Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
ASEE
Paper ID #49201

Explore the Possibility of Monitoring Project Member Interactions Using Natural Language Processing

Kaiwen Guo, New York University Tandon School of Engineering

Computer Science Senior at New York University

Malani Snowden, New York University Tandon School of Engineering
Prof. Rui Li, New York University

Dr. Li earned his master's degree in Chemical Engineering in 2009 from the Imperial College of London and his doctoral degree in 2020 from the University of Georgia, College of Engineering.

Explore the Possibility of Monitoring Project Member Interactions Using Natural Language Processing

Abstract

This study looks into the use of team evaluation software, incorporating peer ratings, peer comments, and machine-learning-based analysis, to assess the project performance of student project teams. Teamwork is an essential competency for students. The early development of collaborative skills is critical for academic success and future career success. Previous studies have suggested that the data-driven team evaluation could help with team performance evaluation. However, most of the team-based software will provide peer rating without detailed feedback of student team performance. CATME (Comprehensive Assessment of Team Member Effectiveness) greatly facilitates peer assessments by allowing students to rate and comment on each other's contributions, fostering accountability and constructive feedback. Additionally, machine learning algorithms analyze the collected data to identify patterns in team dynamics on the five dimensions of CATME, individual participation, and team synergy.

Natural Language Processing (NLP) tools play a crucial role in evaluating team performance via analyzing communication, feedback, and interactions among team members. This study will further explore a variety of natural language processing tools, such as sentimental analysis, text classification, topic modeling, named entity recognition (identify potential project leaders), and keyword extraction. By considering both human and machine evaluations, the study aims to provide a comprehensive assessment of team effectiveness, highlighting areas for growth for individual students. The findings suggest that this approach not only increases the efficiency of the evaluation process, but also possibly improves student engagement, and the overall quality of teamwork amongst student groups.

Introduction

A language can be defined as a system of rules or symbols that combine to express or broadcast information, ultimately shaping how we perceive and communicate within different cultural contexts [1]. Because not all users are familiar with machine-specific languages, Natural Language Processing (NLP) has emerged as a subfield of Artificial Intelligence (AI) that enables computers to understand statements or words written in human languages [2]. Foundational theories by scholars such as Schank (on conceptual dependencies) and Chomsky (on syntax) paved the way for modern NLP, highlighting the complexities of semantics, morphology, and pragmatics [3][4]. More recently, advancements in NLP toolkits and libraries—such as TextBlob—have made sentiment analysis and text classification accessible, thereby enabling more nuanced, context-sensitive applications [5][6][7].

In tandem with these technological advances, large language models (LLMs) and prompt-engineering strategies have become increasingly prevalent, revealing new possibilities and challenges in text generation, reasoning, and named entity recognition [8][9][10][13]. For instance, NER can parse open-ended feedback to identify team leaders or pinpoint students in need of additional support. Meanwhile, robust prompt design allows instructors or researchers to tailor LLMs for specific instructional goals, though the field continues to refine best practices in prompt-engineering [9].

Within higher education, peer evaluation and feedback play critical roles in developing students' teamwork abilities and self-reflection skills. Tools such as CATME (Comprehensive Assessment of Team Member Effectiveness) facilitate structured peer rating and feedback, ensuring that each team member's contributions are accounted for [11][12]. However, while numeric ratings give broad insight into performance, the sheer volume of qualitative comments can overwhelm instructors in large classes. Research indicates that sentiment analysis and topic modeling—when combined with domain knowledge—can help aggregate and interpret open-ended feedback, boosting instructional efficiency and the depth of insights gleaned [13].

Studies focusing on NLP for educational feedback have notably expanded, exploring how machine-learning-driven text analysis can improve formative assessments, student conceptual understanding, and real-time monitoring of classroom discourse [14][15][16]. For example, Shaik et al. [14] emphasize the trends and challenges in adopting NLP to handle large-scale student evaluations, whereas Lukwaro et al. [16] discuss the obstacles associated with data privacy and context-specific language. Emerging work by Bauer et al. [17] proposes a cross-disciplinary framework for harnessing NLP to support peer feedback, illustrating the potential of AI-driven approaches to enhance collaboration. Similarly, Alhawiti et al. [18] demonstrate that NLP-based systems can help instructors tailor interventions to student needs more effectively than traditional manual review processes.

Despite these advancements, team-based learning contexts—such as large engineering design courses—still generate massive amounts of free-text feedback that are time-consuming to summarize manually. Students are often required to assess peers on various competencies, from technical expertise to communication skills, but instructors face challenges in synthesizing these evaluations for timely intervention [14][17]. Consequently, there is a growing need for a systematic, NLP-enhanced pipeline that can not only capture sentiments and topics in peer feedback but also align with institutional requirements for privacy and anonymization [5][9].

In this paper, we propose a comprehensive NLP-based approach to streamline team evaluation by combining sentiment analysis, topic modeling, and named entity recognition in a unified pipeline. Our system processes and anonymizes student feedback offline, thereby reducing the risk of sensitive data exposure. We build on CATME's existing structure for numeric ratings and add layers of textual analysis to prioritize students who may need earlier support. By fusing quantitative and qualitative insights, we aim to develop a robust framework that addresses the challenges of large-scale peer feedback in higher education.

The subsequent sections detail our methodology—including data preprocessing, model setup, and prompt design—followed by an evaluation of how effectively this framework identifies at-risk students, highlights high performers, and supports timely instructor intervention. We close with discussion and conclusion sections that clarify limitations, potential refinements (e.g., fine-tuning LLMs for the educational domain), and the broader implications of NLP-enabled team evaluation for student success.

Methods

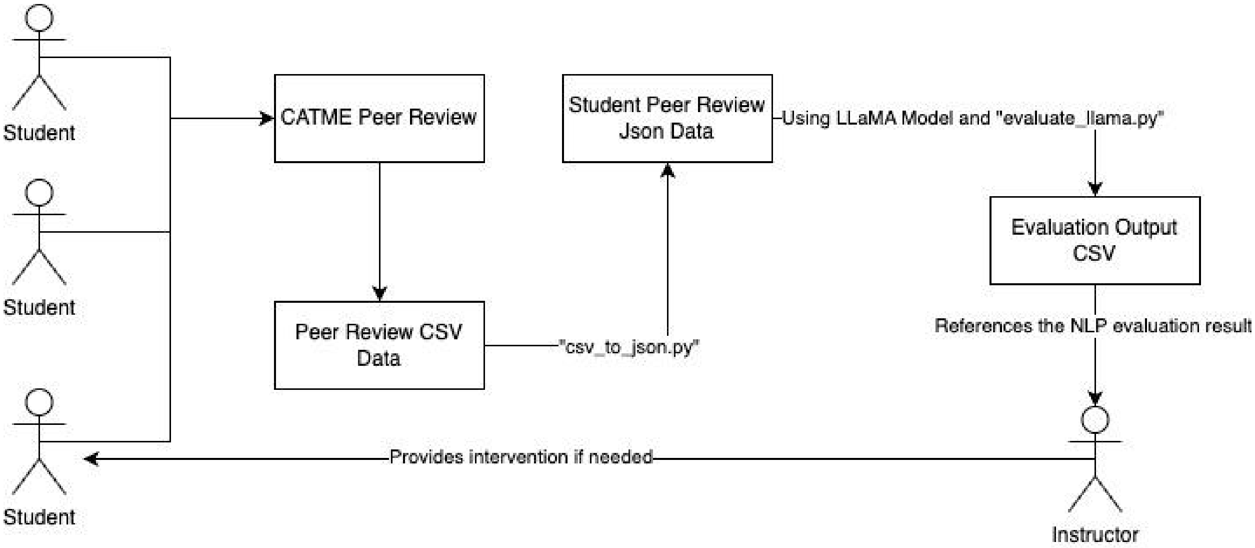
Student
Student
Student
CATME Peer Review
Peer Review CSV Data
"csv_to_json.py"
Student Peer Review Json Data
Using LLaMA Model and "evaluate_llama.py"
Evaluation Output CSV
References the NLP evaluation result
Instructor
Provides intervention if needed

Figure 1: Pipeline Workflow

Workflow explains the entire process. The instructor would need feedback three days after the students submitted their CATME Evaluation.

Participants and Context

This study was conducted in the fall semester at a large private university at Northeast region, focusing on 100 undergraduate students enrolled in the engineering design course EG1003 taught by Dr. Rui Li. The students were organized into 25 project teams, each comprising four members. Over the course of the semester, students completed multiple peer-evaluation surveys using the CATME (Comprehensive Assessment of Team Member Effectiveness) platform. For this research, four consecutive peer-evaluation sessions were examined to assess both individual and team performance.

Data Collection

Data collection centered on exports from the CATME platform, which generates peer-evaluation reports in CSV format after each round of evaluations. Each CSV file contained:

1. **General Activity Information** (e.g., term, instructor, course code).
2. **CATME Rating Data** – including numeric scores for various teamwork dimensions (e.g., Contributing to Team's Work, Interacting with Teammates, etc.).
3. **Professor Comments** – written comments from each student to the professor.
4. **Peer Comments** – free-text evaluations from each student about their teammates.

Together, these four CSV exports contained comprehensive feedback on every student, including numerical peer ratings and open-ended comments.

Data Preprocessing

Because the CATME export format can be complex and inconsistent (e.g., rows with differing column structures, embedded quotes, and irregular spacing), a custom Python script was developed to parse and reorganize the data into a structured JSON file. The script systematically identified sections (e.g., "activity_info," "rater_data," "professor_comments_data," "peer_comments_data," etc.) so that each piece of feedback could be attributed to the correct student. Below is a summary of the key preprocessing steps:

1. **CSV-to-JSON Conversion**
   We used a custom function, *convert_complex_csv_to_json*, to read line by line, detect known section boundaries (e.g., lines beginning with "Q1" or "Student","Team","Comment"), and append them to the appropriate JSON fields. This step facilitated consistent downstream processing.

2. **Student-by-Student Structuring**
   After generating a single JSON file (e.g., peer_eval.json), a second script reorganized comments by student. Specifically, each student record included:
   - **Team ID** – mapped from CSV columns that specify which team the student belonged to.
   - **Comments to the Professor** – aggregated from the "professor_comments_data" section.
   - **Inbound Peer Comments** – all feedback about the student from others.
   - **Outbound Peer Comments** – all feedback the student wrote about their teammates.
3. **Data Cleaning & Anonymization**
   Any personally identifiable information (beyond name and team ID) was either removed or replaced with a placeholder (e.g., "N/A") for student privacy, in accordance with university guidelines and IRB best practices.

Model Setup and Comparison

Our pipeline begins by converting raw CATME CSV files into a structured JSON format through the custom script csv_to_json.py. This script handles the irregularities of CATME's multiline CSV output by parsing each row and categorizing it into specific JSON fields (e.g., activity_info, professor_comments_data, peer_comments_data). By the end of this process, each student's feedback—both from and about the student—resides in a single JSON file that is ready for downstream NLP analysis.

Next, we use evaluate_llama.py to load the JSON data, reorganize it by student, and run inference with two versions of Meta's Llama model:

1. The raw (unfine-tuned) official release, which serves as a baseline without any additional training.
2. A broadly fine-tuned checkpoint provided by Meta, covering general text tasks but not specifically tailored to student evaluations.

We deliberately chose not to perform a domain-specific fine-tuning of Llama in this initial study. Instead, we wanted to assess how well a general-purpose large language model could interpret peer-evaluation data "out of the box." Furthermore, fine-tuning is our planned next step, where we will train the model on a richer dataset of student feedback to improve its ability to detect subtle team-dynamics cues and course-specific language patterns.

To safeguard student privacy, all data processing and model inference occur offline on our private Kubernetes cluster. We never transmit student data to external APIs or third-party services, thus minimizing any risk of leakage. The script evaluate_llama.py encapsulates this offline inference process by loading the final JSON (produced by csv_to_json.py), using a local Llama installation for text generation, and then saving the results into a CSV.

This approach gives us full control over data handling:

- Immediate Anonymization – Before or during the CSV-to-JSON conversion, identifiable student fields (e.g., names, emails) are replaced or hashed (planned for the next iteration) to ensure no personally identifiable information is exposed to the language model.
- GPU Acceleration – We execute the model on an NVIDIA A100 GPU, making it feasible to generate feedback for 100+ students within minutes.
- Modular Design – By splitting parsing logic (csv_to_json.py) from inference logic (evaluate_llama.py), we can easily drop in different models or fine-tuned checkpoints in future experiments.

Prompt Design

To systematically query each model, we constructed a series of eleven prompts. The first ten prompts elicited qualitative feedback about each student's performance [8] [9], focusing on:

1. **Strengths and Weaknesses**
2. **Collaboration and Synergy**
3. **Teamwork Themes**
4. **Independence vs. Integration**
5. **Discrepancies Among Peers**
6. **Alignment or Misalignment with Self-assessment**
7. **Changes Over Time**
8. **Indications of Conflict or Disengagement**
9. **Signs of High Performance**
10. **Short Performance Summary**

The eleventh prompt requested a numeric overall score from 0.0 to 10.0 and a yes/no recommendation on whether the professor should intervene. This final prompt also asked the model to include a concise reason for the intervention decision. Specifically, the instructions required the model to produce a line of the format, so the script could reliably parse and record the results.

```
LLM_SCORE=<number> Intervention=<Yes/No> Reason=<short reason>
```

(1)

Script Workflow

A Python script (see code excerpt below) orchestrated the following steps for **each student**:

1. **Compile Raw Input**
   The script combines each student's inbound peer comments, outbound peer comments, and any comments they wrote to the professor into a single "summary block." This step ensures that all relevant text-based feedback is available for both the model and the computer-based scoring function.
2. **Compute a "Computer-Based" Score via TextBlob**
   To obtain a quick, quantitative snapshot of each student's overall sentiment profile, the script uses the TextBlob library [5][6] to compute sentiment polarity across all comments related to a student. TextBlob is a Python library for processing textual data, and it provides a polarity score in the range −1.0 (negative) to +1.0 (positive)[7]. The script then converts this average polarity to a [0..10] scale using the formula:

$$Score = (avg\_polarity + 1) \times 5 \tag{2}$$

Scores below 0 are clamped to 0, while those exceeding 10 are clamped to 10. If a student has no comments at all, the script defaults to a neutral score of 5.0.

3. **Send Summary + Prompts to Llama** – Each of the eleven prompts was appended to the student's summary block, forming a short "conversation" used as input to the Llama model.
4. **Capture LLM Responses** – The script recorded each model's replies to Prompts #1 through #10, and then parsed the final line of Prompt #11 to extract:
   - LLM_SCORE (0.0–10.0)
   - Intervention (Yes, No)
   - Short rationale
5. **Tabulate Results** – For each student, the script wrote a CSV row containing the student's name, the computer-based score, the model's LLM-based score, and whether the professor should intervene.

Evaluation and Metrics

Time Efficiency

Accuracy and Attention Need

To evaluate how well the model pinpoints students who may need help (e.g., due to subpar engagement, conflict with teammates, or repeated negative feedback), we compared the model's "Intervention=Yes" flags with instructor records from previous semesters where manual analysis identified at-risk students. Agreement rates (i.e., how often the model flagged the same students as the instructor) provided a straightforward measure of accuracy in identifying those needing attention.

Qualitative vs. Quantitative Analyses

- **Quantitative Analysis**: We tracked the distribution of LLM scores (0.0–10.0) and correlated them with simpler "computer-based" scores. Ideally, a more nuanced LLM approach should correlate moderately with raw comment volume but also reflect sentiment and content.
- **Qualitative Analysis**: We examined representative student summaries, comparing LLM responses to known issues. For instance, if a student received consistent peer feedback about poor communication, we checked whether the LLM responses captured that theme and suggested professor intervention.

In practice, the final results for each student explicitly included a line indicating whether the professor should intervene (e.g., Intervention=Yes) and why. This output served as a direct alert system for instructors to prioritize individual follow-up, ensuring that no student "slipped through the cracks" in large classes. The method thus goes beyond mere scoring—it provides actionable recommendations about which students may benefit most from direct professor attention.

Results

Using peer review data from a previous cohort, the model evaluated 66 students based on 11 carefully crafted prompts. In a conversation-style format, the first 10 prompts allowed the model to construct a comprehensive profile of each student based on their self-assessments and peer evaluations. The final prompt leveraged these profiles to assess each student's overall performance and collaboration. This process included calculating two distinct scores for each student: a calculated computer-generated score and an estimated LLM-generated score, both measured on a 0.00-10.00 scale.

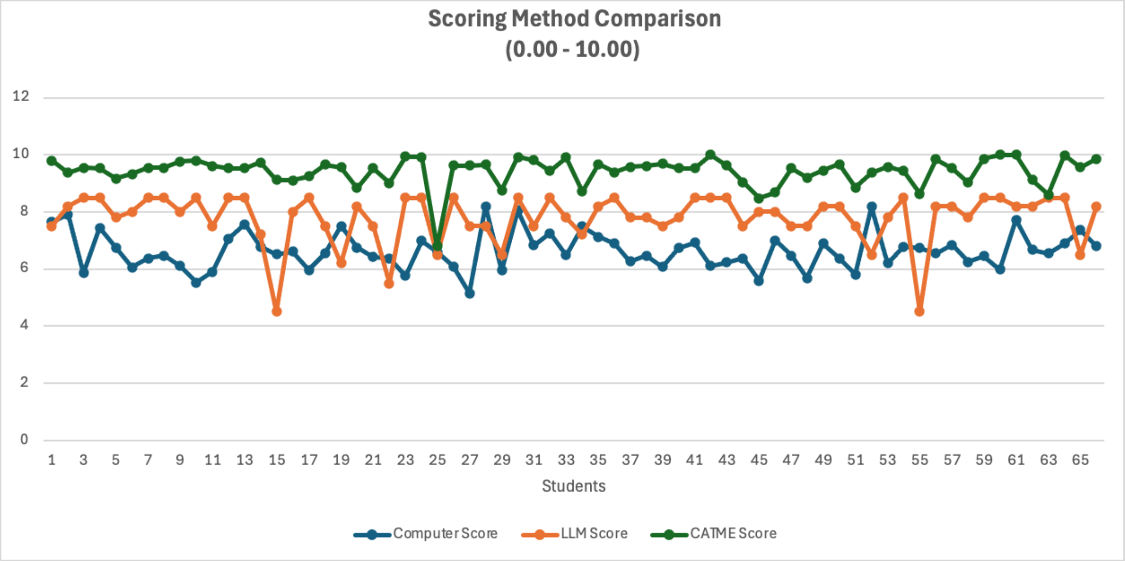

Figure 1: Comparison Between Student CATME, Computer, and LLM Scores

The computer score, derived using the TextBlob library and based on quantitative peer review metrics, serves as the baseline for comparison with the LLM score, which integrates the sentiments expressed in and context of the student feedback. The CATME peer scores, computed on a 1-5 scale using a standardized rubric, are raw numerical averages that exclude consideration of sentiment entirely. These scores, while straightforward, provided minimal insight into team dynamics and individual efforts. To better align the CATME scores with the model's evaluation scale, the raw adjustment factor from the CATME datasheet was applied. This adjustment scaled each student's average score across all five dimensions by dividing it by the team average and converting it to a 0.00-10.00 scale for consistency.

Figure 1 reveals a key pattern in which LLM scores consistently trend higher than computer scores, yet lower than CATME scores across the cohort. Students who received higher LLM scores typically gave and received more positive feedback, suggesting effective collaboration and engagement. Conversely, lower LLM scores correlated with negative or less constructive feedback, often signifying greater challenges in communication or performance.

Compared to both disparities between scoring methods, the model provided median scores, reflecting its sensitivity to qualitative factors, such as positive sentiments in feedback.This reflects the model's sensitivity to qualitative factors, such as positive sentiments in feedback. This highlights the importance of sentiments, as students are scored solely based on *what* is said rather than *how* it is said when it comes to CATME scores.

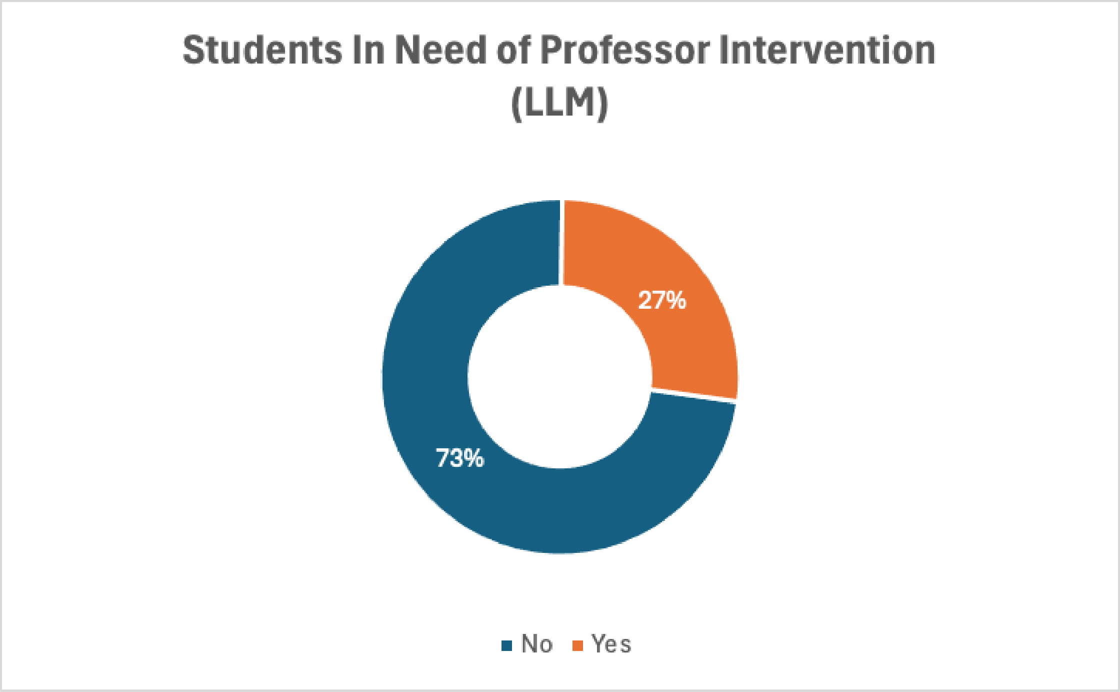

Figure 2: LLM Feedback for Additional Intervention

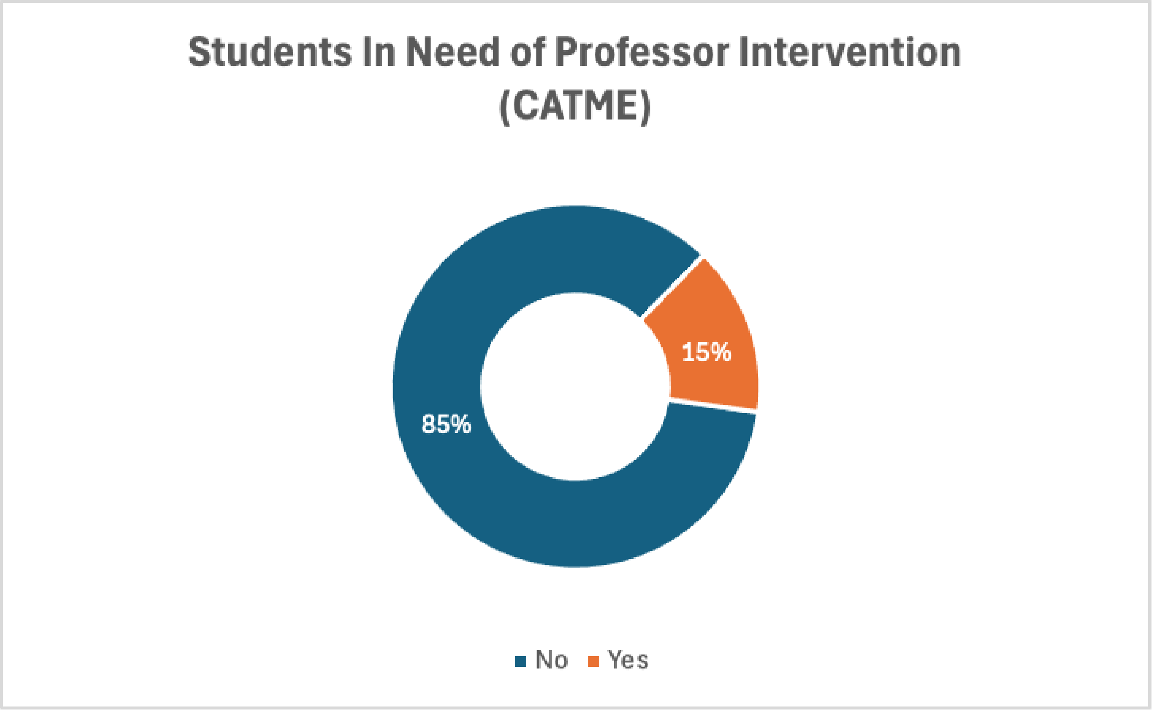

Figure 3: CATME Feedback for Additional Intervention

After evaluating the cohort, CATME flagged 10 out of 66 students (15%) as needing additional intervention from the professor, while the LLM flagged 18 students (27%). Each flagged student received tailored feedback from the LLM, highlighting particular areas of concern. The feedback highlighted specific areas of concern such as breakdowns in communication, disengagement, or workload management, providing actionable insights to guide the professor in offering more targeted support. In contrast, CATME's feedback for flagged students included vague single-worded notes, such as "Under", "Manip" and "Cliq", offering little actionable detail. This comparison highlights the model's ability to prioritize students in need of more focused support while aligning with its role as an assistant to conventional instruction.

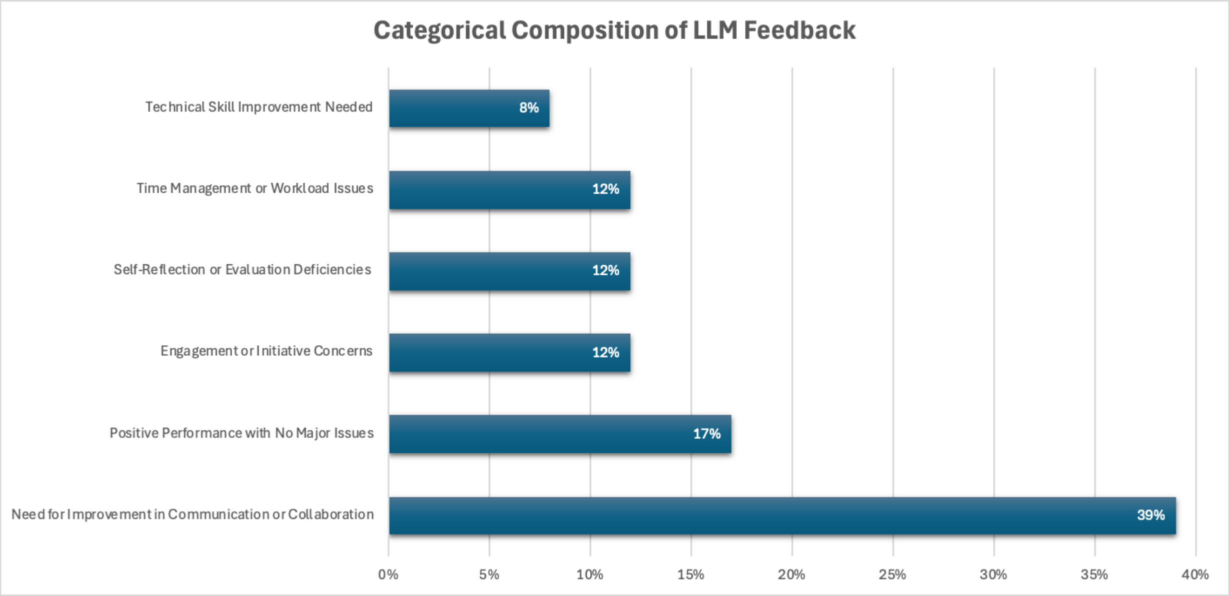

Figure 4: Categorial Composition of LLM Feedback

The most prevalent issue across the cohort was a need for improvement in communication or collaboration skills, accounting for 39% of all feedback (26 students), indicating the diverse nature of feedback needs across the cohort. Other prominent concerns included time management and workload balancing, self-reflection deficiencies, and engagement issues, which collectively comprised 36%. In contrast, positive performance without major issues represented the second largest at 17%, demonstrating the model's balanced approach to evaluating both strengths and weaknesses. Meanwhile, technical skills improvement was noted in 8% of the feedback, suggesting that technical proficiency, while important, was not as widespread an area of concern as interpersonal and project-related skills. This nuanced, actionable feedback allows professors to identify specific areas of improvements for their students that might otherwise go unnoticed in their evaluations due to lack of time and resources.

1. Review results section to ensure accuracy of explanations
    a. Computer scores represent **textblob** (scoring disregards sentiments in feedback
    b. CATME peer scores are raw numerical scores that just averages from scoring evaluations without consideration of sentiments
    c. Since score scale range for CATME evaluations is smaller, better to use textblob for comparison
2. Statistical tests on figure 2 (paired t-test → use 0.5)
3. Include new script results and discussion

Discussion

Conventionally, the instructor would firstly review individual student performance via evaluating their peer feedback and one-on-one appointment. The CATME feedback offers five dimensions: contributions to the team's work, interacting with teammates, keeping the team on track, expecting quality, and having relevant knowledge, skills, and abilities. If the peer rating is lower than 2 on any of the categories, it would raise an alert for the instructor to discuss with the student on future improvement. However, if there is more than one category rated at 2, the instructor would also need to scan through the peer comments and find out the actual cause of the team issues. The AI model would help the instructor with the text scanning and extract information relating to the potential team issues. For example, "rarely show up to the team meeting" would explain why the peers rated this particular student as 2 and lower on categories of contributions to the team's work as well as interacting with teammates. The AI agent would also help to document the trend of student performance as well as the entire class performance. This would give the instructor which student would need more assistance, this would greatly benefit the student learning.

For a cohort of about 100 students, the entire evaluation process—ranging from the submission of student evaluations to the generation of actionable feedback from the professor—typically takes 3-5 days to complete manually. By automating this process, the AI model reduces this time to just 5-10 minutes, offering a dramatic improvement in efficiency.

Performing a paired t-test on the computer and LLM scores ($t=8.36$, $p < 0.0001$) and a second paired t-test on the CATME and LLM scores ($t=-16.0$, $p < 0.0001$), both with a significance level of 0.05, revealed there are significant differences between the score sets. This confirms that the LLM scores differ meaningfully from the other scores generated by traditional numerical analysis, underscoring the value of integrating sentiment and contextual analysis into the evaluation process. Unlike the computer scores, which rely solely on quantitative metrics derived from peer feedback, the LLM incorporates qualitative factors such as tone, sentiment, and other nuanced observations present in the comments. This ability to capture the emotional and contextual elements of student feedback mirrors the professor's manual approach, where qualitative insights are often crucial in identifying underlying issues such as team dynamics, communication challenges, or engagement concerns.

Including sentiment analysis allows the model to provide a more comprehensive view of student performance, better reflecting the complexities of team-based work. By coupling the LLM's efficiency with the professor's expertise, our system flags students who appear disengaged, under-performing, or consistently at odds with their teammates. This "attention-needed" mechanism is critical for early intervention. For instance, if the model detects a series of negative peer comments about a student's communication style, it will mark *Intervention=Yes*, prompting the professor to review additional details. In our pilot test, approximately 27% of students (18 out of 66) were recommended for further action, enabling the instructor to target personalized coaching or arrange one-on-one meetings. The LLM is used as a first-pass tool to save time and resources while flagging students, while professors are expected to review the flagged cases and remain attentive to unflagged students who may require subtle support. This allows students to quickly apply their feedback for improved overall performance and team collaboration.

Despite the promising results of our NLP-enhanced approach, there are several limitations and opportunities for further improvement.

- **Lack of Custom Fine-Tuning**: Neither the raw nor the “officially fine-tuned” Llama model was specifically adapted to student comments. Given that peer evaluations can range from 10 to 100 words and often involve course-specific context, the model can sometimes offer overly generic insights. We plan to address this by fine-tuning on a dataset of anonymized student feedback in our next study, aiming to capture more nuanced team dynamics.
- **Short Comments & Limited Context**: Some peer comments are as brief as one or two sentences, providing limited context. The system’s accuracy could be improved if we encourage students to provide more detailed reflections, or if we integrate additional data sources such as discussion transcripts or project diaries.
- **Privacy Considerations**: Although all inference is done offline using our in-house scripts, we acknowledge that sensitive information might appear in open-ended responses. Our current approach masks obvious identifiers (like names), but we intend to move toward full hashing or token-level anonymization. This would further ensure that no personal details remain in the text when it is passed to the LLM.
- **Hardware Requirements**: Running Llama on an A100 GPU significantly speeds up inference, but poses a computational barrier for instructors without high-end hardware. Future work could explore smaller LLMs or model distillation to reduce overhead, making the solution more accessible to a broader range of educators.
- **Generalizability**: The current pipeline has primarily been tested on an engineering design course with around 100 students. Large-scale validation across diverse subjects, course sizes, and institutions is needed to confirm that the approach generalizes to other domains and learning contexts.

Overall, the system shows considerable potential to streamline large-scale peer evaluation workflows and provide timely, actionable insights to both students and instructors. Our work will support early interventions for at-risk students, strengthen team-based learning, and further safeguard the privacy and integrity of student data.

Conclusion

The AI agent was developed based on Llama 3.0, which is able to assist the course operation via assessing student individual performance and entire class performance. LLM scores differ from the other scores generated by traditional numerical analysis, underscoring the value of integrating sentiment and contextual analysis into the evaluation process. The LLM incorporates qualitative factors such as tone, sentiment, and other nuanced observations present in the comments, this mean the LLM is able to provide more insights on student member interactions.

Acknowledgements

We would like to thank NYU High Speed Research Network for providing access to their Kubernetes cluster. Their high-performance computing resources were instrumental in enabling the rapid data processing and model evaluations that underpinned this project.

References

[1] Bonvillain, N. (2019). *Language, culture, and communication: The meaning of messages*. Rowman & Littlefield.

[2] Chowdhary, K., & Chowdhary, K. R. (2020). Natural language processing. *Fundamentals of artificial intelligence*, 603-649.

[3] Schank, R. C. (1972). Conceptual dependency: A theory of natural language understanding. *Cognitive psychology*, *3*(4), 552-631.

[4] Paris, C. L., Swartout, W. R., & Mann, W. C. (Eds.). (2013). *Natural language generation in artificial intelligence and computational linguistics* (Vol. 119). Springer Science & Business Media.

[5] Loria, S. (2022) Text-Blob: Simplified Text Processing. https://textblob.readthedocs.io/en/dev

[6] Hazarika, Ditiman & Konwar, Gopal & Deb, Shuvam & Bora, Dibya. (2020). Sentiment Analysis on Twitter by Using TextBlob for Natural Language Processing. 63-67. 10.15439/2020KM20.

[7] H R, Dr. (2023). Sentiment analysis: Textblob for decision making.

[8] Sclar, Melanie, et al. "Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting." arXiv preprint arXiv:2310.11324 (2023).

[9] Sahoo, Pranab, et al. "A systematic survey of prompt engineering in large language models: Techniques and applications." *arXiv preprint arXiv:2402.07927* (2024)

[10] Curran, James R., and Stephen Clark. "Language independent NER using a maximum entropy tagger." *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*. 2003.

[11] Loughry, Misty L., Matthew W. Ohland, and David J. Woehr. "Assessing teamwork skills for assurance of learning using CATME team tools." *Journal of Marketing Education* 36.1 (2014): 5-19.

[12] Hrivnak, G. A. "CATME smarter teamwork." (2013): 679-681.

[13] Yadollahi, Ali, Ameneh Gholipour Shahraki, and Osmar R. Zaiane. "Current state of text sentiment analysis from opinion to emotion mining." *ACM Computing Surveys (CSUR)* 50.2 (2017): 1-33.

[14]Shaik, Thanveer, et al. "A review of the trends and challenges in adopting natural language processing methods for education feedback analysis". Ieee Access, vol. 10, 2022, p. 56720-56739. https://doi.org/10.1109/access.2022.3177752

[15] Somers, Rick, et al. "Applying natural language processing to automatically assess student conceptual understanding from textual responses". Australasian Journal of Educational Technology, vol. 37, no. 5, 2021, p. 98-115. https://doi.org/10.14742/ajet.7121

[16] Ahidi Elisante Lukwaro, Elia, et al. "A review on nlp techniques and associated challenges in extracting features from education data". International Journal of Computing and Digital Systems, vol. 15, no. 1, 2024, p. 961-979. https://doi.org/10.12785/ijcds/160170

[17] Bauer, Elisabeth, et al. "Using natural language processing to support peer-feedback in the age of artificial intelligence: a cross-disciplinary framework and a research agenda". British Journal of Educational Technology, vol. 54, no. 5, 2023, p. 1222-1245. https://doi.org/10.1111/bjet.13336

[18] Alhawiti, Khaled M., et al. "Natural language processing and its use in education". International Journal of Advanced Computer Science and Applications, vol. 5, no. 12, 2014. https://doi.org/10.14569/ijacsa.2014.051210

[19] Yin, Wen, et al. "SynPrompt: Syntax-aware Enhanced Prompt Engineering for Aspect-based Sentiment Analysis." *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 2024. https://aclanthology.org/2024.lrec-main.1344/

[20] T. N. Fitria, “ARTIFICIAL INTELLIGENCE (AI) IN EDUCATION: USING AI TOOLS FOR TEACHING AND LEARNING PROCESS”, *prosenas*, vol. 4, no. 1, pp. 134–147, Dec. 2021.https://prosiding.stie-aas.ac.id/index.php/prosenas/article/view/106