GHOSTBUSTER: DETECTING TEXT GHOSTWRITTEN BY LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce Ghostbuster, a state-of-the-art system for detecting AI-generated text. Our method works by passing documents through a series of weaker language models and running a structured search over possible combinations of their features, then training a classifier on the selected features to determine if the target document was AI-generated. Crucially, Ghostbuster does not require access to token probabilities from the target model, making it useful for detecting text generated by black-box models or unknown model versions. In conjunction with our model, we release three new datasets of human and AI-generated text as detection benchmarks that cover multiple domains (student essays, creative fiction, and news). Ghostbuster averages 99.0 F1 across all three datasets, outperforming previous approaches such as GPTZero and DetectGPT by up to 41.6 F1.

1 INTRODUCTION

Text generation tools such as ChatGPT are capable of producing a wide range of fluent text that closely approximates human-authored text. However, the use of language models to generate text that readers do not know is AI-generated introduces concerns about the authenticity and trustworthiness of text across a range of applications. The use of LLMs for classroom assignments raises questions about the originality of student work. Concerns that students are submitting assignments *ghostwritten* by language models has led many schools to adapt by restricting the use of ChatGPT and similar models (Heaven, 2023). In addition, because text generation models are prone to factual errors, the use of text generation to ghostwrite news articles or other informative text means that readers may desire to know if such tools have been used when deciding whether to trust a source.

Several detection frameworks have been proposed to address this issue, such as GPTZero (Tian, 2023) and DetectGPT (Mitchell et al., 2023). While these frameworks offer some level of detection, we find that their performance falters on datasets that they were not originally evaluated on (Section 6). In addition, the high false positive rates of these models raise potential ethical concerns because they jeopardize students whose genuine work is misclassified as AI-generated; in particular, text by non-native speakers of English is disproportionately flagged as AI-generated Liang et al. (2023).

We also introduce three new datasets for benchmarking detection of AI-generated text across different domains. Our *creative writing* dataset includes human-authored stories from the `r/WritingPrompts` subreddit. Our *news* dataset includes human-authored articles from the Reuters 50-50 dataset (Houvardas & Stamatatos, 2006), which consists of 50 train and 50 test articles for each of 50 authors. Finally, our *student essay* dataset includes student essays from the IvyPanda essay dataset (IvyPanda). For each document in each dataset, we generate corresponding ChatGPT articles based on the same prompt, summary, or article headline.

In this paper, we present a method for detection based on structured search and linear classification. First, we pass all documents through a series of weaker language models, ranging from a unigram model to the

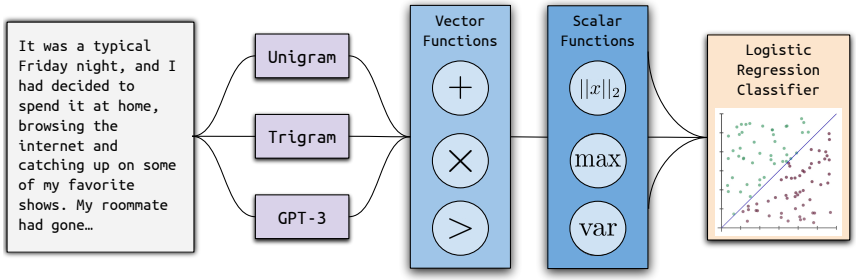

Figure 1: An outline of the classification algorithm. First, we generate possible combinations of features outputted by a sequence of weaker models. Then, we run a structured search over combinations of the model outputs and train a linear classifier on the selected features.

non-instruction-tuned GPT-3 `davinci`. Given the word probabilities from these models, we search over a space of vector and scalar functions which combine these probabilities into a small set of features. We then feed these features into a linear classifier, as described in Section 4. Averaged across all three datasets, our model achieves 99.0 F1 on document-level identification, outperforming GPTZero and DetectGPT by an average margin of 23.7 F1.

2 Related Work

AI-generated text exhibits qualitative differences from human-authored text, though these are often subtle. Guo et al. (2023) found that while volunteers often rated ChatGPT answers as more helpful than human ones, ChatGPT answers were more formal, more strictly focused, and used more conjunctions. Jawahar et al. (2020) found that GPT-2 responses that a model misclassified as human-authored tended to be very short and contained issues of factuality, repetition, contradiction, and incoherence. Other work has aimed to ensure that AI-generated text can be detected through deliberate watermarking of AI outputs (Aaronson, 2023; Kirchenbauer et al., 2023; Zhao et al., 2023; Kamaruddin et al., 2018). Watermarking has the benefit of providing guarantees on the probability that text is successfully detected, though it relies on model designers incorporating watermarks into models.

Several tools have recently been introduced to detect AI-generated text. Gehrmann et al. (2019) introduced GLTR, a suite of statistical tools to aid humans in detecting AI-generated text, which include overlaying text with the text's top-k annotation in different colors. Uchendu et al. (2020) use a RoBERTA-based model to identify whether two texts are generated by the same method, whether a text is AI-generated, and which of a set of candidate methods generated a text. DetectGPT (Mitchell et al., 2023) uses the fact that unlike human-authored text, generated text lies in regions of the probability space where nearby samples often have lower model probability. It generates random perturbations of the text from a generic LM to detect AI-generated text, then gets probabilities of the original text and perturbations from the model that might have generated the text. Recent supervised methods have trained models based on logistic regression, RoBERTa, and T5 to distinguish between human-authored and AI-generated text (Guo et al., 2023; Chen et al., 2023). Concurrent with this work, Bhattacharjee et al. (2023) uses contrastive domain adaptation for unsupervised AI-generated text detection.

However, Sadasivan et al. (2023) argue that there is an upper bound on the performance of generated text detectors and find that many detectors are brittle to paraphrasing attacks, including DetectGPT Mitchell et al. (2023), GLTR Gehrmann et al. (2019), other zero-shot methods Ippolito et al. (2020); Solaiman et al.

Prompt: As the chosen successor to Satan's throne, you must now undergo rigorous training to become the best at literally everything for when people challenge you for their souls.

Human Written Text	ChatGPT Generated Text
Halfway through the four hundred and eighty-seventh game of chess, I looked up at the demon on the opposite side of the table. "Why are we doing this again?" He harrumphed disdainfully. "Are you really that stupid? You need to master all arts and tasks so that you may -" "Yes, yes, I know," I reply, waving a hand to indicate impatience. "Challenge for souls, yadda yadda. But why not just… not let people challenge me? Why not just take their souls?" It seemed like common sense - to me, at least. The demon's jaw dropped, his eyes going wide. "You monster."	In the fiery pits, amidst sin and brimstone, I stood as the anointed heir to Satan's throne. Tasked with safeguarding the souls and warding off contenders, I embarked on rigorous training. Every day, wielding the trident, I mastered ancient combat techniques. By nightfall, I unraveled forbidden knowledge to outwit those desperate for redemption. Treading across desolated lands, I honed my persuasiveness and intellect, leaving no question unanswered. Climbing treacherous mountains, my endurance multiplied, for resilience was my armor against challengers. As the chosen successor, I vowed to become the epitome of mastery, prepared to face any adversary that dared to challenge my reign.

Figure 2: An example comparison between a human written text and a ChatGPT written text. All ChatGPT generated documents are generated by the same prompts used to write the human documents.

(2019), as well as OpenAI's generated text detectors OpenAI (2019). In this paper, we focus primarily on the setting in which entire paragraphs or documents were generated by language models, noting that adversarial prompting or paraphrasing-based attacks are an avenue for future work. Liang et al. (2023) found that essays written by non-native English speakers were disproportionately misclassified as AI-generated by seven GPT-generated text detectors.[1] Section 8 discusses Ghostbuster's performance on non-native English speaker data and mitigation efforts to prevent disproportionate misclassification of non-native English speakers' writing.

3 DATASETS

We collected three new datasets for benchmarking detection of AI-generated text across the domains of creative writing, news, and student essays. For each of the three datasets, we collected ChatGPT-generated text corresponding to the human-authored text. All datasets were generated using `gpt-3.5-turbo`.

Our student essay dataset is based on essays from IvyPanda, which consists of student essays across a range of disciplines. For each essay in the dataset, we first generate a prompt corresponding to the essay (see Appendix A), then generate a corresponding essay that responds to that prompt (see Table 1).

Our news dataset is based on the Reuters 50-50 authorship identification dataset (Houvardas & Stamatatos, 2006), which consists of 5000 news articles by 50 journalists. Because we did not have access to ground truth headlines or summaries for these articles, we first prompted ChatGPT to generate a headline for each article (see Appendix A), then prompted it to write an article based on each generated headline (see Table 1).

Our creative writing dataset is based on the subreddit `r/WritingPrompts`, a forum in which users share creative writing prompts and craft stories in response to these prompts. In order to avoid contamination from ChatGPT-written content, we collected data from the top 50 posters in October 2022 and scraped the last 100 posts by each of these users. For each story in the dataset, we generate a corresponding GPT example (see Table 1).

Additional details on the prompting process and dataset sizes are in Appendix A. All our final training datasets are evenly split between human-authored and ChatGPT-generated text. For each task, the datasets

[1]The exact detectors were not specified.

Dataset	Student Essays	News Articles	Creative Writing
Original Prompt	`Write an essay in {words} words to the prompt: {prompt}`	`Write a news article in {length} words with the following headline {headline}.`	`Write a story in {words} words to the prompt: {prompt}`
Sample Generalization Prompt	`You are a student, who is writing an essay in response to the prompt {prompt}. What would you write in {words} words?`	`You are a news reporter, who is writing an article with the headline {headline}. What would you write in {words} words?`	`You are an author, who is writing a story in response to the prompt {prompt}. What would you write in words words?`
Sample Generalization Prompt	`Write a words-word essay in the style of a high-school student in response to the following prompt: {prompt}.`	`Write a {words}-word news article in the style of a New York Times article based on the headline {headline}.`	`Write a words-word story in the style of a beginner writer in response to the prompt {prompt}.`

Table 1: Sample prompts used to produce paired ChatGPT-generated data. For creative writing, we set `words` equal to the number of words in the human-authored example rounded to the nearest 100, and `prompt` with the prompt corresponding to each story. This approach is intended to prevent document length or content effects from trivializing the detection task.

are divided into train, validation, and test sets. To validate task difficulty and ensure no artifacts remain in the datasets, we asked human reviewers to label subsets of the essays as human- or AI-generated (see Section 5).

Evaluation Datasets To evaluate generalization to different models, we collected Claude-generated text corresponding to the same prompts for each of the three datasets (see Table 1). Because reducing the false positive rate is particularly important for applications such as detecting student use of AI-generated text, we evaluate accuracy on some datasets of human text alone (i.e., a precision-only evaluation), including several datasets of text by non-native English speakers (details in Section 8).

4 MODEL

Our model uses a three-stage training process: *computing probabilities*, *selecting features*, and *classifier training*. First, Ghostbuster passes each document through a series of language models that are weaker than the target model to compute token log probabilities for each document. Our approach uses a unigram model, a Kneser-Ney trigram model, and two early GPT-3 models (`ada` and `davinci`, without instruction tuning) to obtain these probabilities. Then, Ghostbuster selects features by running a structured search procedure over a space of vector and scalar functions that combine these probabilities. To do so, we define a set of operations that combine these features and run forward feature selection on them. Finally, we train a simple classifier on the best probability-based features and some additional manually-selected features.

4.1 FEATURE SELECTION

Feature selection proceeds in two stages: we first generate a set of features (Table 2), then combine them using Algorithm 1. To generate features, we first outline the 13 scalar and vector functions in Table 2.

Vector Functions	Scalar Functions
$f_{\text{add}_i} = p_{1_i} + p_{2_i}$	$f_{\text{max}} = \max p$
$f_{\text{sub}_i} = p_{1_i} - p_{2_i}$	$f_{\text{min}} = \min p$
$f_{\text{mul}_i} = p_{1_i} \cdot p_{2_i}$	$f_{\text{avg}} = \frac{1}{\lvert p \rvert} \sum_i p_i$
$f_{\text{div}_i} = p_{1_i} / p_{2_i}$	$f_{\text{avg-top25}} = \frac{1}{\lvert p \rvert} \sum_{i \in T_p} p_i$
$f_{\dot{\iota}_i} = \mathbb{1}_{\{p_{1_i} > p_{2_i}\}}$	$f_{\text{len}} = \lvert p \rvert$
$f_{\dot{\imath}_i} = \mathbb{1}_{\{p_{1_i} < p_{2_i}\}}$	$f_{\text{L2}} = \lvert\lvert p \rvert\rvert_2$
	$f_{\text{var}} = \frac{1}{n} \sum_i (p_i - \mu_p)^2$

Table 2: List of vector and scalar functions used for feature generation. Vector functions take in two vectors of log probabilities $p_1, p_2 \in \mathbb{R}^n$ and output a single vector $f \in \mathbb{R}^n$, where n is the number of tokens in a document. On the other hand, scalar functions take in an input vector $p \in \mathbb{R}^n$ and output $f \in \mathbb{R}$. Here, T_p denotes the indices that contain the top 25 lowest values in p and μ_p denotes the average value of p.

Algorithm 1 Subroutine FIND-ALL-FEATURES

Require: The previously picked feature p, depth $d \leq$ `max_depth`, vectors V of log probabilities (from unigram, trigram, `ada`, and `davinci` models)
Ensure: A list of all possible features
  Let $S = \emptyset$
  for all scalar functions f_s **do**
    Add $f_s(p)$ to S
  end for
  for all combinations of $p' \in V$ and functions f_v **do**
    Add FIND-ALL-FEATS$(f_v(p, p'), d + 1)$ to S
  end for

The scalar functions convert vector to scalars, and vector functions combine two vectors into one. In order to generate all possible features, we run Algorithm 1 four times, with the log probability vectors from each model as the starting features and a maximum depth of 3. Features thus take the form of combining three arbitrary log probabilities with vector functions, then reducing them to a scalar function. An example feature is `var(unigram_logprobs > ada_logprobs - davinci_logprobs)`. We provide more details on the implementation and outputs of the algorithm in Appendix B. For a version of Ghostbuster trained on each dataset, we run forward feature selection to find the best features, as listed in Appendix D. We analyze the relative importance of different features in Section 7.

4.2 Classifier

Ghostbuster's classifier is trained on combinations of the probability-based features chosen through structured search, and seven additional features (Appendix C) based on word length and the largest log probabilities. These additional features are intended to incorporate qualitative heuristics observed about AI-generated text. For example, AI-generated text may appear "uninformative," exhibiting patterns of surprisal that differ from human-authored text in ways that may be evident in the frequency of outliers or differences in log probabilities between tokens that the models rate as very likely or only moderately likely. In addition, AI-generated text may have a tendency to generate words that are split into fewer subword tokens than human-authored text. We analyze the relative importance of these features in Section 7.

The classifier itself is a logistic regression classifier trained with l_2 regularization and setting $C = 1$ that takes in these features and those chosen through structured search (Section 4.1).

5 Baselines

We evaluate Ghostbuster's performance relative to multiple methods, including supervised and unsupervised machine learning methods, and conduct human evaluation to validate task difficulty.

Model	All Datasets	News	Creative Writing	Student Essays
Perplexity only	81.5	82.2	84.1	92.1
Zero-Shot	51.3	52.4	48.2	53.4
RoBERTa	90.6	93.0	82.1	96.5
GPTZero	93.1	91.5	93.1	83.9
DetectGPT	57.4	56.6	48.2	67.3
Ghostbuster	**99.0**	**99.5**	**98.4**	**99.5**

Table 3: Ghostbuster in-domain results of evaluating methods on each of our datasets (F1). For the perplexity-only baseline, RoBERTa, and Ghostbuster, in-domain results are computed by training only on the target dataset. We note that these datasets are out-of-domain for GPTZero and DetectGPT.

We compare with DetectGPT (Mitchell et al., 2023), an unsupervised method that uses generates random perturbations of the text from a generic LM to detect AI-generated text, then gets probabilities of the original text and perturbations from the model that might have generated the text. We note that DetectGPT differs from our method in requiring access to log probabilities from the target model.

Our simplest supervised baseline is a linear classifier trained only on the perplexities of human-authored and AI-generated documents. In addition, we fine-tuned a RoBERTa-based model on human-authored and AI-generated documents, similar to the RoBERTa-based approaches in Uchendu et al. (2020), Guo et al. (2023), and Chen et al. (2023). We employ `roberta-large` with a logistic regression head, and fine-tune with early stopping. We also compared with GPTZero (Tian, 2023), a commercially available model that uses a mixture of approaches, including supervised training, perplexity, variance in perplexity, and internet search. Lastly, we provide a zero-shot comparison by prompting ChatGPT with `Was this text written by ChatGPT?` followed by the text.

Human Evaluation. We collected human annotations to validate the difficulty of our datasets and provide a human baseline. Six undergraduate and PhD students with previous experience using text generation models were given a random set of 50 documents, evenly sampling human-authored and AI-generated documents, and asked to label whether the documents were written by a human or AI. The average human accuracy was 59% (maximum = 80%, minimum = 34%), suggesting that this is a difficult task for human reviewers.

6 Results

6.1 In-domain Classification

We first evaluate Ghostbuster in-domain, where we train and classify on the same domain, presenting the results in Table 3. We find that, Ghostbuster achieves 99.0 F1 across all three datasets, outperforming DetectGPT by a margin of 41.6 and GPTZero by 5.9. Overall, Ghostbuster has strong in-domain performance, performing significantly better than the baselines presented in Section 5.

6.2 Generalization Across Datasets

While Ghostbuster outperforms previous approaches when evaluating and training on the same domain, we note that this comparison is unfair since these datasets are out-of-domain for GPTZero and DetectGPT. As such, we provide results in Table 4 when evaluating ghostbuster out-of-domain (evaluated on one domain, trained on all other domains). Still, Ghostbuster achieves 97.0 F1 averaged across all conditions, outper-

Model	**News**	**Creative Writing**	**Student Essays**
Perplexity only	71.9	49.0	93.4
Zero-Shot	52.4	48.2	53.4
RoBERTa	83.2	**97.2**	69.3
GPTZero	91.5	93.1	83.9
DetectGPT	56.6	48.2	67.3
Ghostbuster	**97.9**	95.3	**97.7**

Table 4: Out-of-domain results of evaluating methods on each of our datasets (F1). For the out-of-domain setting, the perplexity-only, RoBERTa, and Ghostbuster were trained on two domains and evaluated on the third (e.g., train on news and creative writing, evaluate on essays).

Ablation	**In-Domain**				**Out-of-Domain**		
	All	News	Creative Writing	Student Essays	News	Creative Writing	Student Essays
Perplexity only	81.5	82.2	84.1	92.1	71.8	49.0	93.4
Feature depth 1	93.7	96.9	89.6	93.9	93.7	81.3	87.3
Feature depth 2	98.3	98.1	98.1	98.8	95.9	95.2	93.1
Feature depth 3, only handcrafted	80.5	79.6	78.2	83.6	75.8	77.2	77.2
Feature depth 3, no handcrafted	98.9	99.0	98.9	99.5	97.8	93.4	97.4
Feature depth 3, N-gram only	88.2	91.8	93.7	96.5	70.1	78.5	75.5
Feature depth 3, N-gram + Ada	98.8	99.3	**99.5**	**99.8**	97.3	90.3	91.9
Ghostbuster (full model)	**99.0**	**99.5**	98.4	99.5	**97.9**	**95.3**	**97.7**

Table 5: Ablation results of evaluating methods on each of our datasets (F1).

forming DetectGPT and GPTZero. These results suggest that Ghostbuster's performance gains are robust with respect to the similarity of the training and testing datasets.

6.3 Generalization Across Models

In addition to providing out-of-domain results when generalizing across domains, we also provide results for Ghostbuster when generalizing across the target model. In Table 6, we provide results when evaluating on a Claude-generated dataset. While Ghostbuster outperforms other approaches with 92.2 F1, the lower score suggests that generalization across models requires some additional training data.

7 Analysis

7.1 Ablations

We conduct multiple ablations to understand the role of the depth of structured search over features, probabilities from different models, and types of features used (from structured search vs. handcrafted) on model performance. As per the results in Table 5, we observe that depths lower than 3 tend to underfit the data, whereas depths greater than 3 tend to overfit the data. In addition, we notice that omitting the usage of `davinci` results in decreased performance. Lastly, we observe that while in-domain performance remains similar when removing the handcrafted features, the generalization performance goes down, suggesting its importance in preventing overfitting.

Ablation	Claude	GPT prompt variants
Perplexity only	84.1	85.3
RoBERTa	87.8	97.4
GPTZero	75.6	96.1
DetectGPT	64.2	70.8
Ghostbuster	**92.2**	**99.5**

Table 6: Results on robustness to different generation models and prompts (F1).

7.2 Robustness

We also evaluate Ghostbuster's robustness to changes in prompting strategies, outlined in Table 6. Ghostbuster achieves over 99% accuracy across prompt variants, compared to 95.6% achieved by RoBERTa and 96.1% achieved by GPTZero. In addition to demonstrating that Ghostbuster performs well across multiple variations of a prompt, this suggests that Ghostbuster's performance is not deterred by stylistic or semantic suggestions within the prompt.

8 Ethics and Limitations

We train and evaluate Ghostbuster on three datasets that represent a range of domains, but note that these datasets are not representative of all writing styles or topics and contain predominantly British and American English text. Thus, incorrect predictions by Ghostbuster are particularly likely for text that represents a distributional shift from Ghostbuster's training. Issues relating to improving model performance on shorter text, a broader range of domains, varieties of English besides American and British English, and robustness to edits are important areas for future work. However, for Ghostbuster, we investigated mitigations for improving performance on non-native English speaker data because it is a critical source of potential near-term harms.

Liang et al. (2023) sampled 91 TOEFL[2] essays and found that more than half of the essays were misclassified as AI-generated by seven GPT-generated text detectors. We evaluated Ghostbuster's performance on two datasets of non-native English speaker data: the TOEFL 11 dataset (Blanchard et al., 2013) and the Lang8 dataset (Mizumoto et al., 2011) The TOEFL 11 dataset contains university-level essays written on the TOEFL exam, with an even number of essays by authors whose first languages (L1s) were Arabic, Chinese, French, German, Hindi, Italian, Japanese, Korean, Spanish, Telugu, and Turkish. The Lang8 dataset contains data from an online forum used by language learners (from a range of countries, and particularly from Japan). We divided the TOEFL data into a training set of 1,000 essays and a test set of 1,000 essays. We evaluated Ghostbuster's out-of-the-box performance on 1,000 examples from the Lang8 posts and the TOEFL 11 test set essays.

More broadly, users wishing to apply Ghostbuster to real-world cases of potential off-limits usage of text generation (e.g., identifying ChatGPT-written student essays) should be wary that incorrect predictions by Ghostbuster are particularly likely for shorter text, domains further from those on which Ghostbuster was trained (e.g., text messages), text in varieties of English besides Standard American or British English, or in non-English languages, text written by non-native speakers of English, AI-generated text that has been edited or paraphrased by a human and text that was generated by prompting an AI model to paraphrase or adjust a human-authored input. To avoid perpetuation of algorithmic harms due to these limitations, we

[2]Test of English as a Foreign Language, an exam taken by non-native speakers of English to attend English-speaking universities.

Model	TOEFL 11	Lang8
Perplexity only	98.1	98.6
RoBERTa	98.1	98.6
GPTZero	**100.0**	**99.2**
DetectGPT	**100.0**	98.6
Ghostbuster	99.9	95.5
Ghostbuster + TOEFL	**100.0**	98.6

Table 7: Results on non-native English speaker data (accuracy, which equals precision here since all documents are human-authored).

strongly discourage incorporation of Ghostbuster into any systems that automatically penalize students or other writers for alleged usage of text generation without human supervision.

9 Conclusion

We introduced Ghostbuster, a model for detecting AI-generated language that uses structured search on token probabilities from weaker models to identify whether a given document was AI-generated. We validated Ghostbuster by evaluating its performance on datasets from three domains (news, student essays, and creative fiction writing), as well as through generalization experiments on text generated by different models and using different prompts. We also release our three datasets as benchmarks for evaluating performance on detecting AI-generated text. Ghostbuster achieves over 98 F1 across all datasets on in-domain detection of AI-generated text, representing substantial progress over currently available models for detection of AI-generated text.

Ghostbuster shines when generalizing across domains, achieving over 97 F1 for all generalizations generated by the same model. Furthermore, because Ghostbuster does not require access to target model probabilities, it is well-designed to identify text generated from black-box or unknown models, which is especially an advantageous for the most commonly used commercial models (e.g., ChatGPT or Claude). Further work could improve performance on model generalization by extending the proposed structured search to incorporate different potential features could help to make further progress at identification of AI-generated text.

Future work could examine tradeoffs between lowering the false positive and false negative rates of AI-generated text detectors for different applications. For detection of AI-generated student essays, lowering the risk of false positives is a key priority to avoid false accusations of AI ghostwriting. In other settings, however, false positives are less concerning relative to false negatives. For example, if detectors are used to prevent AI-generated text from being used in training data or to help people decide whether online news might be AI-generated, the ideal model calibration may differ. Other avenues for future work include improving robustness to perturbations of AI-generated outputs, such as lightly editing to avoid detection, and different task formulations, including detection at the paragraph level for documents that combine human-authored and AI-generated text.

References

Scott Aaronson. Watermarking of large language models. Workshop on Large Language Models and Transformers, Simons Institute, UC Berkeley, 2023. URL https://www.youtube.com/watch?v=2Kx9jbSMZqA.

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. Conda: Contrastive domain adaptation for ai-generated text detection, 2023.

Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. TOEFL11: A CORPUS OF NON-NATIVE ENGLISH. *ETS Research Report Series*, 2013(2):i–15, December 2013. doi: 10.1002/j.2333-8504.2013.tb02331.x. URL https://doi.org/10.1002/j.2333-8504.2013.tb02331.x.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. Gpt-sentinel: Distinguishing human and chatgpt generated content, 2023.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. GLTR: Statistical detection and visualization of generated text, 2019.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How close is ChatGPT to human experts? comparison corpus, evaluation, and detection, 2023.

Will Heaven. Chatgpt is going to change education, not destroy it. *MIT Technology Review*, 2023. URL https://www.technologyreview.com/2023/04/06/1071059/chatgpt-change-not-destroy-education-openai/.

John Houvardas and Efstathios Stamatatos. N-gram feature selection for authorship identification. In *Artificial Intelligence: Methodology, Systems, Applications*, 2006.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic detection of generated text is easiest when humans are fooled, 2020.

IvyPanda. IvyPanda essay dataset. URL https://huggingface.co/datasets/qwedsacf/ivypanda-essays.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks V. S. Lakshmanan. Automatic detection of machine generated text: A critical survey, 2020.

Nurul Shamimi Kamaruddin, Amirrudin Kamsin, Lip Yee Por, and Hameedur Rahman. A review of text watermarking: Theory, methods, and applications. *IEEE Access*, 6:8011–8028, 2018. doi: 10.1109/ACCESS.2018.2796585.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. Gpt detectors are biased against non-native english writers. *Patterns*, 4(7):100779, 2023. ISSN 2666-3899. doi: https://doi.org/10.1016/j.patter.2023.100779. URL https://www.sciencedirect.com/science/article/pii/S2666389923001307.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. DetectGPT: Zero-shot machine-generated text detection using probability curvature, 2023.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. Mining revision log of language learning SNS for automated Japanese error correction of second language learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 147–155, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing. URL https://aclanthology.org/I11-1017.

OpenAI. GPT-2: 1.5b release, 2019. URL https://openai.com/research/gpt-2-1-5b-release.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. Can ai-generated text be reliably detected?, 2023.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. Release strategies and the social impacts of language models, 2019.

Edward Tian. GPTZero: Home, 2023. URL https://gptzero.me.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8384–8395, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.673. URL https://aclanthology.org/2020.emnlp-main.673.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for ai-generated text. *arXiv preprint arXiv:2306.17439*, 2023.

A Prompting and Dataset Details

For the news and creative writing datasets, we first prompted the model to generate a headline or writing prompt, respectively, before generating the documents themselves from those prompts. Table 9 gives the full prompting strategy for the original prompts. Table 10 gives all the generalization prompts used by dataset. Table 8 gives details on dataset sizes and splits.

Domain	**Human Text Source**	**# Docs**	**Median Words per Doc**		
			Human (1,000 docs)	ChatGPT (5,000 docs)	*Claude* (1,000 docs)
Student Essays	IvyPanda (IvyPanda)	7,000	529	559	442
News Articles	Reuters 50-50 (Houvardas & Stamatatos, 2006)	7,000	498	510	384
Creative Writing	r/WritingPrompts	7,000	455	512	384

Table 8: Datasets introduced in this paper. For each domain, the 5,000 ChatGPT-generated documents are divided into 1,000 documents from the same prompt, and a 4,000-document "generalization set" that used different prompts to evaluate generalization. For each domain, 1,000 human-authored documents and the 1,000 ChatGPT-generated documents that used the same prompt were split into train, validation, and test sets used by Ghostbuster. The ChatGPT-generated generalization set, the Claude-generated text, and the British Academic Written English corpus were used only for evaluation of Ghostbuster, not training.

B Algorithmic Implementation Details

While the algorithm presented in Algorithm 1 produces equivalent results to our implementation, we make a few additional optimizations. First, we note that all vector functions in Table 2 are commutative, or posses the inverse operator. As such, we first create a list of possible vector combinations, avoiding double-counting. We noticed that this pruning results in around a 2/3rd reduction in the feature space. At a depth of 3, we have 2534 features, and at a depth of 2 we have 322 features.

Dataset	Prompting Strategy	Prompt
Student Essays	Generate essay	`Write a story in {words} words to the prompt: {prompt}`
News Articles	(1) Generate title	`Create a headline for the following news article: {doc}`
	(2) Generate article	`Write a news article in {length} words with the following headline {headline}.`
Student Essay	(1) Generate title	`Given the following essay, write a prompt for it: {doc}`
	(2) Generate story	`Write an essay in {words} words to the prompt: {prompt}`

Table 9: Prompts used to generate documents in each of the three proposed datasets.

C Additional Features

Ghostbuster uses the following handcrafted features in addition to those chosen through feature selection:

- Number of outliers (p_i ¿ 10), average value of top 25 and 25-50$^{\text{th}}$ largest log probabilities
- Average value of the 25 largest and 25-50$^{\text{th}}$ largest log probabilities of the vector $d - a$, where d is a vector of Davinci log probabilities and a is a vector of Ada log probabilities.
- Average length of the 25 longest and 25-50$^{\text{th}}$ longest words, measured in tokens.

D Best Features

In this section, we present the best features chosen through validation on each of the datasets. For a list of functions and features used, refer to Table 2.

```
avg(unigram + trigram < davinci)
var(unigram > ada - davinci)
avg-top-25(unigram - davinci / ada)
avg(ada > davinci / ada)
avg(trigram / ada < davinci)
max(unigram * davinci)
avg(unigram - ada * davinci)
var(unigram * trigram - ada)
max(trigram - davinci / unigram
```

Dataset	Student Essays	News Articles	Creative Writing
Original Prompt	`Write an essay in {words} words to the prompt: {prompt}`	`Write a news article in {length} words with the following headline {headline}.`	`Write a story in {words} words to the prompt: {prompt}`
Generalization Prompt 1	`You are a student, who is writing an essay in response to the prompt {prompt}. What would you write in {words} words?`	`You are a news reporter, who is writing an article with the headline {headline}. What would you write in {words} words?`	`You are an author, who is writing a story in response to the prompt {prompt}. What would you write in words words?`
Generalization Prompt 2	`Hi! I'm trying to write a words-word essay based on the following prompt: {prompt}. Could you please draft something for me?.`	`Hi! I'm trying to write a words-word news article based on the following headline: {headline}. Could you please draft something for me?.`	`Hi! I'm trying to write a words-word story on the following prompt: {prompt}. Could you please draft something for me?`
Generalization Prompt 3	`Write a words-word essay in the style of a high-school student in response to the following prompt: {prompt}.`	`Write a {words}-word news article in the style of a New York Times article based on the headline {headline}.`	`Write a words-word story in the style of a beginner writer in response to the prompt {prompt}.`
Generalization Prompt 4	`Write an essay with very short sentences in words words to the prompt {prompt}.`	`Write a news article with very short sentences in words words based on the headline {headline}.`	`Write a story with very short sentences in words words to the prompt {prompt}.`

Table 10: Full set of prompts used to produce paired ChatGPT-generated data. For creative writing, we set `words` equal to the number of words in the human-authored example rounded to the nearest 50, and `prompt` with the prompt corresponding to each story. This approach is intended to prevent document length or content effects from trivializing the detection task.