2025 IEEE 49th Annual Computers, Software, and Applications Conference (COMPSAC) | 979-8-3315-7434-5/25/$31.00 ©2025 IEEE | DOI: 10.1109/COMPSAC65507.2025.00024

StuLAC: An Adaptive LLM-Driven Framework for Scalable Student Feedback Analysis in Software-Driven Educational Systems

Yicheng Sun, Hi Kuen Yu, Jacky Keung, Yuchen Cao, and Yihan Liao
Department of Computer Science, City University of Hong Kong, Hong Kong, China
{yicsun2-c, hikuenyu2-c, cynthcao2-c, yihanliao4-c}@my.cityu.edu.hk, jacky.keung@cityu.edu.hk

***Abstract*—With the growing scalability challenges in higher education, automated student feedback analysis has become crucial for course evaluation and pedagogical improvements. However, traditional methods struggle to handle mixed sentiments, adapt to evolving feedback trends, and maintain computational efficiency. To address these challenges, we propose StuLAC, a Software Engineering-driven framework that integrates Large Language Models (LLMs) with Adaptive Template-Based Caching (ATC). StuLAC employs hierarchical matching for fine-grained classification and dynamically updates feedback templates through context-aware cache refinement. Empirical results on 80,000 student feedback entries demonstrate that StuLAC-generated summaries improve overall quality by 10.5% compared to manually generated reports, while also achieving faster processing times. Additionally, StuLAC attains an 86.4% accuracy and an 86.24% F1-score in sentiment detection. StuLAC's Feedback Summary Generation provides actionable insights that enhance data-driven decision-making in educational settings. These findings establish StuLAC as a scalable and adaptive solution for improving AI-driven educational feedback systems.**

***Index Terms*—LLM-Driven Feedback Analysis, Adaptive Caching in NLP, Educational Data Engineering, Scalable AI for Academic Evaluation, Software-Optimized Sentiment Classification**

I. Introduction

The rapid expansion of higher education has led to an exponential increase in student feedback data, necessitating scalable, automated analysis techniques [1]. Traditional feedback evaluation methods, often relying on manual review or rudimentary keyword-based sentiment analysis, struggle with handling large-scale, complex, and mixed-sentiment responses [2], [3]. Additionally, static classification models fail to adapt to evolving feedback patterns, leading to poor generalization and inefficiencies in educational decision-making [4].

The increasing volume and diversity of student feedback data pose significant challenges for conventional feedback analysis methods. Traditional approaches [1], [2], [5], [6], often based on keyword matching or basic sentiment analysis, struggle to effectively categorize and interpret complex and nuanced feedback entries. For example, a feedback entry such as "*The instructor was very clear, but some topics were confusing.*" presents both positive and negative sentiments, which are difficult to classify using basic methods [7]. Additionally, many existing systems are not adaptable enough to handle evolving feedback patterns and emerging themes, leading to rigid frameworks that are not responsive to the dynamic nature of student feedback [8]. Without a robust and adaptive system, it becomes increasingly difficult to process, categorize, and analyze feedback as new topics and expressions emerge.

Recent advancements in language models, such as ChatGPT [9], offer a promising solution to the challenges in student feedback analysis [10]. ChatGPT's language understanding and context-handling capabilities enable more nuanced sentiment detection and dynamic refinement of feedback templates, making it well-suited for managing complex feedback and adapting to evolving patterns in student input [11], [12]. While the use of such models in educational feedback systems is still emerging, their potential to enhance feedback accuracy and adaptability is significant, particularly in handling mixed sentiments and evolving themes [13]. However, limited research has explored the integration of ChatGPT into these systems, highlighting a need for further investigation to fully leverage its capabilities in educational settings.

To address these challenges, we propose the Student Feedback Analysis framework using LLMs with Adaptive Template-Based Cache (ATC), named **StuLAC**. StuLAC integrates the semantic capabilities of Large Language Models (LLMs) [14] with a dynamic, adaptive caching system to efficiently classify and analyze student feedback. The framework uses a hierarchical matching approach that categorizes feedback into coarse and fine levels, allowing it to handle complex, long-form feedback with multiple themes or mixed sentiments. By leveraging ChatGPT [9] for cache updating, StuLAC dynamically generates or refines templates, adapting to emerging feedback patterns and ensuring responsiveness to diverse inputs. Additionally, the Feedback Summary Generation component consolidates key insights—such as sentiment trends, category distributions, and actionable suggestions—into clear, data-driven reports that inform decision-making and enhance educational outcomes. In summary, our primary contributions can be summarized in four-fold:

- We introduce an adaptive template-based caching mechanism that dynamically updates and refines templates in response to new feedback patterns, combining LLMs with adaptive caching for scalable, efficient feedback management.
- We apply a hierarchical matching approach that improves classification accuracy by handling long feedback entries

2836-3795/25/$31.00 ©2025 IEEE
DOI 10.1109/COMPSAC65507.2025.00024

with mixed sentiments, increasing granularity in feedback analysis.
- We utilize In-Context Learning (ICL) with ChatGPT to generate and differentiate templates, capturing nuanced feedback without redundancy.
- The Feedback Summary Generation component synthesizes sentiment trends, category distributions, and actionable insights, providing clear, data-driven reports to support decision-making in educational settings.

II. Methodology

A. Overview

The **Adaptive Template-Based Cache (ATC)** in the StuLAC framework efficiently categorizes and analyzes diverse student feedback through hierarchical template-matching and caching, adapting dynamically to new feedback patterns. As shown in Figure 1, the ATC operates in three phases: **initialization**, **cache matching**, and **cache updating**. In the initialization phase, feedback is clustered to identify distinct patterns, from which an initial set of templates is manually curated. These templates are defined by key features, including extracted keywords and sentiment terms, serving as the foundation for future matching. To handle complex, long-form feedback, we employ a hierarchical matching approach with coarse and fine classification. The first level assigns feedback to broad categories, such as learning attitude or teaching content, based on overall themes, while the second level further classifies it into sub-templates for more detailed aspects. This layered approach ensures both broad and nuanced aspects of feedback are captured. During cache matching, incoming feedback is processed using cosine similarity of sentence embeddings (e.g., RoBERTa) to match new entries to existing templates based on a predefined threshold; unmatched feedback triggers cache updating, where ChatGPT generates or refines templates to accommodate emerging patterns. Finally, **Feedback Summary Generation** compiles key statistics—such as category distribution, sentiment trends, and feedback patterns—into a structured report generated by ChatGPT, offering educators and administrators a comprehensive overview to inform decision-making. This adaptive mechanism ensures the ATC remains responsive, efficient, and comprehensive over time.

B. Initialization

The initialization phase in the StuLAC framework is essential for constructing an initial set of templates that capture the diversity of student feedback patterns. We employ the **DBSCAN** [15] algorithm, chosen for its ability to detect clusters of varying densities and its robustness to noise, making it particularly well-suited for handling noisy, unstructured feedback data. One key advantage of DBSCAN is that it does not require the number of clusters to be pre-defined, allowing it to naturally identify clusters based on the density of data points. In DBSCAN, each feedback entry is embedded into a high-dimensional space using sentence embeddings, denoted as $\mathbf{v}_i$ for the i-th feedback. The algorithm uses two parameters: the radius ϵ for neighborhood distance and the minimum number of points $MinPts$ required to form a cluster [16]. Two feedback embeddings $\mathbf{v}_i$ and $\mathbf{v}_j$ are considered part of the same cluster if:

$$\|\mathbf{v}_i - \mathbf{v}_j\| \leq \epsilon \quad \text{and} \quad \text{density}(\mathbf{v}_i) \geq MinPts. \tag{1}$$

This clustering approach allows us to group feedback based on natural patterns, capturing thematic groups such as "positive attitudes toward teaching," "constructive criticism of materials," or "requests for additional resources," and offers a flexible, noise-resistant solution for initial template generation.

To handle the complexity of long feedback entries, we employ a hierarchical matching method [17] with coarse and fine classification [18]. This method allows us to categorize complex, multi-part feedback more effectively by breaking it down into broad themes at the coarse level and refining it into more specific aspects at the fine level. In the initialization phase, feedback clusters are reviewed, and representative feedback sentences are selected to serve as coarse-level templates. These coarse templates are then further refined during the fine classification stage to capture more detailed feedback aspects. For example, a feedback sentence like "*The instructor was clear, but the material was quite complex.*" is first matched to a coarse template for "Instructor Clarity" and subsequently assigned to the fine template "Course Material Difficulty," accurately capturing both positive and negative sentiments.

This two-level classification ensures that the framework handles both broad themes and nuanced details in feedback, improving the precision of categorization. Each template is defined by key features, such as extracted keywords and sentiment markers, which facilitate the matching of new feedback entries. For instance, the "Instructor Clarity" template may include keywords like "clear", "well-explained", and "helpful", with a positive sentiment label. In contrast, the "Course Material Difficulty" template might include keywords such as "difficult", "complex", and "hard to follow", paired with a neutral-to-negative sentiment. This hierarchical approach enhances the framework's ability to categorize feedback entries with mixed sentiments or multiple themes, ensuring more accurate and comprehensive feedback analysis.

To illustrate further, consider the feedback templates and examples in Table I. The hierarchical matching method processes each feedback entry in two stages: at the coarse level, feedback is assigned to broad categories such as *Student Engagement* or *Course Pacing*, based on the overall sentiment. At the fine level, the feedback is further categorized into specific sub-templates. For instance, the feedback "*The material was quite challenging and sometimes difficult to follow, but the instructor was clear in their explanations.*" is first matched to the *Instructor Clarity* template and then refined with the *Course Material Difficulty* template, accurately capturing both positive and negative sentiments. This two-tiered approach ensures precise categorization of feedback, regardless of its length or complexity, maintaining high accuracy in sentiment detection.

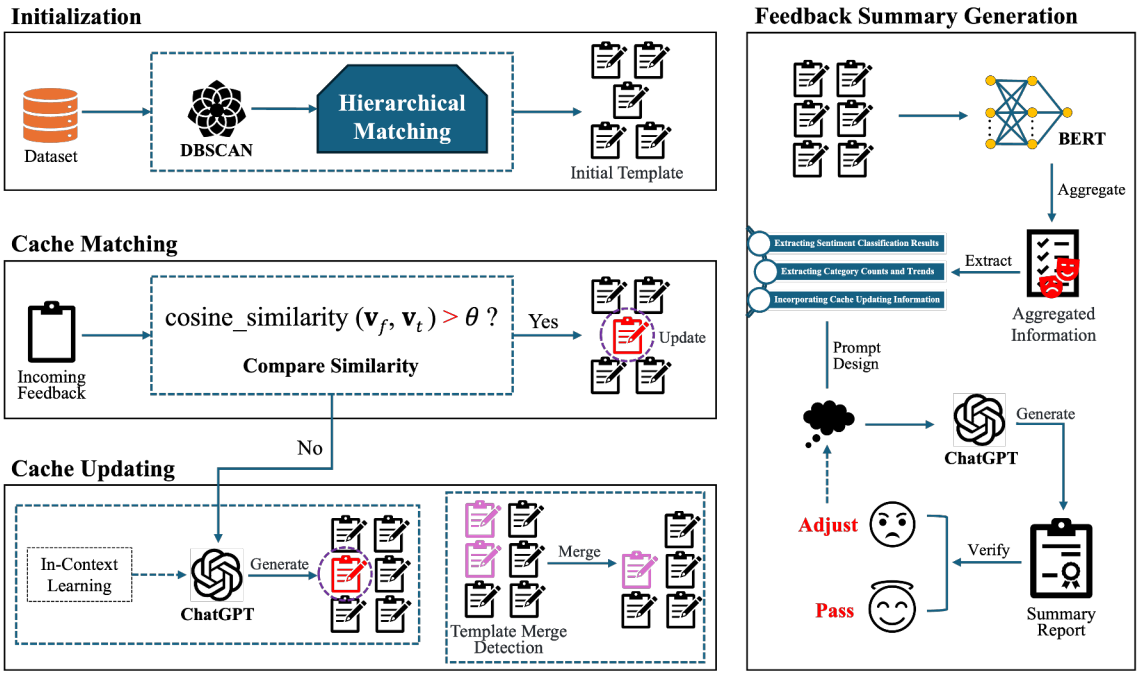

Fig. 1: The overview of our StuLAC framework.

TABLE I: Examples of coarse-level feedback templates with student feedback.

Template	Keywords	Sentiment	Example Feedback
Student Engagement	{"interactive," "participative," "engaging"}	Positive	*The course was very engaging, and the instructor's interactive approach helped me stay focused.*
Course Pacing	{"too fast," "overwhelming," "difficult to keep up"}	Neutral-to-Negative	*I enjoyed the course, but the pace was too fast, making it hard to keep up with the lessons.*
Instructor Clarity	{"clear," "well-explained," "helpful"}	Positive	*The material was quite challenging and sometimes difficult to follow, but the instructor was clear in their explanations.*
Course Material Difficulty	{"difficult," "complex," "hard to follow"}	Neutral-to-Negative	*The course content was a bit overwhelming at times, especially the amount of information we had to process each week.*

To validate the accuracy of the initial clustering and template generation, we conducted a cross-validation process with three Ph.D. students. Each student manually reviewed the initial clusters and templates, verifying the correct assignment of keywords, sentiment markers, and thematic categories. They also provided suggestions for refining templates to better capture nuanced feedback. After incorporating their input, the templates were reassessed for consistency and precision in categorizing new feedback entries. This cross-validation ensured that the templates accurately represented the underlying patterns in the feedback data, significantly improving their quality and enhancing the effectiveness of the initialization phase. As a result, the StuLAC framework is better equipped to handle complex, multi-faceted feedback entries with high precision.

C. Cache Matching

The cache matching phase ensures efficient and accurate matching of new feedback entries to existing templates within the StuLAC framework. Each incoming feedback entry is first converted into a dense vector representation using sentence embeddings, such as those generated by RoBERTa [19]. RoBERTa, a robustly optimized variant of BERT, improves upon the original BERT model by training on larger datasets and using longer sequences, making it well-suited for capturing the contextual nuances of feedback data. These embeddings represent the semantic meaning of the feedback, enabling similarity calculations with stored template embeddings.

To compute the similarity between a feedback entry $\mathbf{f}$ and a template $\mathbf{t}$, we calculate the **cosine similarity** [20], [21] between their respective embeddings, $\mathbf{v}_f$ and $\mathbf{v}_t$. Cosine similarity is given by:

$$\text{cosine\_similarity}(\mathbf{v}_f, \mathbf{v}_t) = \frac{\mathbf{v}_f \cdot \mathbf{v}_t}{\|\mathbf{v}_f\| \|\mathbf{v}_t\|} \tag{2}$$

where $\mathbf{v}_f \cdot \mathbf{v}_t$ is the dot product of the feedback and template embeddings, and $\|\mathbf{v}_f\|$ and $\|\mathbf{v}_t\|$ are the magnitudes of the feedback and template vectors, respectively. This measure produces a similarity score ranging from -1 to 1, where a higher score indicates greater similarity.

The Cache Matching Algorithm is outlined in Algorithm 1, which computes the cosine similarity for each feedback entry against all templates (Line 6), returning the classification result or triggering a cache update if no match is found (Lines 12-16). In detail, each feedback entry is compared against all

existing templates based on a predefined similarity threshold θ (Lines 5-11). If the cosine similarity between the feedback entry $\mathbf{f}$ and an existing template $\mathbf{t}$ exceeds θ, the feedback is classified under the corresponding template (Lines 7-13). For instance, a feedback entry with a high similarity score to the "Instructor Clarity" template would be categorized under that theme. If no match is found, the feedback triggers the cache update phase (Lines 14-15).

Algorithm 1 Cache Matching and Classification Algorithm

1: **Input:** Feedback entry $\mathbf{f}$, set of templates $\mathcal{T} = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n\}$, threshold θ
2: **Output:** Classification of $\mathbf{f}$ or cache update trigger
3: Initialize max_similarity $\leftarrow 0$
4: Initialize matched_template $\leftarrow$ None
5: **for** each template $\mathbf{t}_i \in \mathcal{T}$ **do**
6: Calculate similarity $s_i = \text{cosine\_similarity}(\mathbf{v}_f, \mathbf{v}_{t_i})$
7: **if** $s_i > \theta$ **and** $s_i >$ max_similarity **then**
8: max_similarity $\leftarrow s_i$
9: matched_template $\leftarrow \mathbf{t}_i$
10: **end if**
11: **end for**
12: **if** matched_template is not None **then**
13: **Return** Classification of $\mathbf{f}$ under matched_template
14: **else**
15: Trigger Cache Update for $\mathbf{f}$
16: **end if**

To illustrate, consider the following examples of feedback entries and their cosine similarity scores with two existing templates, "Instructor Clarity" and "Course Material Difficulty", using a threshold $\theta = 0.85$: In the first example of Table II, the feedback entry "*The instructor explained concepts clearly and effectively.*" matches the "Instructor Clarity" template with a similarity score of 0.90, which exceeds the threshold of 0.85, and is classified under this template. In the second example, the feedback entry "*The material was hard to follow, very complex, and overwhelming.*" achieves a similarity score of 0.85 with the "Course Material Difficulty" template and is matched accordingly. The third example, "*The course was engaging, but I struggled to keep up with the pace.*", scores 0.80 for "Instructor Clarity" and 0.75 for "Course Material Difficulty", neither of which meets the threshold, so it triggers a cache update.

Through this cosine similarity-based matching process, the StuLAC framework ensures efficient and accurate classification of feedback. The cache updating mechanism further ensures that the system remains adaptable, continuously refining templates to accommodate new patterns in student feedback.

D. Cache Updating

The cache updating phase is crucial for enabling the StuLAC framework to evolve dynamically in response to new feedback patterns. When an incoming feedback entry does not meet the similarity threshold with existing templates, it triggers this phase. During this phase, we utilize ChatGPT [22], specifically the GPT-4-turbo model via OpenAI's API[1], to generate new templates or refine existing ones. This allows the system to adapt and maintain high accuracy in feedback classification.

Furthermore, In-Context Learning (ICL) plays a critical role in enhancing the adaptability of the StuLAC framework. By providing ChatGPT with contextual examples from the cache, ICL ensures that the generated templates are not only distinct and relevant but also aligned with emerging feedback patterns. This approach is crucial because it allows the model to generate templates that reflect the specific context of the unmatched feedback, avoiding redundancy and ensuring that new templates complement the existing cache. Rather than generating entirely new templates without any reference, ChatGPT is presented with the most similar existing template, which guides it in producing more refined and contextually informed templates. This process improves the specificity of the templates, ensuring they address subtle nuances in the feedback, and enhances their adaptability to a wide range of feedback variations, ultimately improving the classification accuracy.

To implement ICL, we create a prompt that incorporates the most similar templates as context. When a feedback entry is unmatched, the template with the highest similarity (even below the threshold) is included in the prompt. This context helps ChatGPT understand the feedback's nuances and generate a more appropriate template. The prompt structure is as follows:

- **Introduction**: Briefly explain the purpose of generating a new template.
- **ICL Template**: Include the most similar existing template, specifying that the new template should address different aspects.
- **Feedback Entry**: Present the unmatched feedback to guide the generation of a new template.

For example, the following prompt might be used:

> *"Please generate a new template based on the following feedback. The new template should capture aspects that differentiate it from existing templates. The most similar template is: [Instructor Clarity: Keywords - clear, understandable, well-explained]. New feedback: 'The instructor provided detailed explanations, but some sections were hard to follow.' Generate a template that differentiates from the existing ones."*

In addition to generating new templates, we also incorporate a **template merge detection** mechanism to avoid redundancy and improve cache efficiency. This process periodically checks the similarity between existing templates. If two templates have a similarity score above a merge threshold θ_{merge}, they are flagged for merging. For instance, if the templates "Instructor Clarity" (keywords: {clear, well-explained, understandable}) and "Instructor Thoroughness" (keywords: {detailed, thorough, comprehensive}) are highly similar, ChatGPT is prompted to merge them into a single comprehensive template. An example prompt for merging might be:

[1]https://platform.openai.com/docs/introduction

TABLE II: Cosine similarity matching of feedback entries to templates.

Feedback Entry	Instructor Clarity (Similarity)	Course Material Difficulty (Similarity)	Matching Result
The instructor explained concepts clearly and effectively.	0.90	0.35	Instructor Clarity
The material was hard to follow, very complex, and overwhelming.	0.40	0.85	Course Material Difficulty
The course was engaging, but I struggled to keep up with the pace.	0.80	0.75	Cache Update Triggered

"The following two templates have overlapping characteristics. Please merge them into one template that captures both aspects. Template 1: [Instructor Clarity: Keywords - clear, understandable, well-explained]. Template 2: [Instructor Thoroughness: Keywords - detailed, thorough, comprehensive]. Generate a merged template that covers both aspects effectively."

Table III illustrates the cache updating process, showing both template generation and merging. In the first two rows, ChatGPT generates new templates for unmatched feedback, such as "Instructor Thoroughness with Complexity", which distinguishes itself by including the keyword "complex". In the third row, templates for "Instructor Clarity" and "Instructor Thoroughness" are merged into a more comprehensive template "Instructor Clarity and Thoroughness".

This cache updating process, which involves dynamic template generation and merging, allows the StuLAC framework to continuously adapt to new types of feedback while maintaining an efficient and scalable cache structure. ICL and contextually-guided prompt design ensure that the new templates complement the existing cache, reducing redundancy and improving the overall accuracy and adaptability of the system.

E. Feedback Summary Generation

After completing the feedback analysis, the StuLAC framework generates a comprehensive feedback summary report using ChatGPT. This report synthesizes key insights from the feedback, such as classification statistics and sentiment distribution, which are derived from the cache matching and updating phases. This enables efficient generation of a high-level overview, helping stakeholders quickly understand trends in student feedback.

1) Classification Statistics and Sentiment Distribution: The system categorizes feedback entries into predefined themes, such as "Teaching Quality" and "Course Difficulty", and counts the number of entries in each category. Sentiment analysis is then performed to classify the feedback as positive, neutral, or negative. Rather than using a pre-trained model like BERT, our system employs a hybrid language model [23] combining RoBERTa and an Attention-based Bi-LSTM. RoBERTa [19], an advanced variant of BERT [24], generates contextual embeddings that capture the semantic meaning of the feedback, while the Bi-LSTM [25] and Attention mechanisms model long-range dependencies and selectively emphasize relevant sentiment cues. This hybrid approach enhances sentiment classification accuracy by effectively handling the complexity of long and mixed-sentiment feedback, which is common in student responses.

Specifically, each feedback entry is first processed by the RoBERTa model to generate dense vector embeddings that capture the context and key terms. The Bi-LSTM layer then models the sequential dependencies in the feedback, while the Attention mechanism highlights the most important aspects, improving sentiment classification precision. This multi-layered approach enables the model to more accurately classify feedback, even when it contains mixed sentiments or subtle emotional nuances.

For example, a feedback entry like "*The lecture was engaging, though the pace was a bit fast.*" is classified as positive overall, despite the slight critique, due to the model's ability to focus on the predominant sentiment. Similarly, feedback containing multiple sentiments, such as "*The instructor was clear, but some topics were confusing.*" is classified using our hierarchical matching approach, where the positive sentiment (clear instructor) is assigned to the "Instructor Clarity" template, and the negative sentiment (confusing topics) is mapped to the "Course Content" template. The sentiment distribution is then aggregated for each category to give an overview of the overall student sentiment. This information serves as the foundation for generating a summary report that accurately reflects sentiment trends across feedback themes.

2) Prompt Design for ChatGPT Summary Generation: To produce a coherent and informative summary, we design a specific prompt that incorporates key statistical data from the feedback analysis, including sentiment distribution and category counts. Additionally, we include any new templates generated during the cache updating phase to highlight the system's adaptability. The following steps outline how we extract the necessary data and construct the prompt for ChatGPT:

- **Extracting Sentiment Classification Results**: Feedback is classified into positive, neutral, or negative sentiment using a hybrid language model that combines RoBERTa and an Attention-based Bi-LSTM. For each feedback category, we calculate the percentage of entries assigned to each sentiment class. For instance, in the "Teaching Quality" category, out of 150 feedback entries, 82.7% of the feedback might be classified as positive, 6.3% as neutral, and 11.0% as negative. This sentiment distribution is then incorporated into the prompt for generating the feedback summary.
- **Extracting Category Counts and Trends**: We summarize the total number of feedback entries in each category to capture the primary areas of student concern. For example, "Teaching Quality" have 150 entries, representing 30% of all feedback, while "Course Difficulty" have 80 entries, or 16% of the total. Additionally, we track any notable changes in feedback volume, such as a 14.3% increase in "Course Difficulty" feedback, which signals a growing area of concern.
- **Incorporating Cache Updating Information**: When new templates are generated during the cache updating

TABLE III: Examples of cache updating with ChatGPT-generated and merged templates.

Unmatched Feedback / Template Pair	Cache Update Type	New/Refined Template
The instructor was thorough, but I struggled with some complex sections.	New Template Generation	Instructor Thoroughness with Complexity (Keywords: detailed, thorough, complex)
The course content is relevant but lacks depth in some areas, especially in the final project.	New Template Generation	Material Depth and Project Relevance (Keywords: lacks depth, superficial, needs expansion)
{Instructor Clarity, Instructor Thoroughness}	Template Merging	Instructor Clarity and Thoroughness (Keywords: clear, detailed, understandable, comprehensive)

phase, we include information on the newly created templates, such as "Online Interaction" or "Classroom Engagement," to reflect emerging feedback themes.

The prompt structure used to guide ChatGPT is as follows:

> *"Generate a summary report based on the following feedback analysis results:*
> *1. Teaching Quality: 150 feedback entries (30% of total), 82.7% positive, 6.3% neutral, 11.0% negative. Key terms: clear, well-organized, engaging.*
> *2. Course Difficulty: 80 entries (16%), 68.2% positive, 13.6% neutral, 18.2% negative. Key terms: difficult, complex, more support needed.*
> *3. New Templates: 5 new templates created, covering emerging themes such as online interaction, classroom engagement, assessment feedback, teaching methods, and course structure.*
> *4. Trends: Compared to the previous quarter, feedback on course difficulty increased by 14.3%.*
> *Please summarize these findings, emphasizing the main concerns of students and overall sentiment trends."*

This structure ensures that ChatGPT is provided with all relevant statistical information needed to create a meaningful, data-driven summary. Table IV presents an example of the feedback summary generated by ChatGPT, based on the above prompt. The report consolidates key feedback statistics into an easy-to-read format for stakeholders.

Once ChatGPT generates the summary, it is reviewed for accuracy and coherence by our experts. If any critical elements are missing or if the phrasing is unclear, the prompt is refined to ensure the summary accurately reflects the findings and provides actionable insights. This structured approach enables ChatGPT to generate detailed feedback summaries that effectively communicate key insights and trends to stakeholders, thereby facilitating data-driven decision-making in educational contexts.

III. Empirical Setup

In this section, we describe the empirical setup used to evaluate the performance of the StuLAC framework. We validate our model using a proprietary dataset collected from our institution, which spans several years of student feedback data. The evaluation focuses on two main aspects: (1) the overall performance of the framework in terms of accuracy and efficiency, and (2) the effectiveness of its components, such as the accuracy of sentiment classification. The results from this evaluation will demonstrate how the StuLAC framework compares to traditional methods, highlighting its strengths in processing large volumes of diverse and dynamic student feedback data.

A. Dataset

Our dataset is sourced from the student information management system of a large higher education institution, covering a wide range of feedback entries collected from 2018 to 2024. The dataset consists of over 80,000 individual feedback entries, providing a comprehensive representation of student perspectives over several years. These feedback entries were collected from various sources, including student interactions, Q&A sessions, and formal course evaluations. This diversity in data sources allows for a holistic view of student experiences across different aspects of their educational journey, from teaching quality to course content and learning environments.

Each feedback entry is rich in thematic variety and sentiment, with students offering insights on aspects such as course structure, teaching effectiveness, material difficulty, and overall satisfaction. The dataset includes feedback entries of varying lengths, ranging from short comments to long, detailed responses. These varying feedback lengths ensure that our model can be tested on both simple and complex sentences, enabling the evaluation of how well the system handles different types of feedback. Furthermore, the diversity of sentiments (positive, neutral, negative) expressed across these entries helps us assess the accuracy and adaptability of the StuLAC framework in classifying and categorizing feedback. Table V presents a sample of feedback entries from the dataset, showcasing a mix of short and long comments. This dataset serves as a robust foundation for analyzing how the StuLAC framework processes and synthesizes feedback, providing valuable insights for improving teaching and learning experiences at the institution.

B. Research Questions

This paper mainly focuses on the following three research questions and designs our experiments accordingly.

RQ1: How does the StuLAC framework perform in generating feedback summaries?

To thoroughly assess the performance of the StuLAC framework, we conducted evaluations focusing on the quality and effectiveness of its Feedback Summary Generation component. The accuracy and comprehensiveness of these summaries were considered key indicators of the framework's overall performance in processing and synthesizing student feedback. The evaluation specifically examined the quality of feedback summaries across four criteria: **Accuracy**, **Completeness**, **Readability**, and **Language Fluency**.

In detail, Accuracy assessed the extent to which the summaries correctly captured the key points and sentiment trends present in the original feedback data. Completeness evaluated whether the summaries included all major themes and

TABLE IV: The example of ChatGPT-generated feedback summary report.

Category	Summary Report
Teaching Quality *(Course: Intro to Data Science)*	The "Introduction to Data Science" course received predominantly positive feedback regarding teaching quality, with 150 entries, representing 30% of total feedback. A significant 82.7% of students praised the clarity and structure of lectures, noting terms such as "clear", "well-organized", and "engaging". However, 6.3% provided neutral feedback, indicating potential areas for further refinement, and another 11% raised issues related to pacing. Suggested improvements include incorporating more visual aids and practical examples to sustain engagement.
Course Difficulty *(Course: Advanced Machine Learning)*	Students expressed challenges with the "Advanced Machine Learning" course, with 80 feedback entries (16% of total feedback) highlighting the complexity of content. Of these, 68.2% provided positive feedback, recognizing the value of challenging material. However, 13.6% were neutral, and another 18.2% expressed difficulty, particularly in understanding advanced concepts and implementing algorithms. Key terms such as "difficult", "complex", and "steep learning curve" were frequently mentioned. To address these concerns, it is recommended to introduce supplementary tutorials, offer additional office hours, and provide more worked examples in the lectures. This category also saw a 14.3% increase in feedback entries from the previous quarter, suggesting a growing need for enhanced support.
Assessment and Feedback *(Course: Intro to Programming)*	In the "Introduction to Programming" course, 100 feedback entries (20% of total feedback) were related to assessment and feedback. While 79.5% of students appreciated the regular assessments and detailed feedback, 8.6% provided neutral feedback, and 11.9% mentioned delays in receiving grades and comments. Key terms included "helpful feedback", "consistent assessments", and "grade delays". To improve the experience, implementing automated grading for certain assignments and providing more timely feedback would be beneficial.
New Template Generation	During this analysis, students highlighted new themes in their feedback, prompting the creation of 5 additional templates to address specific areas of interest such as "Online Interaction" and "Classroom Engagement". Many students expressed a desire for more interactive elements in online courses, emphasizing the importance of real-time engagement and collaborative activities. Additionally, feedback on classroom participation reflected a need for improved in-class dynamics and more opportunities for active involvement. These new templates will help ensure that future feedback on these aspects is accurately captured and categorized, allowing for more targeted improvements in response to student needs.

TABLE V: Examples of student feedback entries.

Feedback Type	Example Feedback Entry
Short Feedback	*The course material was relevant, but the instructor could have explained some concepts more clearly.* *The teaching was clear and engaging, but the assignments were too difficult and time-consuming.*
Long Feedback	*I appreciate the effort the instructor put into the course, but I found some of the topics to be overly complex. The lectures moved too quickly, and I often had trouble following along, especially when we moved on to more advanced topics without fully covering the basics. I think that the pace needs to be slowed down, and perhaps we could have more examples during lectures to better understand the concepts. Overall, I think the course content is valuable, but I would like to see improvements in the pacing and more opportunities for interaction in class.* *While the course materials were comprehensive and the instructor was clear, I found the difficulty level to be overwhelming. I appreciate the depth of the subject, but the assignments felt like they were designed for students with prior experience. More support, such as additional tutorials or office hours, would be helpful. Moreover, I think the course could benefit from more practical applications or real-world examples to better connect theoretical knowledge to real-life scenarios. Overall, the course could be improved by balancing the difficulty with more accessible resources.*

sentiments without omitting critical information. Readability focused on how well the summaries could be understood, emphasizing logical flow, sentence structure, and the organization of information, ensuring clarity for a general audience. Language Fluency, on the other hand, examined the grammatical correctness, stylistic appropriateness, and overall linguistic polish of the summaries, assessing whether they adhered to high standards of professional writing. While both criteria address linguistic aspects, readability emphasizes accessibility and clarity of content, whereas language fluency focuses on adherence to formal writing conventions and error-free composition. This distinction is particularly important as feedback summaries are a critical tool in teaching evaluation, directly informing administrative and instructional decision-making. Summaries that are linguistically polished not only enhance credibility but also ensure that the insights are effectively communicated to stakeholders such as faculty, administrators, and policy-makers. These criteria provided a comprehensive framework for assessing the quality of the summaries generated by the StuLAC framework.

We enlisted 10 human evaluators with expertise in educational assessment and sentiment analysis to rate each criterion on a scale from 1 to 5, where higher scores indicate better performance. Each evaluator independently reviewed summaries generated for 100 randomly selected feedback samples, providing scores for each of the four criteria. The overall performance of the Feedback Summary Generation component was determined by calculating two metrics: (1) the mean score for each criterion across all evaluators, and (2) the total average score across all evaluators and criteria. These metrics provided a comprehensive assessment of the framework's ability to generate high-quality, informative, and linguistically coherent summaries.

RQ2: How effective is the hybrid language model in StuLAC for sentiment and category distribution accuracy?

The hybrid language model employed in StuLAC, combining RoBERTa with an Attention-based Bi-LSTM, plays a pivotal role in accurately categorizing feedback and detecting sentiment. To assess the performance of the StuLAC framework in generating sentiment and category distributions, we focused on five key feedback categories: **Teaching Quality**, **Course Difficulty**, **Assessment Feedback**, **Classroom Engagement**, and **Resource Availability**. These categories were chosen because they are commonly highlighted in student feedback and encompass a diverse range of themes, allowing us to test the model's robustness in both sentiment detection and categorization.

For each category, we analyzed the sentiment distribution—positive, neutral, and negative—by comparing the re-

sults from the StuLAC-generated summaries with manually compiled baseline statistics. This comparison allowed us to assess the accuracy and reliability of StuLAC in capturing sentiment trends. Furthermore, we calculated key performance metrics such as accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$, precision = $\frac{TP}{TP+FP}$, recall = $\frac{TP}{TP+FN}$, and F1-score = $2 \times \frac{\text{precision} \times \text{recall}}{\text{precision}+\text{recall}}$ for each category, providing a comprehensive evaluation of the framework's effectiveness, where TP is true positives, TN is true negatives, FP is false positives, and FN is false negatives. These metrics were critical for understanding the model's ability to correctly classify feedback, minimize false positives and negatives, and ensure comprehensive sentiment representation across categories.

RQ3: Does the Feedback Summary Generation component offer advantages in terms of efficiency and generation time?

To evaluate the efficiency and scalability of the Feedback Summary Generation component, we tested the generation time across varying dataset sizes of 10,000, 20,000, and 50,000 entries. During the **Initialization** phase, we used 10% of the dataset to construct the initial template library, enabling the system to establish a foundational cache for more efficient matching and updates in the subsequent phases.

IV. Results and Analysis

A. Effectiveness of the StuLAC framework (RQ1)

To thoroughly evaluate the performance of the StuLAC framework, we enlisted 10 human evaluators with expertise in educational assessment and sentiment analysis. The evaluators rated the summaries generated by StuLAC across four criteria—accuracy, completeness, readability, and language fluency—on a scale from 1 to 5, with higher scores indicating better performance. Each evaluator reviewed summaries generated for 100 randomly selected feedback samples. The overall performance was calculated as the mean score for each criterion across all evaluators, along with the total average score.

Additionally, to provide a benchmark, the evaluators also assessed manually generated feedback reports created in recent years using the same four criteria. This comparative analysis allowed us to highlight the strengths and limitations of the StuLAC framework relative to manual processes.

TABLE VI: Summary quality scores for StuLAC and manually generated reports (scale: 1–5).

Eval.	Accuracy		Completeness		Readability		Language Fluency		Avg. Score
	StuLAC	Manual	StuLAC	Manual	StuLAC	Manual	StuLAC	Manual	StuLAC / Manual
1	4.4	3.9	4.1	3.8	4.6	4.2	4.7	4.3	4.45 / 4.05
2	4.7	4.1	4.5	4.0	4.8	4.3	4.6	4.2	4.65 / 4.15
3	4.3	3.8	4.2	3.7	4.7	4.1	4.9	4.2	4.53 / 3.95
4	4.6	4.0	4.3	3.9	4.8	4.2	4.8	4.3	4.63 / 4.10
5	4.2	3.7	4.0	3.6	4.6	4.0	4.5	4.1	4.33 / 3.85
6	4.5	4.0	4.4	3.8	4.7	4.2	4.6	4.3	4.55 / 4.08
7	4.6	4.1	4.2	3.8	4.8	4.3	4.7	4.4	4.58 / 4.15
8	4.4	3.9	4.3	3.7	4.7	4.1	4.7	4.2	4.53 / 3.98
9	4.7	4.2	4.5	4.1	4.9	4.4	4.8	4.5	4.73 / 4.30
10	4.5	4.0	4.3	3.9	4.6	4.2	4.7	4.3	4.53 / 4.10
Total Avg.	**4.49**	**3.97**	**4.28**	**3.83**	**4.72**	**4.19**	**4.70**	**4.28**	**4.55 / 4.07**

As shown in Table VI, the total average score for summaries generated by StuLAC (4.55) was significantly higher than that of manually generated reports (4.07), representing a 10.5% improvement in overall quality. StuLAC consistently outperformed manual reports across all criteria. Specifically, accuracy was rated 4.49 for StuLAC compared to 3.97 for manual reports, reflecting an 11.6% improvement in capturing key points and sentiment trends. Similarly, completeness, readability, and language fluency saw improvements of 10.5%, 11.2%, and 8.9%, respectively. Evaluators noted that the automated summaries provided a more comprehensive coverage of themes and subtopics, while manual reports often lacked nuanced details due to time constraints.

In detail, the evaluators identified several key factors contributing to the improvements observed in the StuLAC-generated summaries. First, manual reports require significant expertise and time to produce, making it difficult to ensure consistency and detail, especially under tight deadlines. In contrast, StuLAC utilizes a hybrid language model and a template-based structure, which enhances the comprehensive coverage and thematic accuracy of the summaries. This automated approach not only increases the breadth of coverage but also enhances precision, reducing the likelihood of overlooking key themes—something that is often problematic in manual summarization. Furthermore, the automation of sentiment classification and feedback categorization through StuLAC helps minimize human error, particularly in identifying subtle nuances or mixed sentiments within the feedback.

Qualitative feedback from two evaluators revealed specific strengths and areas for further improvement. Evaluator 3 highlighted the model's ability to synthesize complex feedback into concise, actionable insights, particularly in areas such as course pacing. They observed, "*The summaries of feedback on course pacing included detailed references to specific challenges, which are often overlooked in manual reports.*" However, they also noted that the completeness of the summaries could be enhanced by providing additional contextual examples, especially in cases of nuanced feedback that require further clarification. Evaluator 9, while praising the fluency and coherence of the summaries, pointed out that some of them tended to generalize themes without sufficiently delving into specific student concerns. For example, while the summaries on teaching quality were generally positive and comprehensive, they could be further enriched by addressing subtopics such as specific teaching techniques. This highlights a continuing need for the model to improve the completeness of its summaries by incorporating more contextual examples and examining particular aspects of feedback in greater detail. Future iterations should prioritize enhancing both completeness and specificity to more accurately reflect the diversity and nuance of student feedback.

Answer to RQ1: StuLAC significantly outperformed manual reports across all evaluation criteria, achieving a higher overall quality score through more accurate, complete, and fluent summaries; however, evaluators also recommended improving completeness by including more contextual examples and addressing specific student concerns.

B. Effectiveness of the Hybrid Language Model in StuLAC (RQ2)

The performance of the hybrid language model within the StuLAC framework in generating sentiment and category distributions was assessed across five key feedback categories, with accuracy, precision, recall, and F1-score metrics used to evaluate its effectiveness in feedback categorization and sentiment detection. As shown in Table VII, the overall accuracy across all categories was 86.4%, with the highest performance observed in the Teaching Quality category, which achieved an accuracy of 91%. In comparison, misclassifying positive and neutral feedback as negative had a notable impact on the quality of the generated summaries. Specifically, such misclassifications can lead to a skewed representation of feedback, affecting the accuracy of sentiment trends and potentially overlooking key areas of concern. Despite these challenges, the hybrid language model demonstrated impressive performance, with only 6 out of 73 negative feedback samples being incorrectly classified, yielding a classification accuracy of 91.8%. This result outperforms several baseline models presented in the literature [26], further demonstrating the model's robust ability to capture and accurately classify sentiment.

However, slightly lower accuracy was observed in the Resource Availability and Assessment Feedback categories, with accuracy scores of 82% and 84%, respectively. These categories presented particular challenges, often involving feedback samples with informal language or ambiguous sentiment, which posed difficulties for precise sentiment classification. For example, in the Resource Availability category, some feedback entries were vague or casually phrased, such as "*The resources are okay, but I think we could use more materials.*" While this comment has a generally positive tone, the vague phrasing caused some challenges in categorizing the feedback as either neutral or positive, resulting in a slight decrease in precision and recall.

Similarly, in the Assessment Feedback category, some feedback entries had mixed sentiments or subtle language that made them difficult to classify accurately. For instance, "*The feedback on assignments is helpful, but I wish it was more timely.*" contains both positive sentiment about the feedback's usefulness and negative sentiment about its timeliness. Although the coarse and fine classification process effectively divided these sentiments into two parts, the subtlety of the wording occasionally caused misclassifications. This resulted in some negative sentiments being underrepresented, leading to a small decrease in accuracy compared to the baseline.

These challenges illustrate that student feedback, unlike more structured critiques such as movie reviews, often contains informal, ambiguous, or indirect language that can be difficult for models to categorize accurately. Feedback is more conversational, and often lacks the explicitness found in professional reviews, making it harder to classify sentiments with high precision. This hybrid approach enables the model to more accurately detect nuanced sentiments, improving the overall sentiment classification accuracy and ensuring a higher-quality representation of feedback. As a result, the StuLAC framework's performance is significantly enhanced in handling informal and complex feedback, leading to more accurate and reliable sentiment and category classification.

Answer to RQ2: The hybrid language model in StuLAC outperformed baseline models, particularly in detecting nuanced feedback; however, it faced challenges in accurately classifying informal or ambiguous sentiments, especially within the Resource Availability and Assessment Feedback categories.

C. Evaluation of Efficiency and Generation Time (RQ3)

The average generation time for processing each dataset size was recorded, as shown in Table VIII. In addition to generation time, we also evaluated the number of templates generated and the processing time per entry for each dataset size.

The results reveal a significant, non-linear increase in generation time as the dataset size grows. For instance, processing 10,000 entries took 576 seconds, while processing 20,000 entries required 1172 seconds—2.03 times longer than the initial dataset. For the largest dataset of 50,000 entries, the generation time increased to 3590 seconds, which is 6.23 times longer than the time required for the initial dataset, demonstrating the increasing complexity as the dataset size expands.

However, the number of templates generated follows an opposite trend. For 10,000 entries, 36 templates were generated, whereas for the largest dataset of 50,000 entries, only 77 templates were created. This difference can be attributed to the efficiency of the cache matching phase. As the dataset grows, the initial templates already capture the majority of feedback patterns, reducing the need for new templates. The system's ability to effectively reuse existing templates leads to fewer new templates being generated, even as the data volume increases. This highlights the efficiency of the StuLAC framework in reusing templates and adapting to new feedback patterns without significant increases in template generation.

Answer to RQ3: StuLAC demonstrates strong efficiency in feedback summarization, with generation time increasing non-linearly with dataset size, while the number of new templates grows modestly due to effective template reuse—highlighting the system's scalability and adaptability to larger datasets.

V. Conclusion

This paper introduced StuLAC, a Software Engineering-driven framework for scalable student feedback analysis using LLMs with Adaptive Template-Based Caching. Empirical results on 80,000 student feedback entries demonstrate that StuLAC-generated summaries improve overall quality by 10.5% compared to manually generated reports, while also achieving faster processing times. Additionally, StuLAC attains an 86.4% accuracy and an 86.24% F1-score in sentiment detection. These results establish StuLAC as a robust, efficient, and adaptive solution for educational feedback systems.

TABLE VII: Sentiment and category distribution accuracy by template category.

Category	Positive	Neutral	Negative	Accuracy	Precision	Recall	F1-score
Teaching Quality	78 (Generated)	12 (Generated)	10 (Generated)	0.91	0.92	0.89	0.905
	80 (Baseline)	10 (Baseline)	10 (Baseline)	-	-	-	
Course Difficulty	38 (Generated)	30 (Generated)	32 (Generated)	0.88	0.86	0.88	0.869
	40 (Baseline)	30 (Baseline)	30 (Baseline)	-	-	-	
Assessment Feedback	70 (Generated)	23 (Generated)	7 (Generated)	0.84	0.85	0.83	0.839
	72 (Baseline)	20 (Baseline)	8 (Baseline)	-	-	-	
Classroom Engagement	65 (Generated)	22 (Generated)	13 (Generated)	0.87	0.89	0.85	0.870
	68 (Baseline)	20 (Baseline)	12 (Baseline)	-	-	-	
Resource Availability	60 (Generated)	25 (Generated)	15 (Generated)	0.82	0.84	0.81	0.825
	62 (Baseline)	25 (Baseline)	13 (Baseline)	-	-	-	
Overall	**-**	**-**	**-**	**86.4%**	**87.2%**	**85.3%**	**86.24%**

TABLE VIII: Feedback summary generation time and performance metrics.

Dataset Size (entries)	Average Generation Time (seconds)	Templates Generated	Processing Time per Entry (seconds)
10,000	576	36	0.0640
20,000	1172	49	0.0651
50,000	3590	77	0.0798

Future work will focus on integrating real-time feedback visualization dashboards, enhancing multilingual feedback support, and deploying StuLAC in large-scale academic institutions. By bridging Software Engineering and AI-driven NLP, StuLAC contributes to the evolution of scalable, automated educational assessment methodologies.

ACKNOWLEDGMENT

This work is partially supported by the General Research Fund of the Research Grants Council of Hong Kong and the research funds from the City University of Hong Kong (6000796, 9229109, 9229098, 9220103, 9229029).

REFERENCES

[1] S. Yu, A. Androsov, H. Yan, and Y. Chen, "Bridging computer and education sciences: a systematic review of automated emotion recognition in online learning environments," *Computers & Education*, p. 105111, 2024.

[2] C. Fissore, F. Floris, M. Marchisio, and S. Rabellino, "Learning analytics to monitor and predict student learning processes in problem solving activities during an online training," in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 2023, pp. 481–489.

[3] W. Ren, R. Wang, S. N. Azlan, and C. Mao, "Factors influencing students' learning satisfaction and students' learning outcomes in blended learning," *International Journal of Education and Practice*, vol. 12, no. 1, pp. 95–108, 2024.

[4] R. R. Siggard and L. Lundgren, "Leveraging epistemic network analysis to understand peer feedback in online courses," *Journal of Science Education and Technology*, pp. 1–12, 2024.

[5] J. T. Richardson, "Instruments for obtaining student feedback: A review of the literature," *Assessment & evaluation in higher education*, vol. 30, no. 4, pp. 387–415, 2005.

[6] B. Wisniewski, K. Zierer, and J. Hattie, "The power of feedback revisited: A meta-analysis of educational feedback research," *Frontiers in psychology*, vol. 10, p. 487662, 2020.

[7] T. Shaik, X. Tao, Y. Li, C. Dann, J. McDonald, P. Redmond, and L. Galligan, "A review of the trends and challenges in adopting natural language processing methods for education feedback analysis," *Ieee Access*, vol. 10, pp. 56 720–56 739, 2022.

[8] I. Benedetto, L. Canale, L. Farinetti, L. Cagliero, and M. La Quatra, "Leveraging summarization techniques in educational technology systems," in *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 2022, pp. 415–416.

[9] J. An, W. Ding, and C. Lin, "Chatgpt," *tackle the growing carbon footprint of generative AI*, vol. 615, p. 586, 2023.

[10] W. Strielkowski, V. Grebennikova, A. Lisovskiy, G. Rakhimova, and T. Vasileva, "Ai-driven adaptive learning for sustainable educational transformation," *Sustainable Development*, 2024.

[11] T. Rasul, S. Nair, D. Kalendra, M. Robin, F. de Oliveira Santini, W. J. Ladeira, M. Sun, I. Day, R. A. Rather, and L. Heathcote, "The role of chatgpt in higher education: Benefits, challenges, and future research directions," *Journal of Applied Learning and Teaching*, vol. 6, no. 1, pp. 41–56, 2023.

[12] D. Baidoo-Anu and L. O. Ansah, "Education in the era of generative artificial intelligence (ai): Understanding the potential benefits of chatgpt in promoting teaching and learning," *Journal of AI*, vol. 7, no. 1, pp. 52–62, 2023.

[13] I. Adeshola and A. P. Adepoju, "The opportunities and challenges of chatgpt in education," *Interactive Learning Environments*, pp. 1–14, 2023.

[14] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier *et al.*, "Chatgpt for good? on opportunities and challenges of large language models for education," *Learning and individual differences*, vol. 103, p. 102274, 2023.

[15] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.

[16] M. Hahsler, M. Piekenbrock, and D. Doran, "dbscan: Fast density-based clustering with r," *Journal of Statistical Software*, vol. 91, pp. 1–30, 2019.

[17] Q. Peng, W. Wang, H. Liu, Y. Wang, H. Xu, and M. Shao, "Towards comprehensive expert finding with a hierarchical matching network," *Knowledge-Based Systems*, vol. 257, p. 109933, 2022.

[18] X. Chen, B. Zhang, C. Zhao, J. Ding, and W. Wang, "From coarse to fine: Hierarchical zero-shot fault diagnosis with multi-grained attributes," *IEEE Transactions on Fuzzy Systems*, 2024.

[19] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[20] P. Xia, L. Zhang, and F. Li, "Learning similarity with cosine similarity ensemble," *Information sciences*, vol. 307, pp. 39–52, 2015.

[21] F. Rahutomo, T. Kitasuka, M. Aritsugi *et al.*, "Semantic cosine similarity," in *The 7th international student conference on advanced science and technology ICAST*, vol. 4, no. 1. University of Seoul South Korea, 2012, p. 1.

[22] C. K. Lo, K. F. Hew, and M. S.-y. Jong, "The influence of chatgpt on student engagement: A systematic review and future research agenda," *Computers & Education*, p. 105100, 2024.

[23] Y. Sun, J. W. Keung, Z. Yang, S. Liu, and H. K. Yu, "Semirald: A semi-supervised hybrid language model for robust anomalous log detection," *Information and Software Technology*, p. 107743, 2025.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[25] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[26] Y. SUN, J. Keung, Z. Yang, H. K. Yu, W. Luo, and Y. Liao, "Adaptive domain-specific document-level sentiment analysis with meta-learning and hybrid language models," *Available at SSRN 5044241*.