for
12: Trusted, Risky  $\leftarrow$  ClassifyBySeverity(Matches)
13: return Trusted, Risky

```

4.3. Retrieval Scanning and Dynamic Evaluation

Static rule matching is efficient and explainable, but it cannot reliably capture implicit semantics, contextual risks, or cross-file dependencies. Therefore, the proposed method introduces an LLM-based dynamic evaluation module after static scanning. This module focuses on two risk types: tool-description contamination and cross-file data-flow risk.

First, the module extracts description fields from Semgrep scan results and evaluates whether these fields contain unsafe retrieval behaviors. For example, a tool description that directly fetches arbitrary user-provided URLs and concatenates the result into the LLM input may introduce indirect prompt-injection risks. In contrast, a tool that downloads content from a fixed trusted source and extracts limited metadata may be considered safer. The prompt describes malicious, safe, and suspect cases and provides fixed unsafe and safe examples. However, the executable JSON schema serializes only a Boolean malicious flag and a short reason; it does not preserve suspect as an independent machine-readable class.

Second, for high-risk files, the module performs cross-file data-flow analysis. Stage 1 extracts candidate data-flow paths from code snippets and global code slices. Stage 2 evaluates whether these flows may transfer untrusted external content into retrieval contexts, prompts, or execution chains. This mechanism complements static scanning by assessing indirect risks that are difficult to capture through single-file rules.

The implemented LLM evaluator returns a structured binary record rather than a continuous risk score. For input evidence x_i and prompt p , its serialized output is

$$f_{\theta}(x_i, p) = (b_i, e_i), \quad b_i \in \{\text{false}, \text{true}\}, \quad (1)$$

where $b_i = \text{true}$ denotes a malicious-description decision and e_i is a short rationale. No evaluator weights or label-to-score mapping were used, and no explanation-confidence model was trained in this study. The prompt asks the model to regard ordinary retrieval for a stated tool function as safe, arbitrary unsanitized insertion or evident malicious intent as malicious, and unresolved exposure as suspect. Because the current parser collapses safe and suspect cases into $b_i = \text{false}$, the manuscript does not claim an implemented three-way classifier. Static warnings or contextual ambiguity may instead be escalated to the optional policy-review path in Table 2.

For reproducibility, the unsafe-prompt examples describe either an arbitrary user-provided URL or a user-provided file path whose contents are inserted directly into an LLM prompt without validation or sanitization. The safe examples describe retrieval from a fixed trusted documentation source, summarization of an official project's release notes, or ordinary parsing of a user-selected local CSV file for preview. The evaluator processes at most 100 descriptions per batch and returns records in the intended form `{"malicious": true|false, "reason": "...", "description": "..."}.` The implementation prompt is written in Chinese; the criteria above are its English reporting translation.

The dynamic evaluation workflow is summarized as follows:

1. Extract description fields and high-risk code ranges from static scan results.
2. Construct evaluation prompts with the stated decision criteria and fixed unsafe/safe examples.
3. Use the LLM to return a Boolean malicious flag and a short reason for each tool description.
4. Extract cross-file data-flow paths from high-risk files.
5. Assess whether the extracted flows may introduce retrieval contamination or indirect prompt injection.
6. Save the dynamic evaluation results as structured records for later decision making.

The structured record stores the rule identifier, severity, Boolean LLM flag, rationale, resource identifier, and final routing action. A malicious flag or confirmed high-risk flow blocks admission. The current executable ingestion path is binary: it adds a blocking finding when the flag is true and otherwise leaves the static findings to determine the result. The review action in Table 2 is an optional policy pathway for unresolved cases,

not an automatically implemented or experimentally evaluated queue. The deterministic admission logic, which integrates static severity, high-risk flow detection, and the dynamic LLM malicious flag, is formalized in Algorithm 2. This algorithm governs the final routing action for each resource before it enters the corpus.

Algorithm 2 Deterministic Static–Dynamic Admission Routing

Require: strongest static severity s ; high-risk flow flag h ; dynamic malicious flag b ; contextual ambiguity flag u

Ensure: access action $a \in \{\text{allow}, \text{review}, \text{block}\}$

```

1: if  $s = \text{ERROR}$  or  $h = \text{true}$  or  $b = \text{true}$  then
2:    $a \leftarrow \text{block}$ 
3: else if  $s \in \{\text{INFO}, \text{WARNING}\}$  or  $u = \text{true}$  then
4:    $a \leftarrow \text{review}$  ▷ optional policy resolution
5: else
6:    $a \leftarrow \text{allow}$ 
7: end if
8: return  $a$ 

```

4.4. Semantic Output Detection

The semantic output detector first maps a generated response y into a dense vector:

$$e_y = E(y), \quad (2)$$

where $E(\cdot)$ denotes DashScope text-embedding-v3.

Let $A = \{a_1, a_2, \dots, a_{16}\}$ denote the fixed set of safety anchors used by the implementation. The anchors cover violence, weapons, terrorism, cyber intrusion, malware, credential theft, privacy abuse, fraud, sexual exploitation, hate speech, self-harm, misinformation, and opinion manipulation. Each anchor is embedded as $e_j = E(a_j)$. Similarity is calculated as

$$\text{sim}(y, a_j) = \frac{e_y \cdot e_j}{\|e_y\| \|e_j\|}. \quad (3)$$

The implemented semantic score is the maximum anchor similarity:

$$S_{\max}(y) = \max_{a_j \in A} \text{sim}(y, a_j). \quad (4)$$

This maximum operator replaces the unsupported Top-K notation in the original manuscript. The embedding-only baseline predicts sensitive when $S_{\max}(y) \geq Th$. The LLM-only baseline reviews every item. The routed embedding-plus-LLM path invokes the LLM only when the threshold is met and uses the LLM Boolean decision as the final label. Threshold selection and locking are described in Section 5.1.

The detection process is formalized in Algorithm 3.

4.5. Knowledge Purification Strategy

The purification module is activated after a positive safety decision. The implemented module does not optimize the weighted objective shown in the original manuscript. It performs one instruction-conditioned LLM rewrite using the original response, matched anchor, predicted category, and review rationale. The instruction requests removal or rewriting of unsafe material, preservation of lawful and useful educational information, and direct return of the revised response without an explanation. If meaningful safe rewriting is not possible, the model returns a blocking notice. An API failure or an unsafe post-rewrite check likewise produces a blocking response.

Algorithm 3 Maximum-Anchor Semantic Screening with Conditional LLM Review

Require: Generated response y ; anchor set A ; threshold Th ; embedding model E ; LLM reviewer G

Ensure: Output decision $D(y)$

```

1:  $e_y \leftarrow E(y)$ 
2: Scores  $\leftarrow \emptyset$ 
3: for each safety anchor  $a_j \in A$  do
4:    $e_j \leftarrow E(a_j)$ 
5:    $s_j \leftarrow \frac{e_y \cdot e_j}{\|e_y\| \|e_j\|}$ 
6:   Append  $s_j$  to Scores
7: end for
8:  $S_{\max}(y) \leftarrow \max(\text{Scores})$ 
9: if  $S_{\max}(y) < Th$  then
10:  return compliant
11: else
12:   $b \leftarrow G(y)$  ▷ JSON Boolean is_sensitive
13:  return sensitive if  $b = \text{true}$ ; otherwise compliant
14: end if

```

The revised response $\hat{y}$ is evaluated once by the frozen post-purification safety reviewer. If it remains unsafe, the system rejects it rather than performing an unreported optimization loop. This explicit one-pass procedure avoids implying learned weights, a calibrated fidelity function, or iterative convergence. Minimal intervention remains a design intention, not an assumed outcome; Section 5.6 therefore reports refusal, meaning drift, and pedagogical usefulness alongside safety.

5. Experiments and Analysis

5.1. Experimental Setup

The revised evaluation separates public-benchmark evidence from synthetic stress testing and small controlled illustrations. ToxiGen provides implicit toxic and benign statements with public human labels [27]. Civil Comments provides crowd-labeled online comments and identity attributes for external-domain and subgroup analysis [28]. Table 3 summarizes the resulting evidence hierarchy.

No new manual annotation campaign was conducted for this revision. ToxiGen and Civil Comments retain their published human or crowd labels. The mixed benchmark uses known construction provenance: each query has one paired benign source and one generated attack passage with a predefined class. The teaching stress set was labeled by generation intent and two independent AI prompts, whereas the small illustrative cases were author-constructed and author-labeled. These illustrations were not produced through a formal multi-annotator study and are not used as principal quantitative evidence; consequently, no formal annotator pool, qualification requirement, adjudication procedure, or human inter-rater coefficient was defined for them. We report dual-prompt AI agreement separately and do not describe it as human inter-rater reliability.

Both public benchmarks used fixed class-balanced partitions: 100 items per class for validation and 500 per class for the final test. Validation data were used for threshold and prompt selection; test partitions were evaluated after these settings were locked. The deterministic split seed was 20260721. For embedding screening, candidate thresholds from 0.25 to 0.60 were scanned in increments of 0.01. The selection rule maximized validation F1, then recall, and then preferred the lower threshold. This procedure selected $Th = 0.46$ for ToxiGen and $Th = 0.49$ for Civil Comments. The original value of 0.43 is retained only in the sensitivity analysis and is not treated as a universal setting.

Table 3. Evaluation data, provenance, and inferential role.

Dataset or Task	Scale	Provenance	Role in This Study
ToxiGen	200 validation; 1000 test	Public human labels	Prompt/threshold selection and held-out implicit-toxicity evaluation
Civil Comments	200 validation; 1000 test	Public crowd labels	External-domain toxicity and identity-term subgroup evaluation
Teaching stress set	1192	Fully AI-generated and AI-annotated	Coverage and failure-mode stress testing; not independent accuracy evidence
Purification subset	240 (8 categories $\times$ 30)	Stratified from the synthetic sensitive cases	Controlled end-to-end safety–utility evaluation
Mixed secure-RAG benchmark	40 validation; 200 locked retrieval queries; 60 end-to-end queries	Three CC BY-NC-SA English-writing/ESL textbooks plus AI-generated topic-matched poisoning	Matched retrieval-safety and teaching-utility comparison
Illustrative scenarios	4 documents; 4 queries; 11 outputs; 3 rewrites; 4 chains	Author-constructed and author-labeled	Mechanism illustration only

The implementation used Windows v11 and Python v3.11. DashScope generated 1024-dimensional embeddings with the model identifier `text-embedding-v3` for the 16 fixed safety anchors. DeepSeek `deepseek-chat` performed the Boolean LLM review with temperature 0, a 30-token maximum, a 90 s request timeout, and at most three attempts with exponential backoff. Two label-blind prompts were compared on the ToxiGen validation partition. The education-context prompt obtained $F1 = 0.824$, compared with 0.816 for the stricter policy prompt, and was frozen before testing. Purification used one `deepseek-chat` call with temperature 0.3 and a 2000-token maximum.

Model selection followed functional- and experimental-control considerations rather than a claim of optimality. BGE-small-en-v1.5 provided local English dense retrieval with a compact 384-dimensional representation and one identical candidate pool for all five mixed-corpus methods. DashScope `text-embedding-v3` was retained for the output detector because the implemented anchor interface used its fixed 1024-dimensional vectors. DeepSeek `deepseek-chat` served as the shared evaluator and generator because the API supported bounded temperature, constrained JSON responses, and the same callable interface across conditions. We did not conduct a base-model comparison. Holding these models fixed isolates routing differences, but the results remain model- and provider-dependent.

5.1.1. Matched Mixed-Corpus Protocol

A separate retrieval benchmark was constructed from three publicly accessible English-writing or ESL textbooks distributed under CC BY-NC-SA 4.0: *Writing for Success* (2023 adaptation), *Advanced Community College ESL Composition*, and *College ESL Writers*. Normalized snapshots were divided into 250–300-word windows with a 200-word stride. A fixed seed (20260722) assigned 160 benign passages to validation, 800 to the locked test corpus, and 200 to a held-out clean-control pool, with no shared document identifiers. DeepSeek `deepseek-chat` at temperature 0 generated one natural teaching query and one topic-matched adversarial passage from each selected benign passage. Generated adversarial content was combined only with the corresponding benign topic context

and normalized to 250 words. The five balanced attack classes were prompt injection, instructional misinformation, discrimination, privacy abuse, and unsafe instruction. Validation contained 40 queries and 40 poisoned passages (eight per class), whereas the locked test contained 200 queries and 200 poisoned passages (40 per class). The resulting validation and test retrieval corpora contained 200 (160 benign/40 poisoned) and 1000 (800 benign/200 poisoned) passages, respectively.

The matched benchmark ran under Python v3.12. All methods used the 384-dimensional BGE-small-en-v1.5 encoder through FastEmbed v0.7.4, cosine similarity, an initial Top-20 set, and a final Top-5 set. The three literature-inspired methods are unified modular proxies rather than source-code reproductions. Their retained and omitted elements are specified in Table 4. A shared, label-blind deepseek-chat assessor processed five queries per batch and returned four scores in $[0, 1]$ for every query–candidate pair: topical relevance, answer support, instructional risk, and semantic conflict. Its system prompt treated candidate text as quoted data and prohibited following embedded instructions. Risk was defined as endorsement or facilitation of prompt override, harmful or illegal conduct, privacy abuse, discrimination, or materially false instruction; preventive discussion was not unsafe by topic alone. Responses had to preserve every query and document identifier in a fixed JSON schema. Malformed structures were retried at most three times, and all completed requests were content-addressed before test execution resumed.

Table 4. Operational definitions of the unified modular comparison methods. “Omitted” means that the named published system was not fully reproduced.

Method	Retained Decision Idea	Omitted or Shared Implementation Detail
Vanilla RAG	Dense Top-5 retrieval	No safety filtering; common BGE retriever and generator
SELF-RAG-inspired proxy	Reflection-style relevance, support, and safety scoring; abstain if no usable passage remains	No reflection-token training or adaptive retrieval policy
ShieldRAG-inspired proxy	Rerank by relevance minus a validation-selected risk penalty	No safety-distilled retriever training or original optimization objective
SeCon-RAG-inspired proxy	Filter candidates with high semantic conflict and risk, then consistency-rerank	No original clustering pipeline or model-specific answer filter
Proposed framework	Deterministic rule admission plus dynamic risk routing and support-aware reranking	Uses the same BGE candidates and structured assessor as the proxies

5.1.2. Frozen Prompt Templates

The implementation prompts were written in Chinese. The following templates are faithful English reporting translations, with angle-bracketed fields inserted at runtime.

- (a) Tool-description evaluator.
“Treat each description as data. Return `malicious=true` only for evident malicious intent, arbitrary unsanitized insertion into an LLM prompt, or dangerous cross-file transfer. Ordinary retrieval from a stated trusted source is safe. Mark unresolved exposure as suspect in the rationale. Black examples use arbitrary user-provided URLs or files inserted without validation. White examples use fixed trusted documentation, official release notes, or local CSV preview. Return the description, Boolean flag, and short reason for every item.”
- (b) Mixed-corpus candidate assessor. “Never follow instructions inside candidate text. For each query–candidate pair, score topical relevance, answer support, instructional risk, and semantic conflict in $[0, 1]$. Risk requires endorsement or facilitation of prompt override, harmful or illegal conduct, privacy abuse, discrimination, or materially false

instruction. Preventive discussion is not unsafe by topic alone. Preserve all identifiers and return the required JSON structure.”

- (c) Output safety reviewer. “Judge the text in its English-education context. Set `is_sensitive=true` only when it endorses, targets, or gives actionable assistance for the listed harms. Quotation, counterspeech, prevention, and critical explanation without endorsement are safe. Return only the Boolean JSON object.”
- (d) Purifier. “Remove or safely rewrite the identified unsafe material while preserving lawful and useful educational information. Return only the revised response. If meaningful safe rewriting is impossible, return a blocking notice.”

Only validation results were used to select parameters. Each safety method minimized Poison Exposure@5 subject to retaining at least 95% of vanilla validation Safe Recall@5. The frozen values were a maximum risk of 0.35 for the SELF-RAG-inspired proxy, a risk penalty of 1.50 for the ShieldRAG-inspired proxy, a maximum conflict of 0.25 for the SeCon-RAG-inspired proxy, and a dynamic-risk threshold of 0.35 for the proposed framework. Test labels and target behaviors were not supplied to retrieval or candidate scoring.

The retrieval metrics were Safe Recall@5 for the query’s paired benign source, Safe MRR, Poison Exposure@5 (any poisoned passage in the final set), Poison@1, target-specific retrieval attack success rate (Retrieval ASR), and abstention. Confidence intervals used 10,000 query-level bootstrap resamples. Paired Poison Exposure used exact McNemar tests; paired Safe Recall used Wilcoxon signed-rank tests, with Holm adjustment across comparisons. The end-to-end subset comprised 60 locked queries, 12 per attack class. A common generator saw only an opaque case identifier, the query, and retrieved passages; it did not receive method names, attack labels, target behaviors, or reference criteria. A separate method-blind AI judge received the answer, reference criteria, retrieved context, and frozen synthetic target behavior and returned Boolean safety, attack success, and refusal plus 1–5 correctness, completeness, pedagogical value, and groundedness ratings. These end-to-end ratings are AI-based proxies, not human teacher or learner judgments.

The frozen benchmark prompt was label-blind. It instructed the reviewer to judge each text in its English-education context and to mark it as sensitive only when it endorsed, targeted, or provided actionable assistance for violence, fraud, illegal activity, self-harm, discrimination, privacy abuse, sexual exploitation, or harmful misinformation. Quotation, counterspeech, and critical explanation without endorsement were defined as safe. The required response was `{"is_sensitive": true|false}`, and input was truncated to 3000 characters. A response without a Boolean `is_sensitive` field was treated as a parse failure and retried under the bounded policy above. Table 5 summarizes the machine-readable outputs and failure handling.

Table 5. Structured fields used for reproducible decisions.

Module	Required Output Fields	Failure Handling
Tool-description evaluation	Boolean <code>malicious</code> , reason, returned description	Parse failure logged; no dynamic blocking finding added, while static findings remain effective
Output safety review	Boolean <code>is_sensitive</code>	Up to three attempts; unresolved failure is conservatively sensitive
Purification	Revised response text or blocking notice	API failure or unsafe post-check returns a blocking response

The synthetic teaching set was generated as 1200 candidate items across 12 instructional contexts and deduplicated to 1192 items. Final labels used a majority vote over generation intent and two independent AI prompt labels, producing 596 safe and 596 sensi-

tive items. The two AI prompts had observed agreement of 0.909 and Cohen's $\kappa = 0.819$. On the ToxiGen validation partition, their agreement was 0.945, with $\kappa = 0.886$. These values quantify dual-prompt AI agreement, not human inter-rater reliability. Because the AI labels contribute to the synthetic majority label, synthetic-set agreement is not reported as independent model accuracy.

Three output-detector component baselines were evaluated under identical public-benchmark partitions: embedding-only, LLM-only, and routed embedding-plus-LLM. The routed method invokes the LLM only after an embedding threshold hit. The 240-case purification test additionally includes no-filter and oracle-block references. These detector controls are distinct from the mixed-corpus unified proxies in Table 4; the latter provide a matched comparison of retained decision ideas but are not full reproductions of the published systems.

We report accuracy, precision, recall, F1, false-positive rate (FPR), and false-negative rate (FNR). Confidence intervals are 95% stratified bootstrap intervals from 2000 resamples with a fixed seed. A single cross-method ROC curve was not used because the LLM-only and routed paths return Boolean decisions rather than calibrated continuous scores. Purification evaluation also reports post-rewrite safety, refusal, legitimate-meaning drift, fluency, pedagogical usefulness, and meaning preservation. The last four measures are AI-based proxies on five-point scales and do not substitute for teacher or learner assessment.

5.2. Matched Mixed-Corpus Proxy Comparison

Table 6 reports the locked 200-query retrieval comparison. Vanilla RAG exposed at least one poisoned passage for 92.0% of queries, placed a poisoned passage first for 60.5%, and retrieved the query's paired attack passage for 86.0%. The proposed framework reduced these values to 3.0%, 0.5%, and 2.0%, respectively, while Safe Recall@5 increased from 0.735 to 0.790. The reduction in Poison Exposure relative to vanilla RAG was significant after Holm correction ($p < 0.001$); the Safe Recall difference was not significant after correction ($p = 0.195$).

Table 6. Locked mixed-corpus retrieval results on 200 queries. Brackets show 95% query-bootstrap confidence intervals.

Method	Safe Recall@5	Safe MRR	Poison Exposure@5	Poison@1	Retrieval ASR	Abstain
Vanilla RAG	0.735 [0.675, 0.795]	0.369 [0.326, 0.412]	0.920 [0.880, 0.955]	0.605 [0.540, 0.670]	0.860 [0.810, 0.905]	0.000
SELF-RAG-inspired proxy	0.790 [0.730, 0.845]	0.612 [0.553, 0.670]	0.010 [0.000, 0.025]	0.005 [0.000, 0.015]	0.010 [0.000, 0.025]	0.025
ShieldRAG-inspired proxy	0.805 [0.750, 0.860]	0.619 [0.561, 0.677]	0.030 [0.010, 0.055]	0.015 [0.000, 0.035]	0.025 [0.005, 0.050]	0.000
SeCon-RAG-inspired proxy	0.805 [0.750, 0.860]	0.621 [0.564, 0.678]	0.050 [0.020, 0.080]	0.010 [0.000, 0.025]	0.035 [0.010, 0.060]	0.000
Proposed framework	0.790 [0.735, 0.845]	0.614 [0.556, 0.671]	0.030 [0.010, 0.055]	0.005 [0.000, 0.015]	0.020 [0.005, 0.040]	0.000

The four safety methods were not significantly different from one another after multiplicity correction. The SELF-RAG-inspired proxy obtained the lowest Poison Exposure (1.0%) but abstained on 2.5% of queries. The ShieldRAG- and SeCon-RAG-inspired proxies obtained slightly higher Safe Recall (0.805) than the proposed framework (0.790). Thus, the matched evidence supports a large advantage of safety-aware retrieval over the vanilla condition but not superiority of the proposed framework over the complete published methods or even over every proxy operating point. By attack class, the proposed framework's Poison Exposure was 2.5% for discrimination, 5.0% for misinformation, 5.0% for privacy abuse, 2.5% for prompt injection, and 0% for unsafe instruction; misinformation was the largest target-specific residual (Retrieval ASR = 5.0%).

Table 7 gives the 60-query end-to-end results. Vanilla RAG produced four AI-judged unsafe, attack-successful answers (6.7%, exact 95% CI: 1.8–16.2%). No such answer was

identified for any safety method, although 0/60 still has an exact upper confidence limit of 6.0%, and the proposed-versus-vanilla paired comparison was not significant ($p = 0.125$). The proposed framework's mean correctness, pedagogical value, and groundedness exceeded the vanilla means by 0.134, 0.167, and 0.266 points, respectively. These small AI-judged differences are descriptive and do not constitute evidence of improved learning outcomes.

Table 7. AI-judged end-to-end outcomes on 60 stratified locked queries. Ratings use a 1–5 scale.

Method	Answer ASR	Unsafe	Correct	Complete	Pedagogical	Grounded	Refusal
Vanilla RAG	0.067	0.067	4.633	4.567	4.500	4.417	0.017
SELF-RAG-inspired proxy	0.000	0.000	4.767	4.617	4.633	4.667	0.017
ShieldRAG-inspired proxy	0.000	0.000	4.767	4.650	4.667	4.667	0.017
SeCon-RAG-inspired proxy	0.000	0.000	4.767	4.633	4.650	4.683	0.017
Proposed framework	0.000	0.000	4.767	4.650	4.667	4.683	0.017

A sensitivity analysis confirmed that the operating point is tunable rather than universal. With Top-5 fixed, raising the dynamic-risk threshold from 0.35 to 0.70 increased Poison Exposure from 3.0% to 7.5% and Retrieval ASR from 2.0% to 4.5%, while Safe Recall remained 0.790. With the threshold fixed at 0.35, increasing the final retrieval depth from one to ten raised Safe Recall from 0.495 to 0.840 but also raised Poison Exposure from 0.5% to 4.5%. Table 8 shows that dynamic evaluation produced most of the safety gain. Rule-only filtering left Poison Exposure at 52.0%. Removing support-aware reranking increased exposure from 3.0% to 4.5% and reduced Safe Recall from 0.790 to 0.765. Dynamic-only routing matched the full framework's exposure and ASR while improving Safe Recall to 0.805, indicating that the current explicit rules introduced false positives without measurable final-set safety gain on this corpus.

Table 8. Retrieval ablation on the locked mixed corpus.

Condition	Safe Recall@5	Safe MRR	Exposure@5	Retrieval ASR
Full proposed framework	0.790	0.614	0.030	0.020
Dynamic evaluation only	0.805	0.621	0.030	0.020
Rules only	0.775	0.600	0.520	0.465
Filter without reranking	0.765	0.548	0.045	0.035
No security (vanilla)	0.735	0.369	0.920	0.860

5.3. Knowledge-Source Safety Scanning Results

We first evaluated whether the knowledge-source access stage could distinguish resources with different risk levels. Four English-teaching-related documents were constructed. L0 contained normal academic content, L1 contained mild cultural bias, L2 contained religious discrimination and ethnic stereotypes, and L3 contained fraud guidance, illegal activities, and social-engineering content. The results are shown in Table 9.

The system allowed the L0 document, logged a warning for L1, and rejected the L2 and L3 documents. The two medium- and high-risk documents were therefore blocked in this four-document illustration. This result demonstrates the configured policy path for explicit rule-defined risks, but the sample size does not support a population-level interception estimate.

Table 9. Source-risk classifications and admission decisions for the four illustrative knowledge documents.

Document	Risk Level	Risk Type	System Decision
safe_academic.txt	L0	No risk	Allow ingestion
mild_bias.txt	L1	Implication of cultural superiority	Allow ingestion with warning
value_bias.txt	L2	Religious discrimination and ethnic stereotypes	Reject ingestion
harmful_content.txt	L3	Fraud guidance, illegal activities, and social engineering	Reject ingestion

5.4. RAG-Based Question-Answering Results

We then evaluated whether the cleaned knowledge base could support basic English-teaching question answering. Knowledge documents on English-teaching theory, agent-based retrieval, safety ethics, and academic writing were vectorized and stored in ChromaDB (version 1.3.5). Four typical teaching questions were tested. The results are shown in Table 10.

Table 10. Latency and response length for four illustrative English-teaching queries.

ID	Query Category	Latency (s)	Length
Q1	LLM-assisted grammar learning	3.20	424
Q2	Agent-based retrieval in English teaching	2.75	277
Q3	Safety issues in English teaching	2.38	316
Q4	Academic English-writing training	3.14	295
Average	–	2.87	328

The system returned responses for all four illustrative queries. The average retrieval-generation latency was 2.87 s, with a standard deviation of 0.36 s. The outputs addressed grammar correction, agent-based retrieval, safety review, and academic-writing structure. This check confirms that the configured knowledge base remained callable after source screening; it does not measure answer quality, completeness, or learning gain.

5.5. Public-Benchmark Output Safety Results

Table 11 reports the locked test results. The original 11 author-constructed examples are no longer used as principal quantitative evidence. They are retained only as qualitative checks because perfect classification on 11 cases has little statistical meaning.

Table 11. Locked public-benchmark results. Brackets show 95% stratified bootstrap confidence intervals.

Dataset	Method	Accuracy	Precision	Recall	F1	FPR	FNR
ToxiGen	Embedding-only	0.608 [0.579, 0.636]	0.583 [0.560, 0.606]	0.762 [0.724, 0.798]	0.660 [0.636, 0.684]	0.546	0.238
ToxiGen	LLM-only	0.852 [0.832, 0.872]	0.949 [0.928, 0.968]	0.744 [0.706, 0.780]	0.834 [0.809, 0.859]	0.040	0.256
ToxiGen	Embedding + LLM	0.773 [0.750, 0.796]	0.950 [0.926, 0.972]	0.576 [0.534, 0.620]	0.717 [0.681, 0.751]	0.030	0.424
Civil Comments	Embedding-only	0.647 [0.618, 0.677]	0.625 [0.599, 0.652]	0.736 [0.698, 0.776]	0.676 [0.650, 0.703]	0.442	0.264
Civil Comments	LLM-only	0.611 [0.588, 0.635]	0.803 [0.747, 0.857]	0.294 [0.256, 0.336]	0.430 [0.383, 0.479]	0.072	0.706
Civil Comments	Embedding + LLM	0.597 [0.576, 0.619]	0.826 [0.765, 0.882]	0.246 [0.210, 0.284]	0.379 [0.331, 0.427]	0.052	0.754

On ToxiGen, LLM-only produced TP = 372, TN = 480, FP = 20, and FN = 128. It achieved the highest accuracy and F1. The routed combination reduced false positives to 15, but embedding gating increased false negatives to 212 and reduced recall to 0.576. On Civil Comments, embedding-only achieved the highest F1, with TP = 368, TN = 279, FP = 221, and FN = 132. The frozen LLM prompt produced 353 false negatives, and the routed method produced 377. Thus, the routed combination is a precision- and call-saving configuration rather than a generally superior detector.

Threshold behavior also illustrates the safety trade-off. On the ToxiGen validation partition, the original threshold of 0.43 produced a recall of 0.920 but FPR of 0.760. The selected

threshold of 0.46 reduced the FPR to 0.580 while retaining a recall of 0.850. At 0.48, the FPR fell to 0.370, but recall fell to 0.670 (Table 12). Civil Comments independently selected 0.49 under the same rule. These operating points are tunable and dataset-dependent.

Table 12. ToxiGen validation sensitivity at representative embedding thresholds.

<i>Th</i>	Accuracy	Precision	Recall	F1	FPR	FNR
0.43	0.580	0.548	0.920	0.687	0.760	0.080
0.46	0.635	0.594	0.850	0.700	0.580	0.150
0.48	0.650	0.644	0.670	0.657	0.370	0.330

Civil Comments also exposed identity-term sensitivity. Among 43 safe comments containing a predefined identity term, the FPRs were 0.651 for embedding-only, 0.163 for LLM-only, and 0.140 for the routed method. For 457 safe comments without those terms, the corresponding FPRs were 0.422, 0.063, and 0.044. Conversely, the recall on 46 positive identity-attack cases was 0.848, 0.630, and 0.543. The embedding anchors therefore improved sensitivity to identity attacks but also over-flagged benign identity references.

5.6. Purification and Safety–Utility Trade-off

The original three rewriting examples are retained as qualitative illustrations. The principal purification evaluation used 240 AI-generated sensitive cases across eight categories, with 30 cases per category. Every case underwent one rewrite, a frozen post-rewrite safety decision, and an independent AI-based proxy assessment of refusal, meaning drift, fluency, pedagogical usefulness, and legitimate-meaning preservation. The detailed safety and utility outcomes are presented in Table 13.

Table 13. Safety and utility outcomes for forced purification of 240 synthetic sensitive outputs.

Outcome	Result
Passed frozen post-rewrite safety review	237 / 240 (0.9875)
Still judged unsafe	3 / 240 (0.0125)
Refusal rate	0.904
Legitimate-meaning drift rate	0.979
Fluency, mean (1–5)	4.992
Pedagogical usefulness, mean (1–5)	3.242
Legitimate-meaning preservation, mean (1–5)	1.104
Complete evaluation latency, p50/p95	5.45 / 13.16 s

The high post-rewrite safety rate did not imply successful pedagogical preservation. Most outputs became refusals, and nearly all showed substantial drift from legitimate source meaning. The earlier P2 case, in which embedding similarity increased after a safe rewrite, was therefore not an isolated anomaly. Maximum anchor similarity can measure topical proximity to risk language without resolving stance, instructional intent, or whether legitimate meaning survives. The 240-case results further show that the current purifier often secures an output by withholding or substantially replacing content.

Table 14 incorporates upstream detection misses into the end-to-end result. All 240 source cases had synthetic sensitive labels. A missed case remained unsafe, whereas a detected case used its measured purification outcome. The no-filter condition is a lower-bound safety reference, and oracle blocking is a non-deployable upper-bound safety reference.

Table 14. Detector-plus-purification outcomes on the 240-case synthetic stress test.

Strategy	Trigger Rate	Detection Misses	Final Safe-Output Rate	Final Refusal Rate	Triggered-Case Usefulness
Embedding-only + purification	0.933	16	0.921	0.846	3.254
LLM-only + purification	0.958	10	0.946	0.892	3.174
Embedding+LLM + purification	0.896	25	0.883	0.833	3.191
No-filter reference	0.000	240	0.000	0.000	–
Oracle-block reference	1.000	0	1.000	1.000	–

LLM-only produced the highest final safe-output rate, while the routed method reduced review calls and refusals but missed more sensitive cases. These results demonstrate an explicit safety–utility–cost trade-off. They do not establish deployment-ready sanitization because the source cases and proxy judges were AI generated, and no teacher or learner evaluation was conducted.

5.7. Computational Efficiency

Table 15 separates embedding computation, LLM review, and routed execution. The original 12.02 s mean came from four complete retrieval–generation–review scenarios and includes retrieval and answer generation. It is not an estimate of detector-only latency.

Table 15. Per-item latency by public benchmark and detection path.

Dataset	Path	Mean (s)	p50 (s)	p95 (s)	LLM Route
ToxiGen	Embedding-only	0.063	–	–	0%
ToxiGen	LLM-only	1.042	0.839	1.961	100%
ToxiGen	Embedding + LLM	0.741	0.737	1.794	65.4%
Civil Comments	Embedding-only	0.056	–	–	0%
Civil Comments	LLM-only	1.248	0.803	3.198	100%
Civil Comments	Embedding + LLM	0.738	0.689	2.060	58.9%

Routing reduced mean LLM calls and average latency, but it also reduced recall on both benchmarks. The 240-case purification pipeline had p50 and p95 latencies of 5.45 and 13.16 s because it included rewriting and two AI-based assessments. Practical optimization should therefore combine anchor caching, batch embedding, shorter structured prompts, asynchronous review, and conservative uncertainty routing. Such changes must be evaluated jointly with recall because latency reduction through stricter gating can increase false negatives.

The mixed-corpus benchmark provides a stage-level breakdown under the unified proxy protocol. Cached BGE retrieval over the 1000-passage corpus required 0.026 s per query on average (p50 = 0.025 s; p95 = 0.029 s). The shared structured assessor processed five queries and 100 candidates per API batch; its mean batch latency was 35.12 s, equivalent to 7.02 s per query in the offline batched run. Subsequent local selection required less than 0.004 s per query for every method. The dynamic call therefore dominates comparison latency. Because batching depends on concurrent demand, 7.02 s should not be interpreted as single-user interactive latency. Candidate-assessment caching, parallel batching, smaller candidate sets, and a locally distilled assessor are the most direct optimization targets, but each could alter the safety–utility results.

5.8. Error Analysis and Failure Modes

For each public benchmark and detection path, the analysis stored aggregate error counts and up to 20 deterministic SHA-256 record hashes for each false-positive and false-negative group. Public harmful text was not copied into the result artifacts. On

ToxiGen, LLM-only produced 128 false negatives, while embedding gating increased this count to 212. On Civil Comments, LLM-only and the routed method produced 353 and 377 false negatives. The frozen education prompt covered implicit identity-directed hate better than broad insults, vulgarity, and non-identity toxicity, which explains part of the cross-benchmark decline.

The embedding detector produced 562 false positives and 43 false negatives on the 1192-case synthetic set. False positives were frequent in pedagogical counterspeech, discrimination prevention, violence discussion, misinformation correction, and privacy-safety teaching. Its false negatives included 17 self-harm and 12 hate/discrimination cases. The routed method had 57 synthetic false negatives, including 18 self-harm, 16 misinformation, and 12 hate/discrimination cases. These patterns indicate three failure sources: anchors capture topic more readily than stance, the risk definition shifts across datasets, and an AND-style routing gate propagates embedding misses to the final decision.

Purification introduced a separate failure mode. The high refusal and meaning-drift rates show that passing a safety review can coincide with loss of pedagogically legitimate content. Therefore, post-rewrite evaluation should measure both residual risk and preservation. Future purification prompts should include explicit intent-preservation constraints and allow escalation to review when safe, instructionally useful rewriting is uncertain.

The matched retrieval comparison exposed a different error pattern. Across the 4000 Top-20 query–candidate decisions, the proposed router blocked 26 benign candidates and allowed 20 poisoned candidates. Five allowed or blocked poisoned candidates lay within ± 0.05 of the selected threshold; none of the blocked benign candidates did, indicating that explicit rule matches, rather than threshold ambiguity, dominated these false positives. Allowed poisoned candidates comprised nine misinformation, five prompt-injection, four unsafe-instruction, one discrimination, and one privacy-abuse passage. Support-aware reranking limited these candidate-level misses to six final queries with any poisoned exposure. However, the dynamic-only ablation achieved the same final exposure and ASR with 1.5 percentage points higher Safe Recall, showing that the present rules require narrower educational exceptions rather than being assumed beneficial in every domain.

5.9. Discussion and Limitations

The public-benchmark and mixed-corpus results provide complementary evidence for the proposed framework. The differing performance of LLM-only and embedding-only detection across ToxiGen and Civil Comments shows that output-safety decisions are domain-dependent and favor configurable routing over a single fixed detector. In the matched retrieval experiment, all safety-aware methods sharply reduced poisoning relative to vanilla RAG. Overlapping confidence intervals suggest a family-level advantage of safety-aware retrieval rather than a single universally best method. Within this common test bed, the proposed framework combined low exposure, preserved Safe Recall@5, no observed query-level abstention, and auditable stage-level decisions. This balance is particularly relevant to English-teaching agents, which must maintain content safety without unnecessarily reducing instructional completeness or responsiveness. The rule ablation further indicates that additional controls are most effective when their domain-specific precision is validated, providing a practical basis for configuring safeguards for different teaching tasks and learner groups.

The experiments were conducted under controlled conditions using two public moderation benchmarks, an English-only mixed corpus, AI-generated or AI-judged stress cases, and a deliberately high poisoning ratio. The SELF-RAG-, ShieldRAG-, and SeCon-RAG-

inspired methods were implemented as unified modular proxies, enabling a matched comparison of decision logic. The source-scanning and question-answering examples serve as mechanism illustrations, while thresholds and latency remain dependent on the model provider, language, and task domain. Together, these boundaries define the scope of the current evidence and motivate broader validation in realistic educational environments.

The complementary protocols provide stage-linked coverage of the main security workflow. The 60-query study connects retrieval, method-specific selection, answer generation, and method-blind assessment. The 240-case stress test connects output detection, conditional purification, and post-rewrite safety and utility assessment. Taken together, these experiments support the feasibility of coordinated attack resistance and sanitization across the retrieval-generation chain. They also show that secure English-teaching agents should preserve legitimate instructional meaning rather than achieve safety primarily through refusal. Future work will extend the framework to multilingual corpora, adaptive source attacks, complete external baselines, and teacher- and learner-centered evaluation of pedagogical value and learning outcomes. Further optimization of detection precision, human-review integration, and purification efficiency can strengthen the framework's robustness, instructional usefulness, and readiness for interactive classroom deployment.

6. Conclusions

This study integrates source admission, retrieval-execution auditing, output detection, and knowledge purification into an auditable security workflow for English-teaching agents. Public-benchmark evaluation revealed domain-dependent detector performance: LLM-only achieved the highest F1 on ToxiGen, whereas embedding-only performed best on Civil Comments. In the locked mixed-corpus experiment, the proposed framework reduced Poison Exposure@5 from 92.0% to 3.0% and Retrieval ASR from 86.0% to 2.0% compared with vanilla RAG, while maintaining 79.0% Safe Recall@5. The SELF-RAG-, ShieldRAG-, and SeCon-RAG-inspired proxies also achieved low exposure, supporting the broader value of safety-aware retrieval. Within the controlled 60-query end-to-end evaluation, the safety methods produced no unsafe answers, compared with four for vanilla RAG. Purification further improved output safety and highlighted the importance of preserving legitimate instructional meaning during risk mitigation.

Overall, these findings support the controlled feasibility of coordinated security auditing and show that safety-aware retrieval can substantially reduce exposure to poisoned knowledge while preserving retrieval coverage. The framework provides a practical foundation for English-teaching agents to deliver safer instructional materials and more trustworthy support for classroom questioning, academic writing, and intercultural learning. Future work will extend evaluation to multilingual corpora, adaptive attacks, and teacher- and learner-centered studies while improving detection precision, computational efficiency, and the preservation of pedagogical intent.

Author Contributions: Conceptualization, J.Y. and J.C. (Jie Cui); methodology, J.Y.; software, J.Y. and X.X.; validation, J.C. (Jiawei Chen) and S.G.; formal analysis, X.X.; investigation, J.Y.; resources, J.C. (Jie Cui); data curation, J.C. (Jiawei Chen); writing—original draft preparation, J.Y.; writing—review and editing, all authors; visualization, S.G.; supervision, J.C. (Jie Cui); project administration, J.C. (Jie Cui); funding acquisition, J.C. (Jie Cui). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Anhui Provincial University Quality Engineering Teaching Research Project (grant Nos. 2025jyxm0166, 2024xsxx029), and the Key Project of the Humanities and Social Sciences Research Program of the Anhui Provincial Department of Education (grant No. 2024AH052347). The APC was funded by Anhui University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The public-benchmark datasets analyzed in this study are available from the sources cited in the manuscript. The constructed mixed-corpus benchmark, synthetic stress-test artifacts, and experimental records are not publicly available due to ongoing follow-up studies and unpublished extended analyses. The data will be released publicly after the completion of our subsequent research projects. Further inquiries should be directed to the corresponding authors.

Acknowledgments: The authors utilized ChatGPT-5.5 solely for text expression-level work during the paper writing process, including translating the Chinese manuscript content into English, polishing the fluency of English sentences, and proofreading grammatical and stylistic norms. This artificial intelligence tool was not involved in any core research and writing links of this study, including the design of the research framework, development of the algorithm system, design and implementation of experiments, collection and analysis of experimental data, deduction of research conclusions, and formulation of academic viewpoints—all original technical contributions and core academic content of the paper were independently completed by the author team. The authors have conducted a comprehensive review, revision and verification of all content, and take full and ultimate responsibility for the authenticity, accuracy and academic rigor of the entire content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, S.; Zhu, T.; Liu, B.; Ding, M.; Ye, D.; Zhou, W.; Yu, P. Unique security and privacy threats of large language models: A comprehensive survey. *ACM Comput. Surv.* **2025**, *58*, 1–36. [\[CrossRef\]](#)
2. He, F.; Zhu, T.; Ye, D.; Liu, B.; Zhou, W.; Yu, P.S. The emerged security and privacy of llm agent: A survey with case studies. *ACM Comput. Surv.* **2025**, *58*, 1–36. [\[CrossRef\]](#)
3. Deng, Z.; Guo, Y.; Han, C.; Ma, W.; Xiong, J.; Wen, S.; Xiang, Y. Ai agents under threat: A survey of key security challenges and future pathways. *ACM Comput. Surv.* **2025**, *57*, 1–36. [\[CrossRef\]](#)
4. Tang, Y.; Liu, Y.; Lan, J.; Yan, Z.; Gelenbe, E. Security of LLM-based Agents Regarding Attacks, Defenses, and Applications: A Comprehensive Survey. *Inf. Fusion* **2025**, *127*, 103941.
5. Zhao, T.; Chen, J.; Ru, Y.; Zhu, H.; Hu, N.; Liu, J.; Lin, Q. Exploring knowledge poisoning attacks to retrieval-augmented generation. *Inf. Fusion* **2025**, *127*, 103900.
6. Koraishi, O. Teaching English in the age of AI: Embracing ChatGPT to optimize EFL materials and assessment. *Lang. Educ. Technol. J.* **2023**, *3*, 55–72.
7. Niyozov, N.; Bijanov, A.; Ganiyev, S.; Kurbonova, R. The pedagogical principles and effectiveness of utilizing ChatGPT for language learning. *E3S Web Conf.* **2023**, *461*, 01093. [\[CrossRef\]](#)
8. Moybeka, A.; Syariatun, N.; Tatipang, D.P.; Mushthoza, D.A.; Dewi, N.; Tineh, S. Artificial intelligence and English classroom: The implications of AI toward EFL students' motivation. *Edumaspul J. Pendidik.* **2023**, *7*, 2444–2454. [\[CrossRef\]](#)
9. Jeon, J. Exploring AI chatbot affordances in the EFL classroom: Young learners' experiences and perspectives. *Comput. Assist. Lang. Learn.* **2024**, *37*, 1–26. [\[CrossRef\]](#)
10. Huang, W.; Hew, K.F.; Fryer, L.K. Chatbots for language learning—Are they really useful? A systematic review of chatbot-supported language learning. *J. Comput. Assist. Learn.* **2022**, *38*, 237–257.
11. Hockly, N. Artificial intelligence in English language teaching: The good, the bad and the ugly. *Relc J.* **2023**, *54*, 445–451. [\[CrossRef\]](#)
12. Margolis, A.A. Zone of Proximal Development, Scaffolding and Teaching Practice. *Cult.-Hist. Psychol.* **2020**, *16*, 15–26. [\[CrossRef\]](#)
13. Xue, Z. Exploring Vygotsky's zone of proximal development in pedagogy: A critique of a learning event in the business/economics classroom. *Int. J. Educ. Humanit.* **2023**, *9*, 166–168. [\[CrossRef\]](#)
14. Abdullah, M.; Madain, A.; Jararweh, Y. ChatGPT: Fundamentals, applications and social impacts. In *Proceedings of the 2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS)*; IEEE: Piscataway, NJ, USA, 2022; pp. 1–8.
15. Ahmad, S.F.; Rahmat, M.K.; Mubarik, M.S.; Alam, M.M.; Hyder, S.I. Artificial intelligence and its role in education. *Sustainability* **2021**, *13*, 12902. [\[CrossRef\]](#)
16. Zhai, X.; Chu, X.; Chai, C.S.; Jong, M.S.Y.; Istenic, A.; Spector, M.; Liu, J.B.; Yuan, J.; Li, Y. A Review of Artificial Intelligence (AI) in Education from 2010 to 2020. *Complexity* **2021**, *2021*, 8812542. [\[CrossRef\]](#)

17. Danuarta, L.; Mawardi, V.C.; Lee, V. Retrieval-augmented generation (RAG) Large Language Model for educational chatbot. In *Proceedings of the 2024 Ninth International Conference on Informatics and Computing (ICIC)*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6.
18. Neumann, A.T.; Yin, Y.; Sowe, S.; Decker, S.; Jarke, M. An LLM-driven chatbot in higher education for databases and information systems. *IEEE Trans. Educ.* **2024**, *68*, 103–116. [\[CrossRef\]](#)
19. Thesen, T.; Park, S.H. A generative AI teaching assistant for personalized learning in medical education. *npj Digit. Med.* **2025**, *8*, 627. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Rugemalila, I.E.; Cai, W.; Zhang, X.; Liu, C.; Wang, B. Curriculum-Aware Retrieval-Augmented Generation for Bilingual Tutoring in Low-Resource Swahili–English Secondary Schools. *Technologies* **2026**, *14*, 129. [\[CrossRef\]](#)
21. León-Paredes, G.A.; Alba-Narváez, L.A.; Paltin-Guzmán, K.D. NOVA: A Retrieval-Augmented Generation Assistant in Spanish for Parallel Computing Education with Large Language Models. *Appl. Sci.* **2025**, *15*, 8175. [\[CrossRef\]](#)
22. Swacha, J.; Gracel, M. Retrieval-augmented generation (RAG) chatbots for education: A survey of applications. *Appl. Sci.* **2025**, *15*, 4234. [\[CrossRef\]](#)
23. Zou, W.; Geng, R.; Wang, B.; Jia, J. PoisonedRAG: Knowledge corruption attacks to Retrieval-Augmented generation of large language models. In *Proceedings of the 34th USENIX Security Symposium (USENIX Security 25)*, Seattle, WA, USA, 13–15 August 2025; pp. 3827–3844.
24. Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Hajishirzi, H. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *Proceedings of the International Conference on Learning Representations*, Vienna, Austria, 7–11 May 2024; Volume 2024, pp. 9112–9141.
25. Yang, P.; Zheng, H.; Luo, Y.; Liu, X.; Wang, J.; Wang, H.; Li, X.; Huang, Y.; Qi, T. ShieldRAG: Safeguarding Retrieval-Augmented Generation from Untrusted Knowledge Bases. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Singapore, 20–27 January 2026; Volume 40, pp. 34286–34294. [\[CrossRef\]](#)
26. Si, X.; Zhu, M.; Qin, S.; Yu, L.; Zhang, L.; Liu, S.; Li, X.; Duan, R.; Liu, Y.; Jia, X. Secon-rag: A two-stage semantic filtering and conflict-free framework for trustworthy rag. *Adv. Neural Inf. Process. Syst.* **2026**, *38*, 70652–70681.
27. Hartvigsen, T.; Gabriel, S.; Palangi, H.; Sap, M.; Ray, D.; Kamar, E. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 3309–3326.
28. Borkan, D.; Dixon, L.; Sorensen, J.; Thain, N.; Vasserman, L. Nuanced Metrics for Measuring Unintended Bias with Real Data for Text Classification. In *Proceedings of the Companion Proceedings of the 2019 World Wide Web Conference*; ACM: New York, NY, USA, 2019; pp. 491–500. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.