Detecting Mix-Case Human Language Model with Zero-Shot Approach

Soniya Chaudhary[1], Anshul Rana[1], Anshil Choudhary[1], Ashuti Mittal[1]

[1]Department of Mathematics and Scientific Computing, National Institute of Technology Hamirpur, Hamirpur (HP), 177005, India

Abstract:

In recent years, AI has taken the lead in technological advancement, and with it, LLM has seen big growth and popularity in its use in society. LLM text generation covers all available fields, from story writing to blog posts to other content generation. With such a fast spread of LLM, it is necessary to detect its presence and distinguish it from human-generated content. While there are some detectors that show remarkable performance in detecting pure LLM-generated text or specific LLM model detectors, there still is no detector that can detect paraphrased LLM-generated text or AI-human mix-case text. Our research tested a statistical zero-shot detection method, Binoculars, which utilizes perplexity scores to detect AI-human mix-case against a benchmark detector, RADAR. Our research shows that Binoculars could not outperform RADAR and, in many cases, detection using RADAR was also inefficient. There is a need to design efficient detectors using three-class classifiers and prominent laws to govern the use of AI. The current study holds promise in enhancing the capacity to identify the presence of Language Model Models (LLMs) across different domains. This advancement aids in safeguarding the authenticity of human-generated content, thwarting cheating in examinations, and identifying AI bots disseminating false narratives or misinformation.

1. Introduction:

Recent technological improvements for detecting LLM human mix cases have made significant strides. One example is OpenAI's ChatGPT and Google Bart, which are fantastic at answering questions, writing emails and articles, and coding. However, as these programs get better at generating, people worry about using them in intellectual, wrong, and deteriorating ways, like blackmailing others with fake emails and messages and spreading fake talks (Sison et al.,2023). Some institutes and colleges have even stopped students from using ChatGPT because they are concerned that it might be used to cheat on assignments and fraud on social networking sites. These concerns and causes have made it difficult to use this technology in significant areas like education and the media (Bail et al.,2023). To ensure these tools and programs are used correctly, it is significant to be able to tell when a text or message was generated and implemented by a computer program like ChatGPT or Google Bart. Being able to highlight computer-generated text is versatile for making the best out of this technology while also stopping any wrong stuff from happening. It would help people believe in these systems more and avoid using them in ways they should not be and will not use them in wrong ways. That is why there has been much interest from both academics and companies in researching how to detect human mix cases and figuring out how it all works and can be used in the future (Crothers et al.,2023). There is advanced communication on whether we can recognize if a computer has written something and how we can evaluate that. We will illustrate how Men and women have sought to work this out.

First, there are two main paths (Tang et al.,2024): black-box and white-box detection. Black-box detection can only enter the computer program, like an API, through its combination. It works by togetherness examples of text written by both generations and computers, then using those instances to instruct a computer program to distinguish between them. This method has been implemented beautifully because computer-generated text often has assured patterns that give it away. However, as computer programs get better at writing, black-box methods might not operate as well.

On the other hand, white-box detection has an entire passage to the internal workings of the computer program. It can oversee how the program writes, which helps with trailing where the text came from. Usually, the people who make computer programs are the ones who make this kind of discovery. This essay talks about this significant issue using views from data mining and natural language processing. First, it recognizes black-box detection methods such as collecting data, choosing which characteristics to look at and formatting a program to show the dissimilarity between human and computer writing. Then, it observes newer methods for white-box detection, like putting marks in the text after it is written or eyeing how the program writes in actual time.

Finally, the essay talks about the complications with ongoing detection methods and suggests notions for future implementation. The main aim is to summarize these vital computer programs best by elaborating the basics, displaying how to discover computer-written text, and giving ideas of how it has been done.

2. Related Works

AI text detection can be classified into three approaches:

Watermark methods: A watermark is carefully embedded into the general texts, which can be verified while preserving the quality of the text. Different watermarking techniques such as rule-based methods (Kankanhalli et al.,2002; Brassil et al.,1995), deep-learning-based methods (Ueoka et al.,2021; Dai et al.,2022), or post-hoc watermarking can be applied to LLMs. (Brassil et al.,1995) created a line-shift watermark, which involves moving a text left or right based on the watermark. However, in paraphrased texts, this AI detection method does not perform well (Sadasivan et al.,2023).

Statistical Methods: Building on the fact that LLM-generated texts follow some statistical patterns or contain some peak statistical values such as entropy (Lavergne et al.,2008) or probability. DetectGPT (Mitchell et al.,2023), which exploits the negative log probability, has set a trend in metric or statistical detection methods (Su et al.,2023; Bao et al.,2023). While others like GLTR (Gehrmann et al.,2019) exploit entropy and probability rank, DNA-GPT (Yang et al.,2023) leverages N-gram for AI detection.

Classification Methods: AI-text detection mainly employs binary classification, but recent advancement in the generation of paraphrased AI text calls for a three-class classifier: human-generated text, AI-generated text, and AI-human paraphrased text or mixed text (Mitchell et al.,2023; Ippolito et al.,2023). Certain binary classifiers are trained for specific language models (Solaiman et al.,2019; Rodriguez et al.,2022). Studies have shown that using pre-trained models to extract semantic textual features followed by SVM for classification can outperform statistical methods (Crothers et al.,2022).

Datasets for AI detection:

Various datasets of MGT are proposed alongside their detectors (Verma et al.,2023; Chen et al., 2023). Some use Question-Answer datasets, allowing LLMs to generate answers (Jin et al.,2019). Other domains of datasets include web scrapping, such as Wikipedia, Reddit comments, news generation, and story creation (Guo et al.,2023). (Mitchell et al., 2023; Su et al., 2023) has also proposed datasets containing MGT from various LLMs like OpenAI.

3. Evaluation of detectors:

Although many datasets and detectors are present, a specific question still arises about evaluating such detectors that can verify their effectiveness. Many of these detectors are evaluated on accessible and reflective datasets (Liang et al.,2023), and some of these only focus on accuracy on balanced test sets or AUC. It has been found that detectors with low false favorable rates across the wide distribution of human written text work well (Hans et al.,2024).

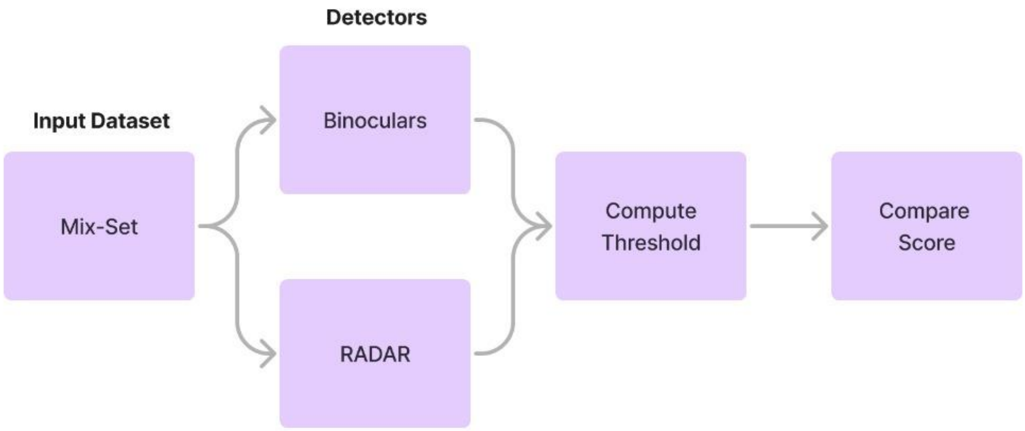


Figure: Workflow of the present study

4. Experiment:

Our experiment is to test the effectiveness of the zero-shot detection model Binoculars (Hans et al.,2024), which uses the ratio of perplexity to cross-perplexity, and RADAR (Hu et al.,2023), which consists of three neural-network-based language models against a mix-set dataset (Gao et al.,2024).

Mix-Set Dataset:

This dataset consists of 3.6k mix-case instances involving AI revised Human-generated text and human-revised AI-generated text.

Five operations were used to generate these mix-cases, which can be further divided into AI Revised, which includes ‘Polish,’ ‘Complete,’ and ‘Rewrite,’ and Human Revised, which includes: ‘Humanize’ and ‘Adapt.’

Polish (Chen,2023; Gao et al.,2024): It contains token and sentence level polishing. Token level makes alterations at individual word level, while sentence level aims to enhance the overall coherence and clarity of the text by revising and restructuring the complete sentence.

Complete (Zhuohan Xie; Gao et al.,2024): It involves taking 1/3 of every text and employing llm to generate the rest of text.

Rewrite (Shu et al. ,2023; Gao et al.,2024): It requires LLMs to initially comprehend and extract key information from the given HWT and then rewrite them.

Humanize (Bhudghar,2023; Gao et al.,2024): It refers to modification of MGT to mimic the natural noise more closely for LM (Wang at al.,2021) that human writing always brings. LLMs were employed to introduce various perturbations to the pure MGT, including typo, grammatical mistakes, links, and tags.

Adapt (Gero et al.,2022; Gao et al.,2024): Adapt operation refers to modifying MGT to ensure its alignment to fluency and naturalness to human linguistic habits without introducing any error expression. This operation is also divided into token and sentence level adaptation.

5. AI Detectors:

Binoculars (Hans et al.,2024): It proposes ratio of two scores, where one is perplexity measurement and other is cross-perplexity.

Binoculars score: $B_{M1,\,M2}(s) = \frac{log\ PPL_{M_1}(s)}{\log X{-}PPL_{M_1,M_2}(s)}$

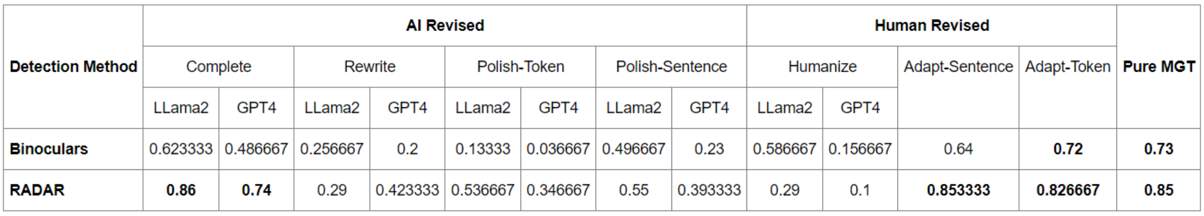
Log $PPL_{M1}(s)$ is perplexity which measures how surprising a string is to M1(language model).

Log X-$PPL_{M1,\,M2}(s)$ is cross perplexity, which measures how surprising the token predictions of M2 are when observed by M1.

RADAR (Hu et al.,2023): It contains a framework of three neural-network based language models (LMs): the target LM T, the detector D, and the paraphraser G. For a given target LLM, RADAR returns a trained paraphraser and a trained detector. In evaluation phase the detector is used to predict he likelihood of AI-text for any input instance. In our experiment RADAR is being used as a benchmark as it was the only one which shows good results in this experiment (Gao et al.,2024).

One important point to note is that both the detectors are not trained on the dataset tested.

Results:

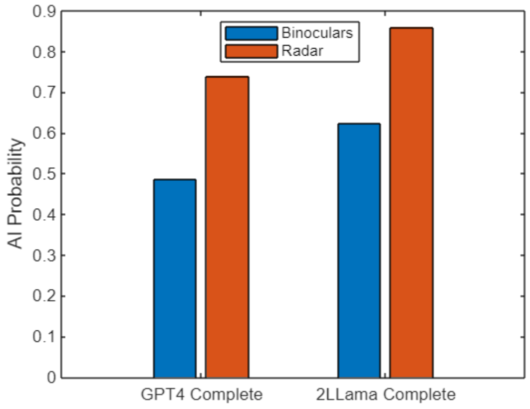


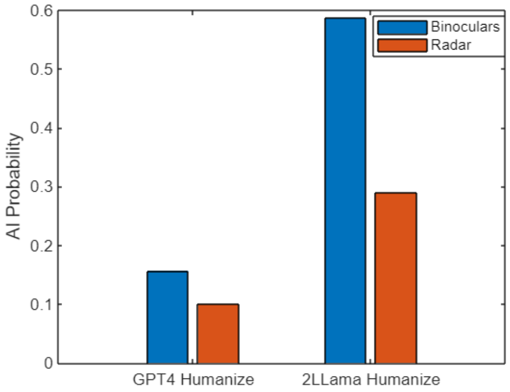
Detection Method	**AI Revised**								**Human Revised**				**Pure MGT**
	Complete		Rewrite		Polish-Token		Polish-Sentence		Humanize		Adapt-Sentence	Adapt-Token	
	LLama2	GPT4	LLama2	GPT4	LLama2	GPT4	LLama2	GPT4	LLama2	GPT4			
Binoculars	0.623333	0.486667	0.256667	0.2	0.13333	0.036667	0.496667	0.23	0.586667	0.156667	0.64	**0.72**	**0.73**
RADAR	**0.86**	**0.74**	0.29	0.423333	0.536667	0.346667	0.55	0.393333	0.29	0.1	**0.853333**	**0.826667**	**0.85**

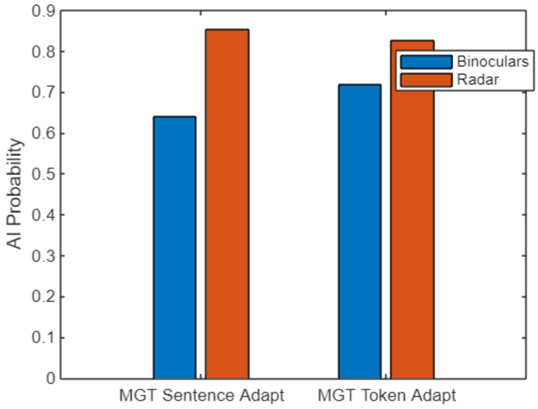
Fig: Normalized AI detection score

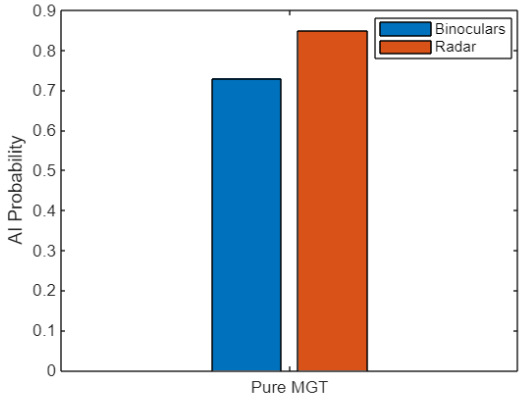
We have tested both the detectors against all twelve subsets of the mixed case dataset and one pure MGT. In most cases, RADAR outperformed binoculars, except in the humanized function. RADAR has only performed in only half of the test cases and has not even crossed a score of 0.5 in the rest of the cases. In the case of Pure MGT and Adapt operator, both the detectors achieved a good enough score, and in the case of the complete operator, RADAR achieved a good score.

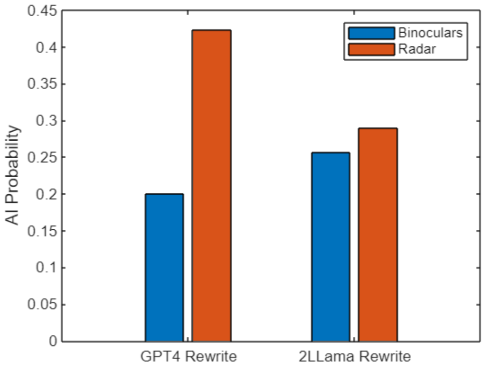
Bar plots comparing individual operator dataset on both detectors:

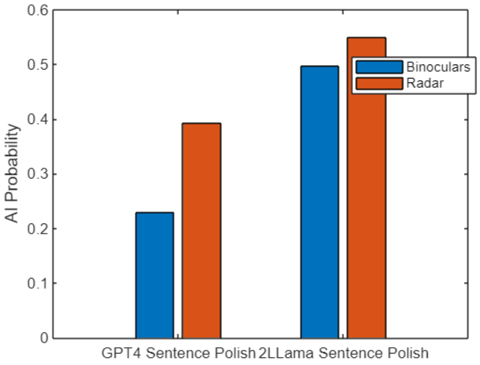

AI Probability
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Binoculars
Radar
GPT4 Complete
2LLama Complete

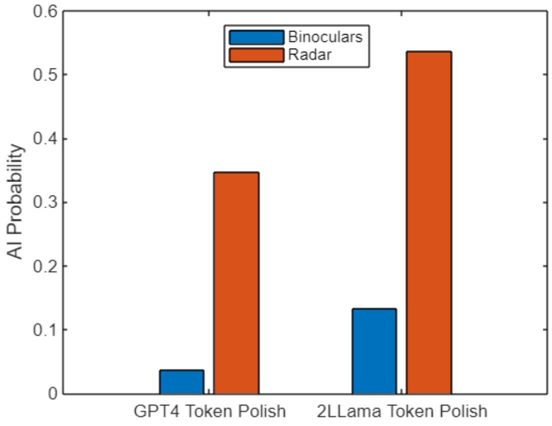

AI Probability
0
0.1
0.2
0.3
0.4
0.5
0.6
Binoculars
Radar
GPT4 Humanize
2LLama Humanize


AI Probability
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Binoculars
Radar
MGT Sentence Adapt
MGT Token Adapt


AI Probability
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Binoculars
Radar
Pure MGT


AI Probability
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
Binoculars
Radar
GPT4 Rewrite
2LLama Rewrite


AI Probability
0
0.1
0.2
0.3
0.4
0.5
0.6
Binoculars
Radar
GPT4 Sentence Polish
2LLama Sentence Polish


AI Probability
0
0.1
0.2
0.3
0.4
0.5
0.6
Binoculars
Radar
GPT4 Token Polish
2LLama Token Polish

6. Conclusion:

As we have tested the detection of AI-human mix-case using two detectors, both untrained, we found that we need better detection algorithms to detect mix-case data. Binoculars that use perplexity scores did not perform in the mix-case datasets and could only detect pure MGT. This shows us that using perplexity as a detection measure may not be helpful in such cases. LLM-generating models have been advanced to more heights to replicate human writing styles, making it difficult to detect its presence using a perplexity score. There are some limitations in this study, as the study is quite limited, leaving out the comparison with other detectors due to the use of paid APIs. The study classified mix-case as AI and tested it against binary classifiers; we recommend testing it using a three-class classifier where mix-case can be classified separately. There is a need to formulate laws supervising the use of AI in various fields, helping humans preserve their work's authenticity. Trained LLMs can be biased upon the data they had been trained on, so they can give biased outputs on specific topics, which can affect some people in societies. Therefore, it is necessary to have control over the use of LLMs. The findings of this study will prove valuable for future endeavors aimed at enhancing detection capabilities. Areas for improvement include integrating a three-class classifier, training detectors with extensive data from diverse sources and languages, and devising a generalized approach for detecting the utilization of various language model models rather than focusing on specific ones.

References:

1. Sison, A.J.G., Daza, M.T., Gozalo-Brizuela, R. and Garrido-Merchán, E.C., 2023. ChatGPT: More than a “weapon of mass deception” ethical challenges and responses from the human-centered artificial intelligence (HCAI) perspective. *International Journal of Human–Computer Interaction*, pp.1-20.
2. Bail, C., Pinheiro, L. and Royer, J., 2023. Difficulty Of Detecting AI Content Poses Legal Challenges. *Law360, April*.
3. Crothers, E., Japkowicz, N. and Viktor, H.L., 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*.
4. Tang, R., Chuang, Y.N. and Hu, X., 2024. The Science of Detecting LLM-Generated Text. *Communications of the ACM*, *67*(4), pp.50-59.
5. Kankanhalli, M.S. and Hau, K.F., 2002. Watermarking of electronic text documents. *Electronic Commerce Research*, *2*, pp.169-187.
6. Brassil, J.T., Low, S., Maxemchuk, N.F. and O'Gorman, L., 1995. Electronic marking and identification techniques to discourage document copying. *IEEE Journal on Selected Areas in Communications*, *13*(8), pp.1495-1504.
7. Ueoka, H., Murawaki, Y. and Kurohashi, S., 2021. Frustratingly easy edit-based linguistic steganography with a masked language model. *arXiv preprint arXiv:2104.09833*.
8. Dai, L., Mao, J., Fan, X. and Zhou, X., 2022. Deephider: A multi-module and invisibility watermarking scheme for language model. *arXiv preprint arXiv:2208.04676*, pp.1-16.
9. Sadasivan, V.S., Kumar, A., Balasubramanian, S., Wang, W. and Feizi, S., 2023. Can AI-generated text be reliably detected?. *arXiv preprint arXiv:2303.11156*.

10. Lavergne, T., Urvoy, T. and Yvon, F., 2008. Detecting Fake Content with Relative Entropy Scoring. *Pan*, *8*(27-31), p.4.
11. Mitchell, E., Lee, Y., Khazatsky, A., Manning, C.D. and Finn, C., 2023, July. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning* (pp. 24950-24962). PMLR.
12. Su, J., Zhuo, T.Y., Wang, D. and Nakov, P., 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *arXiv preprint arXiv:2306.05540*.
13. Bao, G., Zhao, Y., Teng, Z., Yang, L. and Zhang, Y., 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*.
14. Gehrmann, S., Strobelt, H. and Rush, A.M., 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.
15. Yang, X., Cheng, W., Petzold, L., Wang, W.Y. and Chen, H., 2023. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. *arXiv preprint arXiv:2305.17359*.
16. Ippolito, D., Duckworth, D., Callison-Burch, C. and Eck, D., 2019. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*.
17. Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., Radford, A., Krueger, G., Kim, J.W., Kreps, S. and McCain, M., 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.
18. Rodriguez, J.D., Hay, T., Gros, D., Shamsi, Z. and Srinivasan, R., 2022, July. Cross-domain detection of GPT-2-generated technical text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1213-1233).
19. Crothers, E., Japkowicz, N., Viktor, H. and Branco, P., 2022, July. Adversarial robustness of neural-statistical features in detection of generative transformers. In *2022 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.
20. Verma, V., Fleisig, E., Tomlin, N. and Klein, D., 2023. Ghostbuster: Detecting text ghostwritten by large language models. *arXiv preprint arXiv:2305.15047*.
21. Chen, Y., Kang, H., Zhai, V., Li, L., Singh, R. and Ramakrishnan, B., 2023. Gpt-sentinel: Distinguishing human and chatgpt generated content. *arXiv preprint arXiv:2305.07969*.
22. Jin, Q., Dhingra, B., Liu, Z., Cohen, W.W. and Lu, X., 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.
23. Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., Yue, J. and Wu, Y., 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.
24. Liang, W., Yuksekgonul, M., Mao, Y., Wu, E. and Zou, J., 2023. GPT detectors are biased against non-native English writers. *Patterns*, *4*(7).
25. Hans, A., Schwarzschild, A., Cherepanova, V., Kazemi, H., Saha, A., Goldblum, M., Geiping, J. and Goldstein, T., 2024. Spotting LLMs With Binoculars: Zero-Shot Detection of Machine-Generated Text. *arXiv preprint arXiv:2401.12070*.

26. Hu, X., Chen, P.Y. and Ho, T.Y., 2023. Radar: Robust ai-text detection via adversarial learning. *Advances in Neural Information Processing Systems, 36*, pp.15077-15095.
27. Gao, C., Chen, D., Zhang, Q., Huang, Y., Wan, Y. and Sun, L., 2024. Llm-as-a-coauthor: The challenges of detecting llm-human mixcase. *arXiv preprint arXiv:2401.05952*.
28. Chen. 2023. Gpt academic prompt. https://github.com/xuhangc/ ChatGPT-Academic-Prompt.
29. Xie, Z., Cohn, T. and Lau, J.H., 2023. The next chapter: A study of large language models in storytelling. *arXiv preprint arXiv:2301.09790*.
30. Shu, L., Luo, L., Hoskere, J., Zhu, Y., Liu, Y., Tong, S., Chen, J. and Meng, L., 2024, March. Rewritelm: An instruction-tuned large language model for text rewriting. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 38, No. 17, pp. 18970-18980).
31. Bhudghar. 2023. Ai text converter. https://aitextconverter.com/.
32. Wang, B., Xu, C., Wang, S., Gan, Z., Cheng, Y., Gao, J., Awadallah, A.H. and Li, B., 2021. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. *arXiv preprint arXiv:2111.02840*.
33. Gero, K.I., Liu, V. and Chilton, L., 2022, June. Sparks: Inspiration for science writing using language models. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference* (pp. 1002-1019).