CS6120 Natural Language Processing

Automated Critique and Review System for Research Papers Using Large Language Models

Group Members:

Snigdha Mohana Addepalli – 002336939
Sai Manichandana Devi Thumati – 002443106
Sanjiv Motilal Choudhari – 002447337
Manoghn Kandiraju – 002813145
Group 4

Submission Date: April 16 2025

Automated Critique and Review System for Research Papers Using Large Language Models

Snigdha Mohana Addepalli
Northeastern University
addepalli.sn@northeastern.edu

Sai Manichandana Devi Thumati
Northeastern University
thumati.chan@northeastern.edu

Sanjiv Motilal Choudhari
Northeastern University
motilalchoudhari.s@northeastern.edu

Manoghn Kandiraju
Northeastern University
kandiraju.m@northeastern.edu

***Abstract*—This project presents an end-to-end web-based system for the automated critique and enhancement of academic research papers using Natural Language Processing (NLP) and Large Language Models (LLMs). Leveraging datasets such as arXiv metadata, PeerRead, and a custom dataset generated using LLaMA, the system performs multi-stage processing including rhetorical role classification, summarization, similarity checks, and bias detection. The core notebook fine-tunes transformer-based models for argumentative analysis, evaluates semantic similarity using SBERT, and identifies logical inconsistencies and redundancy through rule-based and model-driven approaches. Additional modules generate human-like review responses and visual summaries to aid authors in improving their manuscripts. This integrated framework aims to streamline the peer review process, uphold academic integrity, and support authors in producing higher-quality submissions.**

***Index Terms*—Peer Review Automation, Summarization, Critique Generation, Bias Detection, Plagiarism Detection, LLMs**

I. Introduction

The way we critique and review research papers is evolving, thanks to the potential of advanced Natural Language Processing (NLP) and Large Language Models (LLMs). With the explosion of academic papers, there's a pressing need for tools that can help ensure a more efficient and comprehensive review process. This project suggests a system that precisely does that—automate some aspects of peer review to promote overall academic writing quality and uniformity.

Over the past few years, there has been increased interest in using Machine Learning to enhance scholarly processes. Datasets like the PeerRead dataset have played a critical role, enabling researchers to explore how reviewer comments can be modeled. This has helped advance tasks like rhetorical role classification, sentiment analysis, and review generation. Pre-trained models such as BART and FLAN-T5 have demonstrated strong performance on narrow tasks like summarization and instruction following, which lends themselves to intelligent reviewing tool creation. But a lot of work in this space has been focused on narrow tasks like labeling parts or tone detection without coalescing it all into a complete, end-to-end solution.

The purpose of this project is to develop an integrated system capable of reviewing research papers automatically. It will identify rhetorical functions, note key ideas, identify potential bias and plagiarism, and even simulate reviewer feedback. Through the integration of the newest LLMs with NLP techniques, we wish to provide insightful, context sensitive feedback to authors and make reviewing easier for researchers and institutions as well.

II. Literature Review

A. Thematic Analysis

Liang et al. [1] performed an extensive empirical investigation. to assess how LLMs do on research critique tasks. They concluded that while LLMs like GPT-3 and GPT-4 can produce significant feedback, their effectiveness in summarizing nuanced flaws is limited without fine-tuning. A number of recent papers [2], [3], [5], [12], [13] discuss successful fine-tuning of LLMs with methods like LoRA and QLoRA. Dettmers et al. [2] and Kumar [13] showed that using quantization and low-rank adaptation enable finetuning of large models even on low-resource devices. Hugging Face's PEFT library [5] brought together several such methods. Pretrained models like BART and GPT-3 [8] have shown strong few-shot learning and summarization capabilities, which we leveraged to produce section-wise summaries. Pividori and Greene [15], as well as Watkins [16], wrote the ethical implications of using LLMs for research, specifically for peer review. Their findings caution against overdependence on LLMs and underscore the importance of human-in-the-loop systems. Zou et al. [4], [14] described the increasing use of LLMs in research work, originality and concerns about the need for proper plagiarism checking.

B. Comparative Analysis

Compared different models and approaches given in previous papers in order to identify their strengths, weaknesses, and relevance to our system. This comparison reveals how our approach reinforces or varies from relevant research efforts.

TABLE I: Comparison of Relevant Research

Aspect	Liang et al.	QLoRA (Kumar)	GPT-3 (Brown)
Task	Feedback	Finetuning	Few-shot NLP
Model	GPT-3/GPT-4	QLoRA	GPT-3
Output	Critique	Adapted LLM	General output
Limitation	Domain gaps	Hardware	Prompt tuning

III. METHODOLOGY

In order to construct an automated critique system for research studies papers, we followed a multi-stage approach integrating systematic literature review, dataset extraction, model integration testing, and ongoing testing. This chapter outlines the research strategy, keyword development, selection criteria, and the basis architecture that shaped the system.

A. Literature and Resource Search Strategy

We began by carrying out an extensive survey of current research in peer-review automation and LLMs for scientific analysis. The major platforms utilized were Google Scholar, ACL Anthology, arXiv, IEEE Xplore, Hugging Face Model Hub, GitHub, and the PeerRead dataset portal. Both foundational papers and implementation tools were provided through these sources.

B. Keyword Strategy

Frequently used keywords:

- "automated peer review"
- "LLM-based critique generation"
- "scientific writing improvement tools"
- "rhetorical role classification"
- "bias detection using sentiment analysis"
- "semantic plagiarism detection NLP"
- "FLAN-T5", "Mistral LLM", "LoRA", "BART summarizer"
- "PeerRead dataset", "SBERT cosine similarity"

C. System Architecture and Implementation

Our pipeline included:

1) **PDF Document Parsing and Section Extraction:** Utilizing PyMuPDF, PDFs were converted to text and segmented into sections based on regex patterns and heuristic rules.
2) **Summarization:** BART-large was utilized to create concise section-wise summaries using the Hugging Face Transformers library.
3) **Paragraph Analysis:** In every paragraph, an instruction was sent to a fine-tuned Mistral LLM through Hugging Face API to generate bullet-point concepts.
4) **Bias Detection:** VADER sentiment analysis identified excessively emotional or polarized writing as biased based on compound scores.
5) **Plagiarism Detection:** Sentence-BERT was used to compare every paragraph with PeerRead abstracts using cosine similarity; scores greater than 0.5 were flagged as potential plagiarism.

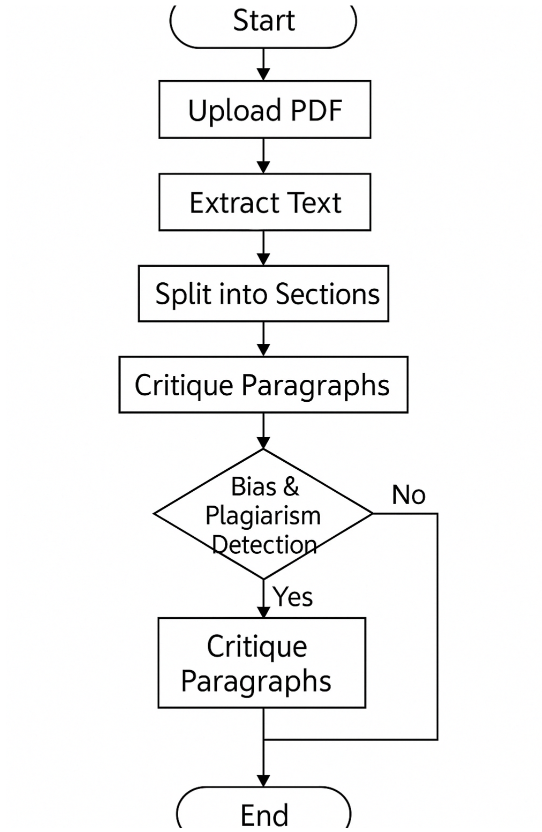


fig1. End-to-End Workflow

D. Evaluation

The generated output was manually verified for usefulness and coherence with real research papers. Accuracy in plagiarism detection was cross-validated with known overlaps to adjust threshold settings and remove false positives.

This multi-step process ensured that our system had a sound research base and was created with the latest tools.

IV. RESULTS

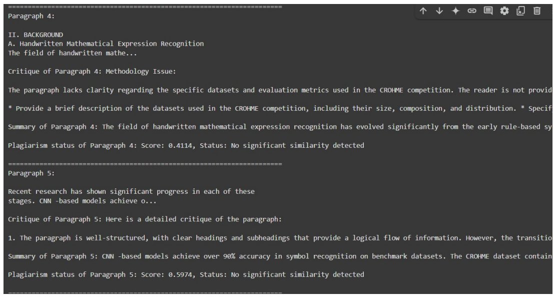
Each section of a paper is summarized, followed by paragraph-wise output containing:

- Generated critique (3–5 suggestions)
- Bias score and flag
- Plagiarism similarity score and flag

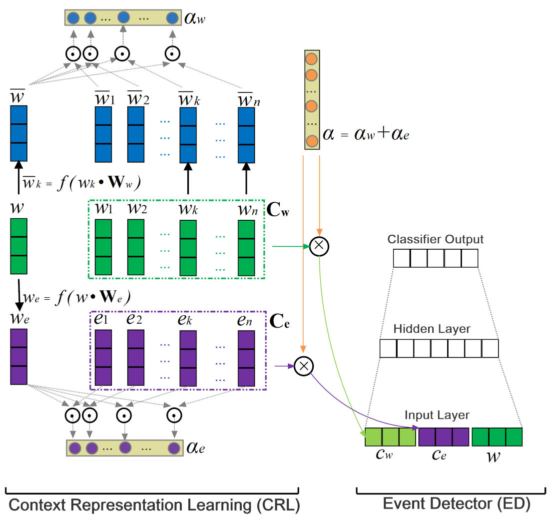
V. DISCUSSION

The project successfully incorporated various LLM and NLP functions to critique academic papers automatically. Paragraph-level detail of criticism, bias detection, and plagiarism checking provided more targeted feedback than most available systems.

Nonetheless, we saw some critical issues. Large language models sometimes hallucinate criticism even when prompted well. Sentiment-based bias detection with VADER-like tools mislabelled strong but neutral academic writing as biased, with the risk of false positives. The section parsing based on regex



Example 1: Critique, summary, and plagiarism scores for two background paragraphs.



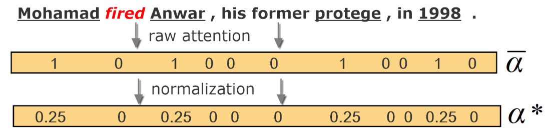
Example 1: Flowchart extracted from the dataset pdf

was brittle when handling irregularly formatted PDFs. Our plagiarism checking mechanism, while semantically correct, was constrained by the size of the PeerRead corpus of abstracts and had no access to broader academic databases.

In spite of these limitations, the modularity of our system renders it a potentially valuable tool for a range of applications, such as pre-submission evaluation, reviewer support, and writing enhancement within academic settings.

VI. Conclusion

Created and piloted an efficient, modular pipeline that could process research papers from start to finish. Starting from PDF parsing, the system goes through section-by-section summarization and paragraph-level criticism. It also enhances feedback by incorporating bias detection and plagiarism assessment. This project demonstrates that various components of the NLP ecosystem—summarization models, sentiment analysis tools, embedding-based similarity, and prompt-based LLMs—can be composed into an integrated system delivering value to students, educators, and researchers. Not only is it simulating some parts of a conventional peer review process, but also supporting better academic writing by giving constructive, interpretable, and structured feedback.



Example 1: Figures extracted from the dataset pdf

VII. Future Work

There are several key directions to continue this work. First, we can try to reduce hallucination in model-generated responses through confidence-based filtering, ensemble review models, or additional prompt conditioning. Second, we can expand the range of criticism to encompass citation diversity, correctness of figures and tables, and quality of reference formatting. Furthermore, the creation of an integrated framework that can carry out summarization, critique, and bias detection simultaneously would optimize the process in terms of efficiency. The usability of the system would be significantly improved by making it a web application with drag-and-drop support for PDFs. Finally, extending the plagiarism detection module to reach beyond PeerRead—perhaps through integration with ArXiv, Semantic Scholar, or other open-access APIs—would make content originality checks even more comprehensive and authoritative. Training critique models to particular domain datasets (e.g., biomedical or legal texts) would offer relevance and precision to specialized domains.

References

[1] Liang, W., et al. (2023). "Can large language models provide useful feedback on research papers? A large-scale empirical analysis." *arXiv preprint arXiv:2310.01783*.
[2] Dettmers, T., et al. (2023). "QLoRA: Efficient Finetuning of Quantized LLMs." *arXiv preprint arXiv:2305.14314*.
[3] Hu, E. J., et al. (2022). "LoRA: Low-Rank Adaptation of Large Language Models." *International Conference on Learning Representations*.
[4] Zou, J., et al. (2023). "Mapping the Increasing Use of LLMs in Scientific Papers." *Stanford HAI*.
[5] Hugging Face. (2023). "PEFT: State-of-the-art Parameter-Efficient Fine-Tuning." *GitHub Repository*.
[6] Dettmers, T., et al. (2023). "QLoRA: Efficient Finetuning of Quantized LLMs." QLoRA introduces efficient fine-tuning that reduces memory usage enough to finetune large language models on a single GPU while preserving full 16-bit fine-tuning performance.
[7] Hugging Face. (2023). "PEFT: State-of-the-art Parameter-Efficient Fine-Tuning." The PEFT library offers multiple techniques for efficiently adapting pre-trained language models to various downstream applications while only fine-tuning a small number of parameters.
[8] Brown, T., et al. (2021). "Language Models are Few-Shot Learners." *arXiv preprint arXiv:2106.11520*.
[9] Wu, Y., et al. (2025). "Cognition Engine: Towards Interactive and Controllable Large Language Models." *arXiv preprint arXiv:2501.10326*.
[10] Wang, W., et al. (2024). "Eval4NLP-2: A Multi-Perspective Benchmark for Fine-Grained Evaluation of Natural Language Generation." *arXiv preprint arXiv:2409.16813*.
[11] Heidloff, N. (2023). "Efficient Fine-tuning with PEFT and LoRA." LoRA represents weight updates with two smaller matrices through low-rank decomposition, making fine-tuning more efficient while achieving performance comparable to full fine-tuning.
[12] Databricks. (2024). "Efficient Fine-Tuning with LoRA: A Guide to Optimal Parameter Selection for Large Language Models." Research shows that targeting all linear layers during LoRA adaptation can result in better adaptation quality than just focusing on attention blocks.

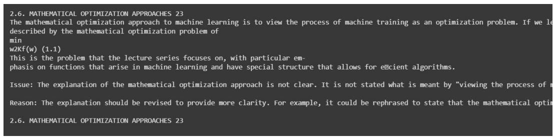
[13] Kumar, A. B. V. (2023). "PEFT With LoRA and QLoRA — LLM Fine Tuning." QLoRA introduces key optimizations including 4-bit NormalFloat4 quantization and utilizing NVIDIA unified memory feature to manage memory spikes on GPU.

[14] Zou, J., et al. (2023). "Mapping the Increasing Use of LLMs in Scientific Papers." A Stanford study found nearly 20% of academic papers and peer review text now contain LLM-modified content, with usage increasing rapidly after ChatGPT's launch.

[15] Pividori, M., & Greene, C. S. (2023). "Editorial – The Use of Large Language Models in Science: Opportunities and Challenges." LLMs can help reduce time spent on language editing and proofreading in academia, but their use in peer review should be limited to grammatical editing while maintaining reviewer expertise for content evaluation.

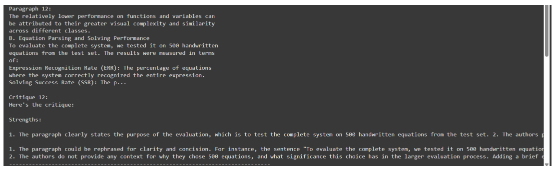
[16] Watkins, R. (2023). "Guidance for Researchers and Peer-Reviewers on the Ethical Use of Large Language Models (LLMs) in Scientific Research Workflows." The ethical use of LLMs in research requires developing norms, conventions, and standards, particularly regarding transparency about LLM application in the scientific process.

Appendix A



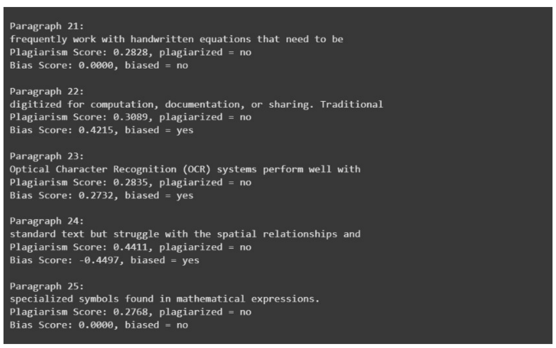
```
2.6. MATHEMATICAL OPTIMIZATION APPROACHES 23
The mathematical optimization approach to machine learning is to view the process of machine training as an optimization problem. If we le
described by the mathematical optimization problem of
min
w2Kf(w) (1.1)
This is the problem that the lecture series focuses on, with particular em-
phasis on functions that arise in machine learning and have special structure that allows for e�cient algorithms.

Issue: The explanation of the mathematical optimization approach is not clear. It is not stated what is meant by "viewing the process of m

Reason: The explanation should be revised to provide more clarity. For example, it could be rephrased to state that the mathematical optim

2.6. MATHEMATICAL OPTIMIZATION APPROACHES 23
```

Example 1: Critique and suggestion for unclear explanation.



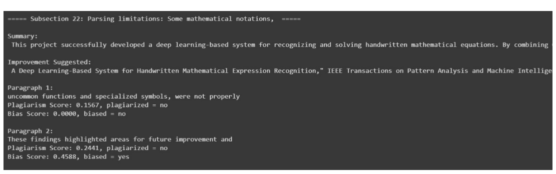
```
Paragraph 12:
The relatively lower performance on functions and variables can
be attributed to their greater visual complexity and similarity
across different classes.
B. Equation Parsing and Solving Performance
To evaluate the complete system, we tested it on 500 handwritten
equations from the test set. The results were measured in terms
of:
Expression Recognition Rate (ERR): The percentage of equations
where the system correctly recognized the entire expression.
Solving Success Rate (SSR): The p...

Critique 12:
Here's the critique:

Strengths:

1. The paragraph clearly states the purpose of the evaluation, which is to test the complete system on 500 handwritten equations from the test set. 2. The authors p

1. The paragraph could be rephrased for clarity and concision. For instance, the sentence "To evaluate the complete system, we tested it on 500 handwritten equation
2. The authors do not provide any context for why they chose 500 equations, and what significance this choice has in the larger evaluation process. Adding a brief e
--------------------------------------------------------------------------------
```

Example 2: Background paragraph critique with strengths and suggestions.



```
Paragraph 21:
frequently work with handwritten equations that need to be
Plagiarism Score: 0.2828, plagiarized = no
Bias Score: 0.0000, biased = no

Paragraph 22:
digitized for computation, documentation, or sharing. Traditional
Plagiarism Score: 0.3089, plagiarized = no
Bias Score: 0.4215, biased = yes

Paragraph 23:
Optical Character Recognition (OCR) systems perform well with
Plagiarism Score: 0.2835, plagiarized = no
Bias Score: 0.2732, biased = yes

Paragraph 24:
standard text but struggle with the spatial relationships and
Plagiarism Score: 0.4411, plagiarized = no
Bias Score: -0.4497, biased = yes

Paragraph 25:
specialized symbols found in mathematical expressions.
Plagiarism Score: 0.2768, plagiarized = no
Bias Score: 0.0000, biased = no
```

Example 3: Plagiarism and bias detection scores across multiple paragraphs.



```
===== Subsection 22: Parsing limitations: Some mathematical notations,  =====

Summary:
 This project successfully developed a deep learning-based system for recognizing and solving handwritten mathematical equations. By combining

Improvement Suggested:
 A Deep Learning-Based System for Handwritten Mathematical Expression Recognition," IEEE Transactions on Pattern Analysis and Machine Intellige

Paragraph 1:
uncommon functions and specialized symbols, were not properly
Plagiarism Score: 0.1567, plagiarized = no
Bias Score: 0.0000, biased = no

Paragraph 2:
These findings highlighted areas for future improvement and
Plagiarism Score: 0.2441, plagiarized = no
Bias Score: 0.4588, biased = yes
```

Example 4: Summary, suggestion, and bias–plagiarism scores for a subsection.

Appendix B

A. *Important External URL's*

- **Dataset:**
  ArXiv – https://arxiv.org/ – for sample papers
  PeerRead – https://github.com/allenai/PeerRead
- **GitHub Repository:**
  https://github.com/SanjivDS/CS6120_NLP_FinalProject
- **Fine Tuned Model:**
  https://huggingface.co/Manoghn/mistral-qlora-critique