Author Detection Final Report

Ivan Micanovic
University of Wisconsin–Madison
Madison, WI 53703
imicanovic@wisc.edu

Ayush Mahapatra
University of Wisconsin - Madison
Madison, WI, 53703
mahapatra3@wisc.edu

Leo Xu
University of Wisconsin–Madison
Madison, WI 53703
lpxu@wisc.edu

Abstract

The mass adoption of LLMs in academic settings has created many challenges for maintaining academic integrity, particularly in a student setting, where it becomes easy to generate high-quality essays in seconds. Although AI detection systems exist, they typically require massive datasets to train over and can be easily fooled with basic editing techniques. We introduce a novel method for improving AI detection systems in low-data environments using summarization-based, emotion-based, and style-transfer augmentations. We build on the current SOTA for AI content detection (Ghostbuster) and adapt it for scenarios where it is only allowed to learn on tens or hundreds of samples rather than thousands. Currently, we have rewritten the deprecated portions of the original Ghostbuster repository to add support for newer models from OpenAI. We are able to replicate the original results from the Ghostbuster paper, achieving a 99.0 F1 score on an evaluation data set with training on the whole data set. We plan to conduct training on varying amounts of training data availability: 25, 50, 100, 200, and 500 samples. We will run these tests on our baseline models and also on our augmentation methods.

1 Introduction

With the rise of Large Language Models, like OpenAI's ChatGPT LLMs, there has a been a significant rise in the amount of academy integration students' work, such as homework, essays, and online exams. These LLMs have allowed students the ability to have another resource write whole essays, find and create solutions on homework, and devise answers for exams without any thought or input from the student itself. While these LLMs may not take over human creativity, these LLMs are widely undermining the education principles of students creating their own original work and challenging themselves to consistently work and exercise their brains to come up with ideas and bring them into fruition, whether it be homework, essays, or even exams (Mahama et al., 2023)

In this paper, we are specifically focusing on the use of LLMs within the domain of student essays. While there has been an effort to develop AI detection methods to identify AI generated Content (AIGC), most of these methods have been shown to be easily avoided by using different methods, such as word substitution and sentence substitution (Peng et al., 2024), and there is a lack of data to evaluate them as well. There are many times when students will use an LLM and try to edit the essay just enough so that it will not be detected as AIGC. As a result, there is a need for more robust and precise AI detection methods to be developed in order to upheld Academic Integrity across schools all over the world.

With limited access to data, we look into potential data augmentation that can be applied to the original training data used for Ghostbusters to create more data to improve Ghostbuster's ability to detect AI-generated essays and uphold academic integrity and creativity.

2 Related Works

Multiple works have attempted to address the issue of detecting AI–generated text. One very promising line of work involves watermarking the text that come out of generative AI systems, providing more confidence in identifying AIGC (Kirchenbauer et al., 2024). However, since watermarking requires the developer to encode a signal into the generated text, solely relying on watermarks is not a generalizable.

Other works (Zellers et al., 2020) have attempted to detect AIGC by training models to classify text as either human or AI–generated, but (Mitchell et al., 2023) (Verma et al., 2024) have shown that such approaches are prone to overfitting to training data.

As a result, other works like GLTR (Gehrmann et al., 2019) try to detect AIGC through purely statistical methods, where the authors visualize the probability of each token being generated by a language model and overlay it on the text. They show that doing so improves the human detection rate of AIGC from 54% to 72%. DetectGPT (Mitchell et al., 2023) adopts a similar approach, where the authors rely on the insight that generated text from LLMs lie in regions of the probability space where nearby samples are given lower probabilities. The authors use this insight by examining the probabilities of randomly perturbed text to determine the likelihood of a given text sample coming from a language model.

However, approaches like GLTR (Gehrmann et al., 2019) and DetectGPT (Mitchell et al., 2023) require access to a target model's probability distributions, and have been shown to be vulnerable to various paraphrasing attacks. Since students mostly use large closed-source models like OpenAI's ChatGPT, Anthropic's Claude, and Google's Gemini, due to their accessibility, approaches like DetectGPT and GLTR are not feasible for a classroom setting.

To our knowledge, Ghostbuster (Verma et al., 2024) is the current state-of-the-art for detecting AIGC as it has been shown to be robust to different language models and avoid the overfitting issue (Mitchell et al., 2023) seen by fully supervised AI text detectors. Ghostbuster instead leverages the probability distributions of the input text from various small, publically-available models and trains a logistic regression model on different combinations of these probability distributions as feature vectors. Their work was shown to outperform GPTZero and DetectGPT by over 6 and 40 points in F1 score, respectively.

However, Ghostbuster was trained on over 3,000 human written texts and over 29,000 AI-generated texts. For a smaller classroom setting, teachers may only have access to hundreds of samples to create a custom AI-text detection system for the class. In our paper, we explore the efficacy of leveraging high-quality data augmentations to create robust AI-text detection systems in scarce data environments.

While there have been many works (Feng et al., 2021) (Chen et al., 2023) (Ding et al. 2024) on text augmentations in NLP ranging from statistical or rule-based methods (back translation, synonym swapping, stop word removal, etc.) to model-augmentation strategies (synthetic, rewriting, paraphrasing), we introduce three novel data augmentation techniques specifically for detecting AI text that leverage domain specific information for our task. To our knowledge, there have been no works that have explicitly explored the efficacy of style transfer, emotion augmentation, and summary-based rewrite augmentations.

3 Approach

3.1 Initial Symbolic Data Generation

When using the Ghostbuster model, we encountered several problems. The first thing we needed to fix was Ghostbusters using deprecated OpenAI models 'ada' and 'davinci.' To fix this issue, we manually regenerated log probabilities for 'ada' and 'davinci' by replacing them with log probabilities from OpenAI's 'baggage-002' and 'davinci-002' models. This cost us around $10.

After fixing these initial errors, we needed to do a structured search over the probability vectors returned from a unigram model, trigram model, babbage-002, and davinci-002. This step took about 5 hours as we needed to do this structured search for our training data and evaluation data separately. During training, we then greedily select the best features by iteratively running k-folds and selecting features that improve our best F1. At each iteration of k-folds, we select the symbolic data that improves our running F1 score, and then remove the selected feature from our set of symbolic data before running another iteration of k-folds. If the k-fold run doesn't improve our F1 score, we exit and return the set of features we greedily selected. In our runs, the average number of features tended to be around 8 for both evaluation and training data.

For each of our augmentations detailed below, we needed to run this process of generating symbolic data for each of the augmented probability vectors from our unigram, trigram, babbage-002, and davinci-002 models.

3.2 Baseline Model

After ironing out the issues above, we were able to successfully run the Ghostbuster authors' training code and replicate the results that they reported with the new "up-to-date" models. Specifically, we achieved a 99.0 F1 score on the evaluation set while training. It remains for us to simply run this script over various training size lengths: 10, 25, 50, 75, 100 samples.

3.3 Dataset Curation for Existing Augmentations

In our research, data set augmentation is vital as it enriches the dataset by introducing natural variations, improving model generalization and improving robustness. Specifically, we focus on two widely used augmentation methods: synonym swapping and back translation. These methods serve as baseline enhancement techniques to compare with our custom enhancement strategies, ensuring a robust evaluation framework.

3.3.1 Synonym Swapping

Synonym swapping introduces lexical diversity by replacing words in the dataset with their synonyms, while maintaining the original sentence structure and meaning. This method is particularly effective for tasks that require robust semantic understanding, as it ensures minimal distortion of the underlying data distribution.

Process
Using the WordNet-based augmentation module from the nlpaug library, we implemented synonym swapping with the following approach:

1. **Setup:** The augmentation script initializes the naw.SynonymAug augmenting, which leverages WordNet to identify and replace words with appropriate synonyms. We configured the augmenter with an augmentation probability (`aug_p=0.8`) and a minimum number of words to augment (`aug_min=125`) to ensure noticeable variations across the dataset.
2. **Text Processing:** Each file's content is tokenized, augmented using synonym replacement, and then written back as a single string to the new file. The augmented outputs retain the structure of the input but exhibit diverse lexical alternatives.

3.3.2 Back Translation

Back translation is a powerful data augmentation technique that introduces natural linguistic variations by translating text into another language and then back into the original language. This method preserves the semantic meaning while generating syntactic and lexical diversity. It is particularly effective for tasks like text classification, where nuanced differences in phrasing improve model robustness. In our research, back translation serves as another baseline to compare against custom augmentations, helping to evaluate their efficacy in enhancing dataset quality.

Process
To implement back translation, we used the googletrans library with the following steps:

1. **Language Pair Selection:** We translated the text from English to German and then back to English. This pair was chosen due to the significant syntactic differences between the two languages, which increase the diversity of augmented text while maintaining fidelity to the original meaning.
2. **Translational Functionality:** The translation was performed using a custom function that handles both forward and backward translations. The process includes:
   - Translating the content from English (`src_lang = 'en'`) to German (`dest_lang = 'de'`).
   - Translating it back to English (`dest_lang= 'en'`), yielding back the translated text.
3. **Error Handling:** Rate limits and transient errors from the Google Translate API were managed with retries and timeouts, ensuring a reliable process for large datasets. If translation failed after retries, the original text was saved instead, maintaining the dataset's completeness.

Challenges: A significant challenge in implementing back translation was avoiding the high costs typically associated with commercial translation APIs, as back translation is often computationally expensive. To address this, we used the free, unofficial Google Translate API provided by the googletrans library. While this approach avoided financial costs, it introduced challenges such as rate limits and network timeouts. To mitigate these issues, we implemented a retry mechanism with delays (time.sleep(10)) to handle API failures gracefully, ensuring all files were processed without manual intervention. This allowed us to scale the augmentation process effectively without incurring additional expenses.

3.4 Dataset Curation for Novel Augmentations

The next three augmentations that we introduce are all generated using gpt-3.5-turbo, offered by OpenAI API. We set max tokens to 1500 and found that a temperature of 0.9 suffices for our use.

3.4.1 Summary Augmentation

The first of our own augmentation strategies involves creating a synthesized essay by summarizing the original student essay. The process begins with an automatic summarization of the essay using a pre-trained model, followed by asking gpt-3.5-turbo to generate a new essay that maintains the meaning but introduces different sentence structure, vocabulary, and flow. The idea behind this is to create diverse versions of the same content written by an AI model, allowing for variations of expression that might be seen in these models. This augmentation will provide examples of an AI written text.

Process
The augmentation relies on two carefully crafted GPT prompts to ensure consistency and creativity:

1. **Summarization Prompt:** The model is instructed to extract the key ideas of the input text and summarize them in 8–10 concise points. To aid in subsequent processing, the output also includes the word count of the original text, explicitly labeled to ensure proper utilization. An example prompt to gpt-3.5-turbo would be: "You are an AI assistant that is good at summarizing texts into a list of points. You also grab the word counts of a text”
2. **Rewriting Prompt:** The summary points, along with the original text's word count, are then passed to the model to generate a rewritten passage. The model is explicitly instructed to produce an essay matching the word count provided, ensuring consistency in data length. An example prompt to gpt-3.5-turbo would be: “Summarize the following text in 8-10 concise points. Make sure the points encompass what the passage is trying to say. Along with the summary, include the word count in your output so that it can be used later. Make sure to clearly label that it is in fact the word count, so that another model would be able to identify that clearly.”

3.4.2 Stylistic Augmentation

Stylistic augmentation enhances the diversity of the dataset by transforming the writing style of the text. Two distinct stylistic transformations were applied based on the data source:

- Human Data: Rewritten to mimic the style of a foreign exchange student with limited English proficiency.
- GPT Data: Rewritten to emulate a student asking GPT to rewrite their essay while maintaining the original meaning.

Stylistic augmentation effectively introduces linguistic diversity by transforming text into distinct styles representative of real-world language use. The two targeted approaches ensure that human and GPT data augmentations reflect different yet plausible variations, enriching the dataset for downstream tasks involving nuanced stylistic or linguistic analysis.

Process

1. **Human Data Augmentation:** For human-labeled data, the model was prompted to rewrite the text in the style of a foreign exchange student. This transformation focused on using simpler vocabulary and sentence structures, along with introducing subtle grammatical mistakes typical of non-native English speakers. An example prompt to the model would be: "Rewrite the following passage in the style of a foreign exchange student with limited experience in the English language. Use simpler vocabulary, simpler sentence structures, and subtle grammatical mistakes typical of non-native speakers. Keep the new passage around the same length as the original passage. Also return the new passage in the same format."
2. **GPT Data Augmentation:** For GPT-labeled data, the model was instructed to rewrite the text with stylistic variations. The rewritten text maintained the original tone and meaning while varying sentence structure and length to produce a distinct but comparable passage. The example prompt to the model would be: "Rewrite the following passage with stylistic variations. Vary the sentence structure and length, use different stylistic choices to maintain the tone, and make the rewrite distinct from the original while keeping the content and meaning intact. Ensure the rewritten passage is approximately the same length as the original.

3.4.3 Emotional Augmentation

The third augmentation strategy simulates variations in a student's writing style under different emotional tones. Using emotion classifiers (positive, negative, or neutral) we detect the dominant emotional tone of the original essay and then apply transformations that shift the tone to different a different emotional state while preserving the student's stylistic writing traits. This approach creates a more realistic variability by accounting for the influence of emotional context on writing style, further boosting the training data and allowing the model to better generalize across the different emotional states. This augmentation will provide examples of a human written text.

Process
The augmentation process uses a single prompt that combines emotion detection and transformation. The prompt ensures the rewritten passage reflects the new tone without altering the core content or overall length.

1. **Emotional Identification and Rewriting:** The model first identifies the emotional tone of the input text (positive, negative, or neutral). It then rewrites the text, replacing the identified tone with one of the other two tones. An example prompt to the model is: "Identify the emotion of the following text (positive, negative, or neutral). Then rewrite the text with a new emotional tone, selecting one of the other two options (e.g., if the text is positive, rewrite it as negative or neutral). Ensure the new version maintains the original meaning as much as possible while reflecting the chosen tone. Keep the new passage similar in length to the original passage. "

4 Experimental Overview

Our experiments aimed to evaluate the effectiveness of various data augmentation techniques in enhancing the performance of the Ghostbusters model. The evaluations focused on understanding how augmentations influence the model's ability to generalize across both in-domain and out-of-domain datasets, as measured by F1-scores.

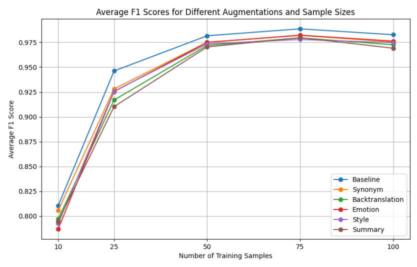
4.1 Experiment Setup

1. **Baselines:**
    - **Ghostbusters with Existing Data Only:** The model was trained solely on the unaugmented dataset to establish a baseline.
    - **Ghostbusters with Back Translation:** A standard augmentation technique was applied to the dataset, translating text to another language and back to English.
    - **Ghostbusters with Synonym Swapping:** Words in the dataset were replaced with their synonyms using a WordNet-based approach.
2. **Augmented Data Configurations**: Models were trained with the original dataset augmented by varying proportions of the following augmentations:
    - **Ghostbusters with Summary-Based Augmentations:** Training sets included varying proportions of summary-augmented data alongside the original dataset.
    - **Ghostbusters with Stylistic-Based Augmentations:** Training sets incorporated differing percentages of data with adjusted emotional tones, in combination with the original dataset.
    - **Ghostbusters with Emotion-Based Augmentations:** Training sets contained original data supplemented with stylistically transformed text in varying proportions.
3. **Training Regimen**: Models were trained using datasets with incrementally increasing numbers of training points: 10, 25, 50, 75, and 100 training samples. Performance was evaluated on test sets comprising both in-domain data (matching the training distribution) and out-of-domain data (from a different distribution) to measure the model's generalization capabilities.
4. **Measurements**: As mentioned prior, F1-scores were used to evaluate the performance. The reason for this is because of the balance F1 provides between the precision and recall. The reason why this is important for our authorship identification task is because there are huge consequences for both false positives and false negatives. Wrongly accusing someone of plagiarism, or not detecting it, should both be considered when measuring the performance of the model, and F1-scores best provide that type of measurement.

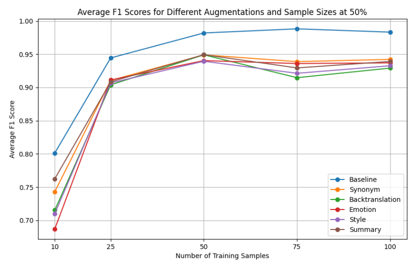
5 Results

To evaluate the performance of the Ghostbusters model under different configurations, we calculated the average F1-scores across multiple training seeds and plotted the results. Each seed represents a unique subset of training points sampled from the available dataset. Since the model was trained with varying amounts of data (10, 15, 25, 50, 75, and 100 points), the specific training points selected for each configuration varied across seeds, providing a robust measure of the model's performance under diverse conditions. These averaged F1-scores offer insight into the impact of data augmentations on both in-domain and out-of-domain generalization. Below we show the performance of our model with a proportion of 10 percent of augmented data compared to 50 percent augmented data to show the results of what would occur if we included more augmented data into the training set.

The results show a clear trend: as the proportion of augmented data increases, the performance of the model tends to decrease. The model performs decently well when only including 10 percent augmented data, but when we include 50 percent the performance drops considerably. This is evident across all types of data augmentation (synonym swapping, back translation, emotional tone adjustments, stylistic transformations, and summary-based augmentations). For example, with increasing training points (10 to 100), the baseline model consistently achieves higher F1 scores than models trained with augmented data. This indicates that the synthetic data generated through augmentation does not provide a performance boost, and is even detrimental to the model's effectiveness. We also see that the type of data augmentation does not matter too much. All of them perform similarly, with some having small F1 gains over others, but only for certain number of training points used. Overall, the data suggests that the synthetic augmentations did not enhance the model's ability to generalize or improve its performance, highlighting that the added complexity from synthetic data may not be beneficial in this context.



(a) Avg F1 Scores for Different augmentations with 10 percent augmented data



(b) Avg F1 scores for different augmentations with 50 percent augmented data

Figure 1

	10	25	50	75	100
Baseline	**0.8104**	**0.9464**	**0.9816**	**0.9886**	**0.9826**
Synonym	0.8056	0.9281	0.9751	0.9820	0.9753
Back Translation	0.7970	0.9168	0.9719	0.9795	0.9725
Emotion	0.7869	0.9254	0.9750	0.9822	0.9763
Style	0.7927	0.9257	0.9734	0.9779	0.9746
Summary	0.7946	0.9101	0.9703	0.9797	0.9691

Table 1: F1 scores for different models with 10 percent augmented data included at different amounts of training data

	10	25	50	75	100
Baseline	**0.8013**	**0.9443**	**0.9819**	**0.9881**	**0.9832**
Synonym	0.7430	0.9105	0.9490	0.9388	0.9421
Back Translation	0.7153	0.9038	0.9491	0.9146	0.9291
Emotion	0.6869	0.9112	0.9403	0.9359	0.9369
Style	0.7101	0.9067	0.9394	0.9212	0.9326
Summary	0.7625	0.9084	0.9492	0.9293	0.9390

Table 2: F1 scores for different models with 50 percent augmented data included at different amounts of training data

6 Discussion

Following our experiments, we found that our results were quite noisy across different proportions of augmented data. After averaging our results across seeds, we see that for ghostbuster, adding augmented data only seems to make the model perform worse than before on every size training size we gave it. There are a couple hypotheses we have for why this happened. Firstly, it is possible that our augmentations were not high quality enough. For instance, in our style augmentations, there were times when GPT-3.5 simply refused to significantly change the style of the text it was given. Moreover, the question of labeling the augmented data comes into question as we use GPT-3.5 itself to generate the augmentations. It can be argued that if we use any AI model to do augmentations, the text itself must then be labeled as AI-generated. The fact that we use GPT-3.5 to the augmentations may also heavily affect how our model sees a piece of text, as it only works off of the probability vectors generated from our set of weaker models. Using GPT-3.5 to do augmentations may have just inflated the probabilities for whatever text we augmented, leading to simply more noise for our logistic regression model to handle. Lastly, we have to consider that if the data augmentation gave back a drastic change, then the original label assigned to the training point would not even be relevant to it anymore, confusing the model with the amount of variation now present in the training data.

7 Future Work

There are a couple of directions that we can go in the future. One of the things that we could look into is the data augmentation quality control. One way that we could increase the quality of the augmentations is to potentially use more robust LLM models, from ChatGPT or Claude. This way, we ensure that whatever data augmentation procedure we are using is up-to-par and does not degrade our performance. One of the other things we noticed was that when we increased the number of changes to a text, the performance of the model decreased. We should do some studies to figure out a saturation point: what is the point at which there are too many augmentations to the original training data. Another thing we could look into is running experiment exclusively on the out-of-domain data that the original Ghostbusters model wasn't particularly good at, like analyzing essay's in the tone of a foreign exchange student. This way, we can truly is the effectiveness of out data augmentation methods and see which augmentation methods improve performance in this domain of data, while not sacrificing performance on the in-domain data.

8 Contributions

- **Ivan**
  - Scripts for Data Augmentations
  - Obtained augmented datasets
- **Leo**
  - Rewriting Code base to remove deprecated models
  - Running experiments
- **Ayush**
  - Debugging Codebase to create log probabilities for human and AI-generated essays

References

[1] A. Bakhtin, S. Gross, M. Ott, Y. Deng, M. Ranzato, and A. Szlam, "Real or Fake? Learning to Discriminate Machine from Human Generated Text," Nov. 25, 2019, arXiv: arXiv:1906.03351. Accessed: Oct. 13, 2024. [Online]. Available: http://arxiv.org/abs/1906.03351

[2] J. Chen, D. Tam, C. Raffel, M. Bansal, and D. Yang, "An Empirical Survey of Data Augmentation for Limited Data Learning in NLP," Transactions of the Association for Computational Linguistics, vol. 11, pp. 191–211, 2023, doi: 10.1162/tacl_a_00542.

[3] H. Dai et al., "AugGPT: Leveraging ChatGPT for Text Data Augmentation," Mar. 20, 2023, arXiv: arXiv:2302.13007. Accessed: Oct. 13, 2024. [Online]. Available: http://arxiv.org/abs/2302.13007

[4] B. Ding et al., "Data Augmentation using Large Language Models: Data Perspectives, Learning Paradigms and Challenges," Jul. 02, 2024, arXiv: arXiv:2403.02990. Accessed: Oct. 12, 2024. [Online]. Available: http://arxiv.org/abs/2403.02990

[5] D. O. Eke, "ChatGPT and the rise of generative AI: Threat to academic integrity?," Journal of Responsible Technology, vol. 13, p. 100060, Apr. 2023, doi: 10.1016/j.jrt.2023.100060.

[6] S. Y. Feng et al., "A Survey of Data Augmentation Approaches for NLP," in Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 968–988. doi: 10.18653/v1/2021.findings-acl.84.

[7] S. Gehrmann, H. Strobelt, and A. M. Rush, "GLTR: Statistical Detection and Visualization of Generated Text," Jun. 10, 2019, arXiv: arXiv:1906.04043. Accessed: Oct. 13, 2024. [Online]. Available: http://arxiv.org/abs/1906.04043

[8] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein, "A Watermark for Large Language Models," May 01, 2024, arXiv: arXiv:2301.10226. Accessed: Oct. 13, 2024. [Online]. Available: http://arxiv.org/abs/2301.10226

[9] I. Mahama, D. Baidoo-Anu, P. Eshun, B. Ayimbire, and V. Eggley, "ChatGPT in Academic Writing: A Threat to Human Creativity and Academic Integrity? An Exploratory Study," Indonesian Journal of Innovation and Applied Sciences (IJIAS), vol. 3, pp. 228–239, Oct. 2023, doi: 10.47540/ijias.v3i3.1005.

[10] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn, “DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature,” Jul. 23, 2023, arXiv: arXiv:2301.11305. Accessed: Oct. 13, 2024. [Online]. Available: http://arxiv.org/abs/2301.11305

[11] X. Peng, Y. Zhou, B. He, L. Sun, and Y. Sun, “Hidding the Ghostwriters: An Adversarial Evaluation of AI-Generated Student Essay Detection,” in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 10406–10419. doi: 10.18653/v1/2023.emnlp-main.644.

[12] M. Toshevska and S. Gievska, “A Review of Text Style Transfer using Deep Learning,” Sep. 30, 2021, arXiv: arXiv:2109.15144. Accessed: Oct. 13, 2024. [Online]. Available: http://arxiv.org/abs/2109.15144

[13] A. Uchendu, T. Le, K. Shu, and D. Lee, “Authorship Attribution for Neural Text Generation,” in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 8384–8395. doi: 10.18653/v1/2020.emnlp-main.673.

[14] V. Verma, E. Fleisig, N. Tomlin, and D. Klein, “Ghostbuster: Detecting Text Ghostwritten by Large Language Models,” Apr. 05, 2024, arXiv: arXiv:2305.15047. Accessed: Oct. 13, 2024. [Online]. Available: http://arxiv.org/abs/2305.15047

[15] R. Zellers et al., “Defending Against Neural Fake News,” Dec. 11, 2020, arXiv: arXiv:1905.12616. Accessed: Oct. 13, 2024. [Online]. Available: http://arxiv.org/abs/1905.12616