MixRevDetect: Towards Detecting AI-Generated Content in Hybrid Peer Reviews

Sandeep Kumar†, Samarth Garg‡, Sagnik Sengupta¶*, Tirthankar Ghosal§, Asif Ekbal† ♢
† Department of Computer Science, Indian Institute of Technology Patna
‡ Atal Bihari Vajpayee Indian Institute of Information Technology and Management, Gwalior
¶ Manipal Institute of Technology, India
§National Center for Computational Sciences, Oak Ridge National Laboratory, USA
♢ School of AI and Data Science, IIT Jodhpur, India
†sandeep_2121cs29@iitp.ac.in, †♢asif@{iitp,iitj}.ac.in

Abstract

The growing use of large language models (LLMs) in academic peer review poses significant challenges, particularly in distinguishing AI-generated content from human-written feedback. This research addresses the problem of identifying AI-generated peer review comments, which are crucial to maintaining the integrity of scholarly evaluation. Prior research has primarily focused on generic AI-generated text detection or on estimating the fraction of peer reviews that may be AI-generated, often treating reviews as monolithic units. However, these methods fail to detect finer-grained AI-generated points within mixed-authorship reviews. To address this gap, we propose MixRevDetect, a novel method to identify AI-generated points in peer reviews. Our approach achieved an F1 score of 88.86%, significantly outperforming existing AI text detection methods. We make our dataset and code public[1].

1 Introduction

The rapid development of large language models (LLMs) has brought about significant advances in natural language generation, including applications in diverse fields, such as content creation, code generation, and academic peer review. As academic publishing grows in complexity and volume, researchers have increasingly turned to LLMs to assist in automating or augmenting the peer review process. While these models can generate insightful points, critiques, and suggestions at scale, the use of AI-generated content in peer reviews raises critical concerns about the authenticity, quality, and ethical implications of such reviews. In particular, distinguishing between human-generated and AI-generated review points has emerged as a critical challenge for maintaining the integrity of the peer review process.

A study (Liang et al., 2024) found that LLMs may have significantly influenced 6.5% to 16.9% of peer-review text in AI conferences. ChatGPT usage spikes near review deadlines, especially among reviewers who skip rebuttals, and is linked to lower self-reported confidence. Additionally, Springer retracted 107 cancer papers due to compromised peer-review processes involving fake reviewers (Chris Graf, 2022). Previous work (Kumar et al., 2024) has primarily investigated methods for detecting fully AI-generated peer reviews. However, in practical scenarios, a reviewer may write some review points themselves while relying on AI to generate others. So, we ask the question below:-

> What if peer reviews are a mix of AI and Human points?

In such cases, it becomes crucial to detect which specific review points are written by the reviewer and which are generated by AI. By addressing the challenge of detecting AI-generated peer review points, this work aims to contribute to the ongoing discourse on the ethical and practical implications of AI in academic publishing. We propose a framework for systematically evaluating peer review content, offering solutions that can be integrated into existing editorial workflows to enhance transparency, accountability, and trust in the peer review process.

Our contributions are summarized as follows:-

- We propose a novel idea of AI-based text detection of peer review comments (when the review is a mix of AI and Human).
- We design a novel method of review pruning and completion to solve this task.
- Our results show an 88.86% F1 score in detecting AI-based peer review comments.

* This work was done during internship at IIT Patna.
[1] https://github.com/sandeep82945/AI-text-Points

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), pages 944–953
April 29 - May 4, 2025

2 Related Work

Early approaches utilized metrics such as entropy (Lavergne et al., 2008), log-probability scores (Solaiman et al., 2019), perplexity (Beresneva, 2016), and rare n-gram frequencies (Badaskar et al., 2008) to differentiate between human and machine-generated text. Recent advancements like DetectGPT (Mitchell et al., 2023) suggest that AI-generated content often resides in regions with negative log probability curvature. Fast-DetectGPT (Bao et al., 2023b) enhances efficiency by employing conditional probability curvature. Research by Tulchinskii et al. (Tulchinskii et al., 2023) shows that AI-generated text tends to have lower intrinsic dimensionality than human writing.

Few studies applied classifiers to detect synthetic text in contexts like peer review corpora (Bhagat and Hovy, 2013), media outlets (Zellers et al., 2019), and various domains (Uchendu et al., 2020; Bakhtin et al., 2019). GPT-Sentinel (Chen et al., 2023), trained classifiers like RoBERTa (Liu et al., 2019) and T5 (Raffel et al., 2020) on the OpenGPT-Text dataset. GPT-Pat (Yu et al., 2023) uses a siamese neural network to measure the similarity between original and re-decoded text. Li et al. (Li et al., 2023a) developed a large-scale testbed by collecting human and AI-generated texts from multiple sources. Additionally, contrastive and adversarial learning techniques have been introduced to enhance classifier robustness (Bhattacharjee et al., 2023; Hu et al., 2023a; Liu et al., 2022).

Watermarking offers a method for detecting AI-generated text by embedding identifiable signals directly into the text. Early techniques modified existing text through synonym substitution (Chiang et al., 2003), syntactic restructuring (Topkara et al., 2006; Atallah et al., 2001), or paraphrasing (Atallah et al., 2002). Watermarking typically requires active involvement from the model or service provider and may risk degrading text quality, potentially impacting the coherence and depth of LLM responses (Singh and Zou, 2023).

Our work differs from previous studies as we focus on detecting peer review points. A recent paper on AI-generated peer review detection (Kumar et al., 2024) focuses on determining whether the entire review is AI-generated. In contrast, our work focuses on identifying cases where a review contains a mix of human and AI-generated comments. This hybrid nature presents unique challenges that traditional AI-text detection models fail to address. To bridge this gap, we propose MixRevDetect, the first method explicitly designed to detect AI-generated review points rather than classifying entire reviews, enabling fine-grained AI detection within peer review comments.

3 Methodology

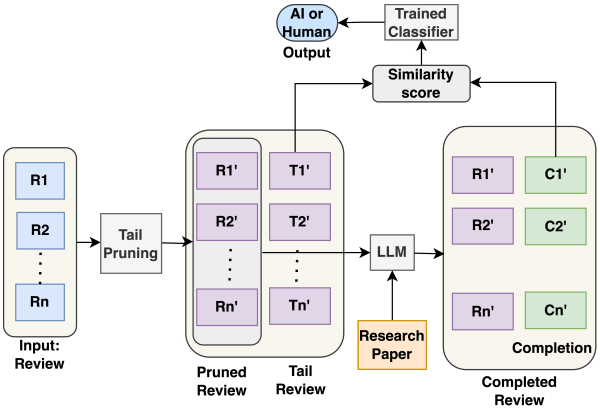

Figure 1: Overall architecture of the proposed method.

Figure 1 illustrates our proposed method's architecture. First, a review R is divided into review comments $R_1, R_2, R_3, \ldots, R_n$ (here, the review comments represent the strengths and weaknesses mentioned by the reviewer). These review comments are then trail-pruned into pruned review comments $R'_1, R'_2, R'_3, \ldots, R'_n$ and tail review comments $T'_1, T'_2, T'_3, \ldots, T'_n$. The pruned review comments $R'_1, R'_2, R'_3, \ldots, R'_n$, along with the completion prompt and the research paper, are passed through a language model to generate the completions $C'_1, C'_2, C'_3, \ldots, C'_n$. Finally, we calculate the similarity between each completion C_i and its corresponding tail T_i. Then, we pass the result through a trained classifier to detect whether the review comment was AI-generated or human-written. We explain the components of our methodology—Tail Pruning, Completion, Similarity Evaluation, and Classification—below:

3.1 Tail Pruning

We apply a pruning process for each sentence $s \in S$ to simulate incomplete information. Let α be the tail pruning ratio, where $0 < \alpha < 1$. We remove $\alpha|s|$ tokens from the tail of each sentence, where $|s|$ denotes the length of the sentence in tokens. We denote the tail-pruned sentence as s_t:

$$s_t = \text{pruning}(s, \alpha|s|). \quad (1)$$

Details on choosing the value of α and the effect of varying the tail pruning ratio are discussed

in Section 4.4. This pruning simulates a scenario where only the initial portion of the sentence is available, and we aim to generate the missing content.

As illustrated in Figure 3 in Appendix D, tail pruning involves pruning each sentence to simulate incomplete information. For example, a review sentence like:

"The introduction of the AMDKD scheme is a novel approach to enhancing the generalization of deep models for VRPs."

is pruned to:

"The introduction of the AMDKD scheme is a novel approach to enhancing the generalization of deep",

effectively masking the tail end of the sentence. The pruned sentence is then used as input for the completion process.

To explain how pruning helps in isolating indicative aspect categories of the reviews (Ghosal et al., 2022) (For example Presentation and Formatting, clarity, novelty, etc) , we provide the following examples. Consider the review sentence:

"The study introduces novel embedding schemes and || empirically demonstrates their effectiveness in improving model performance..." Here, the pruned review sentence before truncation (*"The study introduces novel embedding schemes and"*) already contains an implicit indicator that the expected completion should focus on the novelty aspect category. In our analysis, we found that in most cases, the pruned review text provides sufficient context to guide the generation of a completion that aligns with the appropriate aspect category.

3.2 Completion

We use the GPT-4o model to generate completions for the tail-pruned sentences. The prompt used for the completion is shown in the Appendix D.

The completion function CF can be represented as:

$$C_i = CF(R'_i, P), \tag{2}$$

where C_i is the completed review comment, and P is the content of the research paper associated with the review. The model is prompted to complete the tail review comment R'_i utilizing the context of the paper P.

3.3 Similarity Evaluation

BERTScore, based on contextual embeddings, is designed to measure semantic similarity and performs effectively even with partial sentence fragments, as its focus is on meaning rather than syntactic structure. Our tail-pruning approach ensures that the sentence suffixes retain sufficient semantic context, allowing BERTScore to evaluate the fidelity of generated completions to the intended continuation. To evaluate the similarity between the tail review comment T'_i and C'_i, we employ BERTScore (Zhang et al., 2019) that measures the semantic similarity between two texts using contextual embeddings from BERT. It returns precision, recall, and F1-score based on the matching of tokens in the embedding space:

$$B(T_i, C'_i) = (\text{Precision}, \text{Recall}, \text{F1-score}) \tag{3}$$

3.4 Classification of Sentences

We use a classifier that applies the sigmoid activation function to linear combinations of input features to differentiate between AI-generated and human-written sentences based on similarity metrics. The input features $\mathbf{X}$ for this classifier consist of:

$$\mathbf{X} = [B_{\text{Precision}}, B_{\text{Recall}}], \tag{4}$$

where $B_{\text{Precision}}$ and B_{Recall} represent the BERTScore precision and recall, respectively.

The sigmoid layer of the MLP model M predicts the probability P of a sentence being human-written:

$$P(\text{human} \mid \mathbf{X}) = \sigma(\mathbf{W}^\top \mathbf{X} + b), \tag{5}$$

Here, σ is the sigmoid function, $\mathbf{W}$ represents the learned weights and b is the bias term.

4 Experiments

4.1 Data Collection

We collected 1,000 papers and their corresponding human-written peer reviews from NeurIPS 2022, prior to the release of advanced models like ChatGPT, to avoid AI influence. Using the same set of papers, we also generated AI-written reviews. Figure 4 illustrates the length distribution of the reviews in our dataset. The dataset is split into training (70%), validation (10%), and test (20%) sets. We discuss this in detail in Appendix Section 4.

4.2 Experimental Setup

The logistic classifier, with three hidden layers, is trained for 100 epochs using the collected dataset of similarity metrics for both AI-generated and human-written sentences. We evaluate the classifier's performance in distinguishing between AI-generated and human-written sentences based on standard metrics, i.e., precision, recall, and F1 score.

4.2.1 Results and Analysis

We compare the results of MixRevDetect with those of RADAR, DEEPFAKE, DETECT GPT, and LLMDET. We discuss the details of the baselines in Appendix A.

4.3 Main Results

The results presented in Table 1 indicate that our proposed method achieves an F1 score of 0.8886, representing a 27.5% improvement over the best-performing baseline model, FAST-DETECT GPT, which has an F1 score of 0.6968. Compared to DEEP-FAKE and LLMDET, with F1 scores of 0.6755 and 0.6536, our method shows relative improvements of 31.5% and 35.9%, respectively. The most significant improvement is observed against the RADAR model, where our method achieves a 112.3% increase over its F1 score of 0.4186. These results highlight the effectiveness of our approach compared to existing models.

Model	P	R	F1
RADAR (Hu et al., 2023b)	0.5744	0.3292	0.4186
LLMDET (Wu et al., 2023)	0.5942	0.7257	0.6536
DEEP-FAKE (Li et al., 2023b)	0.6345	0.6750	0.6755
FAST-DETECT GPT (Li et al., 2023b)	0.6580	0.7054	0.6968
MixRevDetect	**0.8799**	**0.8982**	**0.8886**

Table 1: Comparison result of our proposed method

4.4 Effect of Changing the Tail Pruning Ratio

The tail pruning ratio is the portion of review comments that are removed from the end. We investigated the effect of the tail pruning ratio on the F1 score. Figure 2 shows the result of the tail pruning ratio on the F1 score. As the tail pruning ratio decreases, meaning that fewer of the review comments are pruned, there is a significant fluctuation in the F1 score. A tail pruning ratio of 0.7 yields the highest F1 score at 0.884, suggesting that this level of pruning provides the optimal balance between retaining relevant information and avoiding noise from excessive comments. On the other hand, reducing the tail pruning ratio further results in a sharp drop in performance, with an F1 score of 0.721 at a ratio of 0.5. However, as the pruning ratio approaches 0.1, the F1 score improves slightly, reaching 0.758, though it never regains the performance seen at higher ratios.

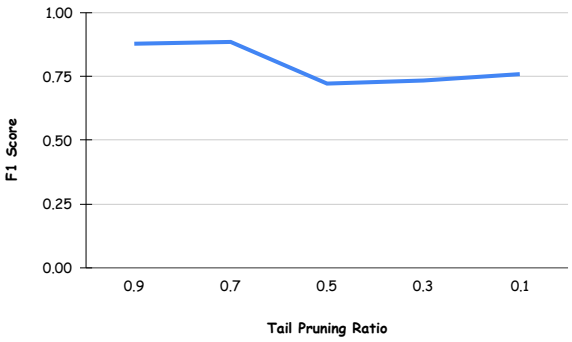

Figure 2: Tail Pruning Ratio vs. F1 Score

4.5 Effect of Paraphrasing

Reviewers can potentially paraphrase their AI-generated review comments to evade AI-based detection systems. To address this, we also incorporated an evaluation of paraphrasing to better understand its impact on detection accuracy.

Specifically, we used the following prompt to paraphrase the review comments:

> Paraphrase the review comment below such that it looks like it is human written.

We employed the LLaMA 70B (Touvron et al., 2023) model with this prompt to generate the paraphrased review comments.

The comparison between the non-paraphrased and paraphrased results shows that all baseline models experience a notable decline in performance, especially DEEP-FAKE (47.77% drop) and FAST-DETECT GPT (38.17% drop). The LLMDET model also suffers a considerable reduction of 37.00%. On the other hand, the RADAR model shows a moderate drop of 6.92%, and our Proposed Method shows the smallest drop of only 6.34%, maintaining its superiority in generalization across the paraphrased tasks.

Model	P	R	F1
RADAR (Hu et al., 2023b)	0.5051	0.3171	0.3896
DEEP-FAKE (Li et al., 2023b)	0.4045	0.3125	0.3528
LLMDET (Wu et al., 2023)	0.5121	0.3438	0.4117
FAST-DETECT GPT (Li et al., 2023b)	0.5364	0.3601	0.4309
MixRevDetect	**0.8462**	**0.8201**	**0.8322**

Table 2: Comparison results after paraphrasing.

4.6 Analysis of BERTScore Trends

To validate whether BERTScore effectively differentiates AI-generated and human-written reviews, we analyzed cases where high and low BERT scores correspond to AI or human completions, respectively. We provide examples that illustrate these trends in Appendix B.

4.7 Error Analysis

We also conducted human analyses to understand when and why our models fail. Our model sometimes fails when paraphrasing alters the style or when AI-generated reviews closely resemble human writing, resulting in low similarity scores and incorrect predictions. We discuss this extensive error analysis in the Appendix C.

5 Conclusion and Future Work

In this paper, we addressed the growing concern of AI-generated peer reviews by focusing on detecting hybrid reviews where both AI and human-authored comments are present. We proposed the MixRevDetect framework, which leverages tail pruning, completion through LLMs, and similarity evaluation to distinguish between AI-generated and human-written peer review points. Our approach demonstrated a significant improvement in detection performance, achieving an F1 score of 88.86%, outperforming existing baselines by a large margin. Future research could explore the performance of MixRevDetect across a wider variety of LLMs, particularly as new models emerge. An interesting direction for future work is to categorize the 'human' dataset based on different topics and analyze how the results vary across these categories.

Limitations

This study mainly relied on GPT-4o for generating AI-generated texts, given its widespread use as an LLM for long-context content generation. We suggest that future researchers select the LLM that most closely matches the model likely used in generating their target corpus to better capture the usage trends prevalent during its creation.

Ethics Statement

We have utilized an open-source dataset for this study. We neither suggest that using AI tools for drafting reviews is inherently good or bad nor do we provide conclusive evidence that reviewers are using ChatGPT to compose reviews. The primary goal of this system is to assist editors in identifying potentially AI-generated reviews, and it is intended solely for internal use by editors, not for authors or reviewers.

Our model generates a completed review using LLMs based on the paper's content. Open-source LLMs running locally do not pose privacy concerns. OpenAI has implemented a Zero Data Retention policy to protect data security and privacy, and users of ChatGPT Enterprise can manage data retention periods themselves[2]. Additionally, many papers are publicly available on platforms like arXiv[3]. However, editors and chairs should exercise caution when using this tool, mindful of the potential risks to privacy and anonymity.

The system cannot detect all AI-generated reviews and may produce false negatives, so it should not be used as the sole decision-making tool. Results should be thoroughly verified and analyzed before any conclusions are drawn. We hope our data and analysis will foster constructive discussions within the community and contribute to preventing AI misuse.

Acknowledgement

Sandeep Kumar acknowledges the Prime Minister Research Fellowship (PMRF) program of the Govt of India for its support. We acknowledge Google for the "Gemma Academic Program GCP Credit Award", which provided Cloud credits to support this research.

References

Mikhail J Atallah, Colleen McDonough, Sergei Nirenburg, and Victor Raskin. 2001. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *Information Hiding*, pages 185–199. Springer.

Mikhail J Atallah, Victor Raskin, Christian Hempelmann, Mustafa Karahan, and Florian Kerschbaum. 2002. Natural language watermarking: Preserving meaning and reconstructing order. In *International Workshop on Information Hiding*, pages 196–212. Springer.

Shantanu Badaskar et al. 2008. N-gram based methods for detecting machine-generated text. In *Proceedings of the 2008 AAAI Workshop on AI-generated Content*.

[2] https://openai.com/index/introducing-chatgpt-enterprise/

[3] https://arxiv.org/

Mikhail Bakhtin et al. 2019. Domain-specific classifiers for machine-generated text detection. In *Proceedings of the 2019 Annual Conference on Neural Information Processing Systems*.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023a. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*.

Jiaxin Bao et al. 2023b. Fast-detectgpt: Enhancing detection efficiency with conditional probability curvature. In *Proceedings of the 2023 International Conference on Learning Representations*.

Olga Beresneva. 2016. Using perplexity to detect machine-generated text. *Journal of Natural Language Processing*, 23(2):34–45.

Rahul Bhagat and Eduard Hovy. 2013. Detecting machine-generated text in peer review corpora. In *Proceedings of the 2013 NAACL-HLT Conference*.

Sankha Bhattacharjee et al. 2023. Adversarial learning for robust ai-generated text detection. *arXiv preprint arXiv:2304.07812*.

Ji Chen et al. 2023. Gpt-sentinel: A robust approach to ai-generated text detection. In *Proceedings of the 2023 ACL Conference*.

Yao-Jen Chiang, Richard Chow, and Wesley Chu. 2003. Watermarking techniques for tree structured data. In *Proceedings of the 2003 ACM workshops on Multimedia*, pages 370–374. ACM.

The Editor Engagement Chris Graf. 2022. Upholding research integrity and publishing ethics – identifying ethical concerns.

Tirthankar Ghosal, Sandeep Kumar, Prabhat Kumar Bharti, and Asif Ekbal. 2022. Peer review analyze: A novel benchmark resource for computational analysis of peer reviews. *PLOS ONE*, 17(1):1–29.

Kai Hu et al. 2023a. Towards robust ai-generated text detection: An adversarial learning approach. *arXiv preprint arXiv:2305.03872*.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023b. RADAR: robust ai-text detection via adversarial learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Sandeep Kumar, Mohit Sahu, Vardhan Gacche, Tirthankar Ghosal, and Asif Ekbal. 2024. 'quis custodiet ipsos custodes?' who will watch the watchmen? on detecting ai-generated peer-reviews. In *arXiv preprint arXiv:2410.09770*.

Thomas Lavergne et al. 2008. Entropy-based detection of machine-generated text. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*.

Hang Li et al. 2023a. Wild testbeds for ai-generated text detection: Lessons from human and deepfake texts. *arXiv preprint arXiv:2303.01742*.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2023b. Deepfake text detection in the wild. *arXiv preprint arXiv:2305.13242*.

Weixin Liang, Zachary Izzo, Yaohui Zhang, Haley Lepp, Hancheng Cao, Xuandong Zhao, Lingjiao Chen, Haotian Ye, Sheng Liu, Zhi Huang, Daniel A. McFarland, and James Y. Zou. 2024. Monitoring ai-modified content at scale: A case study on the impact of chatgpt on AI conference peer reviews. *CoRR*, abs/2403.07183.

Jing Liu et al. 2022. Improving text classifier robustness with contrastive learning. *arXiv preprint arXiv:2204.01561*.

Yinhan Liu et al. 2019. Roberta: A robustly optimized bert pretraining approach. In *arXiv preprint arXiv:1907.11692*.

Eric Mitchell et al. 2023. Detectgpt: Zero-shot machine-generated text detection using negative curvature in probability space. In *Proceedings of the 2023 ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Colin Raffel et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. In *Journal of Machine Learning Research*.

A. Singh and J. Zou. 2023. Evaluating watermarking in language models: Impact on quality and coherence. *Nature Machine Intelligence*, 5(2):120–130.

Irene Solaiman et al. 2019. Log-probability based classification of ai-generated text. In *Proceedings of the NeurIPS Workshop on AI for Social Good*.

Mercan Topkara, Cuneyt Taskiran, and Edward Delp. 2006. Watermarking natural language text through syntactic transformations. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP 2006)*, volume 5, pages V–V. IEEE.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Leonid Tulchinskii et al. 2023. Intrinsic dimensionality of machine-generated text: A study using persistent homology. In *Proceedings of the 2023 International Conference on Machine Learning*.

Ada Uchendu et al. 2020. Authorship attribution of ai-generated text. In *Proceedings of the 2020 International Conference on Computational Linguistics*.

Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2023. Llmdet: A third party large language models generated text detection tool. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2113–2133.

Kangrui Yu et al. 2023. Gpt-pat: A twin network for detecting ai-generated text via re-decoding similarity. *arXiv preprint arXiv:2302.03205*.

Rowan Zellers et al. 2019. Neural text deception: Generating media articles for fake news detection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675.

A Baseline Comparison

A.0.1 RADAR (Hu et al., 2023b)

The RADAR model has the following mechanism - Initially, an AI-text corpus is generated from a frozen target language model from a human text corpus. Next, it introduces two tunable language models: a paraphraser and a detector. The detector's object in the training stage is to distinguish between human-generated and AI-generated text. In contrast, the paraphraser's goal is to paraphrase the AI-generated text to avoid detection. The parameters of both these models are updated in an adversarial learning manner. During evaluation, the detector utilizes its training to assess the probability of the content being AI-generated for any given input instance. The RADAR model was originally trained using a large-scale generic dataset of English-language AI text (160K documents sampled from WebText).

A.0.2 LLMDET (Wu et al., 2023):

The framework of LLMDET consists of two main components - 1) Dictionary creation and 2) Text detection. The main idea was to use perplexity to identify text generated by different LLMs. The dictionary has *n*-grams as the keys, and their corresponding next-token probabilities are the values. This dictionary functions as prior information during the text detection process. Once the *n*-gram dictionary and its probabilities were set up, it allowed for the use of corresponding dictionaries from various models as background information for detecting text from third parties. This approach made it easier to calculate proxy perplexity for the text being analyzed with each model. Then, this proxy perplexity was incorporated as a feature in a trained text classifier, which was used to generate the detection results.

A.0.3 DEEP-FAKE (Li et al., 2023b)

To determine whether machine-generated text can be discerned from human-written content, data was collected and categorized into six settings based on its sources, and used for model training and evaluation. These settings progressively increase the difficulty of machine-generated text detection. The classifier assigns a probability to each text, indicating the likelihood of it being authored by humans or generated by language models. AvgRec (average recall), the average recall score between the human-written (HumanRec) and machine-generated (MachineRec) texts, was the principal metric.

A.0.4 FAST-DETECT GPT (Bao et al., 2023a)

The model comprises a three-part architecture - 1) It reveals and confirms a novel conjecture that humans and machines show distinct word selection patterns in a given context; 2) It introduces conditional probability curvature as a new feature for identifying machine-generated text, reducing detection costs by two orders of magnitude; 3) It achieves the highest average detection accuracy in both white-box and black-box settings, outperforming current zero-shot text detection systems.

B BERT Score Analysis

AI-Generated Reviews (Higher BERT Score): In these cases, the AI-generated completions tend to be highly similar to the pruned tail, leading to a high BERT score:

- **Example 1:**
  - **T (AI-generated):** *The scalability of VNNs in terms of computational complexity for || high-dimensional datasets, especially considering the practical implications, could be further discussed.*
  - **G (Generated completion):** *The scalability of VNNs in terms of computational complexity for || high-dimensional datasets needs further exploration.*
- **Example 2:**
  - **T (AI-generated):** *The theoretical analysis establishing the stability of VNNs to || perturbations in the sample covariance matrix is thorough and well-supported.*

- **G (Generated completion):** *The theoretical analysis establishing the stability of VNNs to || perturbations in the sample covariance matrix is well-founded.*

As seen in these examples, the AI-generated completions remain highly similar to the original sentence, leading to a high BERT similarity score.

Human-Written Reviews (Lower BERT Score): In contrast, human-written completions exhibit greater variance, making them less similar to the pruned tail, resulting in a lower BERT score:

- **Example 1:**
  - **T (Human-written):** This paper follows the promising trend of task-unification under a transformer framework with sequence || modeling, and the authors extend the Pix2Seq model to learn four specific tasks in COCO datasets.
  - **G (Generated completion):** This paper follows the promising trend of task-unification under a transformer framework with sequence || modeling, which has shown great potential in both NLP and vision tasks.
- **Example 2:**
  - **T (Human-written):** The paper is well written and easy to follow. Especially, the comparison between QAT and PTQ in Section 2.2 provides good motivation for the paper. The experiments are very well organized and support the advantages of the proposed method. Previous works are also sufficiently addressed. Teacher forcing seems to be a good approach to dividing modules and performing separate optimization for each. The || linear annealing schedule is reasonable, and the authors sufficiently support the necessity of the teacher forcing by experiments.
  - **G (Generated completion):** The paper is well written and easy to follow. Especially, the comparison between QAT and PTQ in Section 2.2 provides good motivation for the paper. The experiments are very well organized and support the advantages of the proposed method. Previous works are also sufficiently addressed. Teacher forcing seems to be a good approach to dividing modules and performing separate optimization for each. The || method effectively mitigates the propagation of reconstruction errors across modules.

As observed in these examples, human-written completions introduce more variation in word choice and structure, leading to lower BERT similarity scores compared to AI-generated completions.

C Error Analysis

Error Categories

- **Formality and Abstraction:**
  - **Instance 1:**
    - **True Sentence:** *and more diverse experiments with different levels of exploration should be conducted.*
    - **Generated Sentence:** *which may not fully capture the potential of the broader range of strategies.*
    - **True Label:** 0 (AI-generated)
    - **Model Prediction:** 1 (Predicted as Human-written)
    - **Error Cause:** The generated sentence introduces a level of abstraction and generalization. The model incorrectly predicted it as human-written, likely due to the use of formal language, which can occur in both human and AI-generated texts.
  - **Instance 2:**
    - **True Sentence:** *as critic, actor, and exploration, on transfer learning.*
    - **Generated Sentence:** *this thorough investigation reveals the critical roles of actors and critics in transfer learning.*
    - **True Label:** 0 (AI-generated)
    - **Model Prediction:** 1 (Predicted as Human-written)
    - **Error Cause:** The model was misled by formal and detailed phrasing, such as "thorough investigation," which is often found in academic writing. However, this formality is not exclusive to human-written text, leading to the incorrect classification.

- **Conciseness:**

  - **Instance 3:**

    * **True Sentence:** *to be more detailed. for example, when it is sufficient to...*

    * **Generated Sentence:** *additionally, the paper could benefit from a more detailed explanation of the examples provided.*

    * **True Label:** 1 (Human-written)

    * **Model Prediction:** 0 (Predicted as AI-generated)

    * **Error Cause:** The generated sentence is concise and formal, resembling AI-generated text. However, it was actually human-written, and the model misclassified it as AI-generated due to the simple structure and direct language.

D Prompt for LLM completion

To determine the completion prompt, we used over 100 tail-pruned reviews along with their corresponding golden completion reviews. Our goal was to ensure that the tail-pruned review, after prompting, closely resembled the golden completion. However, we observed that in some cases, the completion introduced additional information or altered the original intent of the review. We use the below prompt for our experiments:-

You are a reviewer for a research paper.
Your task is to complete the review of the paper from the `<completion>` tag after analyzing the research paper provided to you.
You will do this in the following steps:
1. Read the research paper provided to you.
2. Read the review point provided to you.
3. Complete the review point based on the research paper.

The research paper and review point are delimited by triple backticks (“‘) for your reference.
Paper:
`{paper_content}`

Review:
`{review_content}`

Return the output in the following format:
{
`"review": [sentence1, sentence2, sentence3, ...]`
}
Each `sentence_i` in itself will be a list of the previous sentences and generated sentences.

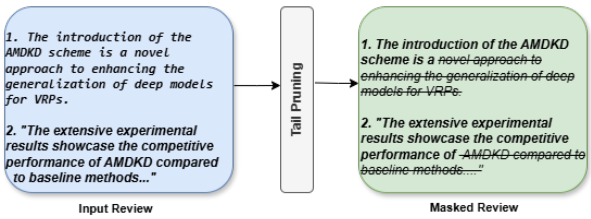

Figure 3: Example of tail pruning

E Dataset Details

We collect 1,000 papers and their corresponding peer reviews from the NeurIPS 2022 conference via the OpenReview platform. We ensure that the reviews are written before the widespread availability of advanced language models like ChatGPT, which was released in November 2022, to minimize the likelihood of any reviews being influenced by AI-generated content. We obtain peer reviews provided by human reviewers to form our human-written review dataset. We also use the same set of papers and a language model to generate reviews

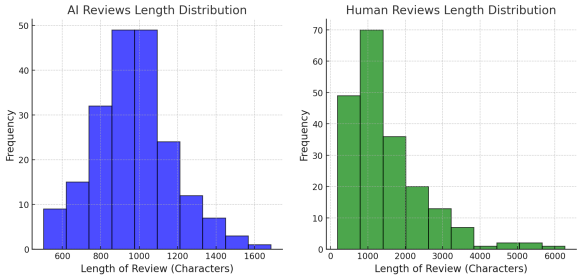

Figure 4: The left side shows the length distribution of AI-generated reviews, while the right side shows that of human-written reviews. The lengths are measured in the number of characters.

for them, creating the AI-generated review dataset. Both human and AI-generated reviews are based on the same content, allowing for a direct comparison. The complete dataset, combining human-written and AI-generated reviews, is split into training, validation, and test sets with proportions of 70%, 10%, and 20%, respectively.