LLM-DetectAIve: a Tool for Fine-Grained Machine-Generated Text Detection

Mervat Abassy,[1,2*] Kareem Elozeiri,[1,3*] Alexander Aziz,[1,4*] Minh Ngoc Ta,[1,5*] Raj Vardhan Tomar,[1,6*] Bimarsha Adhikari,[1,9*] Saad El Dine Ahmed,[1,2*] Yuxia Wang,[1] Osama Mohammed Afzal,[1] Zhuohan Xie,[1] Jonibek Mansurov,[1] Ekaterina Artemova,[7] Vladislav Mikhailov,[8] Rui Xing,[1] Jiahui Geng,[1] Hasan Iqbal,[1] Zain Muhammad Mujahid,[1] Tarek Mahmoud,[1] Akim Tsvigun,[10] Alham Fikri Aji,[1] Artem Shelmanov,[1] Nizar Habash,[1,9] Iryna Gurevych,[1] Preslav Nakov[1]

[1]MBZUAI, [2]Alexandria University, [3]Zewail City of Science and Technology, [4]University of Florida, [5]Hanoi University of Science and Technology, [6]Cluster Innovation Center, University of Delhi, [7]Toloka AI, [8]University of Oslo, [9]New York University Abu Dhabi, [10]KU Leuven

Abstract

The ease of access to large language models (LLMs) has enabled a widespread of machine-generated texts, and now it is often hard to tell whether a piece of text was human-written or machine-generated. This raises concerns about potential misuse, particularly within educational and academic domains. Thus, it is important to develop practical systems that can automate the process. Here, we present one such system, **LLM-DetectAIve**, designed for fine-grained detection. Unlike most previous work on machine-generated text detection, which focused on binary classification, LLM-DetectAIve supports four categories: (i) human-written, (ii) machine-generated, (iii) machine-written, then machine-humanized, and (iv) human-written, then machine-polished. Category (iii) aims to detect attempts to obfuscate the fact that a text was machine-generated, while category (iv) looks for cases where the LLM was used to polish a human-written text, which is typically acceptable in academic writing, but not in education. Our experiments show that LLM-DetectAIve can effectively identify the above four categories, which makes it a potentially useful tool in education, academia, and other domains. LLM-DetectAIve is publicly accessible at https://github.com/mbzuai-nlp/LLM-DetectAIve.[1] The video describing our system is available at https://youtu.be/E8eT_bE7k8c.

[*]Equal contribution.

[1]This work was done during a summer internship at the NLP department, MBZUAI.

1 Introduction

The development of advanced large language models (LLMs), such as GPT-4, Claude-3.5, Gemini-1.5, Llama-70b (OpenAI, 2023; Anthropic, 2024; Gemini, 2023; Llama, 2024), improved the prevalence and the coherence of machine-generated content. This trend makes it increasingly difficult to differentiate between texts produced by machines from such written by humans (Macko et al., 2023; Wang et al., 2024b,c). As a result, there have been growing concerns about the authenticity and integrity of textual content (Crothers et al., 2023; Tang et al., 2024).

While many detectors have been developed to address this new challenge (Mitchell et al., 2023; Wang et al., 2024a), they often struggle to keep up with the rapid development of LLMs. Generations produced by new models are hard to detect as they become more coherent and represent out-of-distribution instances, compared to what detecting systems saw during training (Macko et al., 2024; Koike et al., 2024). Moreover, the use of prompting to generate more human-like texts or applying LLMs to refine or change the tone of human writings further complicates detection.

Most prior work on detecting machine-generated text focused on binary detection, i.e., predicting whether the text is generated by a machine or written by a human. This dichotomy leaves no space for mixed categories of human-machine collaboration. However, we argue for the need for additional categories, as machine-polishing of human-written text is acceptable in certain cases (e.g., for academic papers), but not in other (e.g., in education).

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing
System Demonstrations, pages 336–343

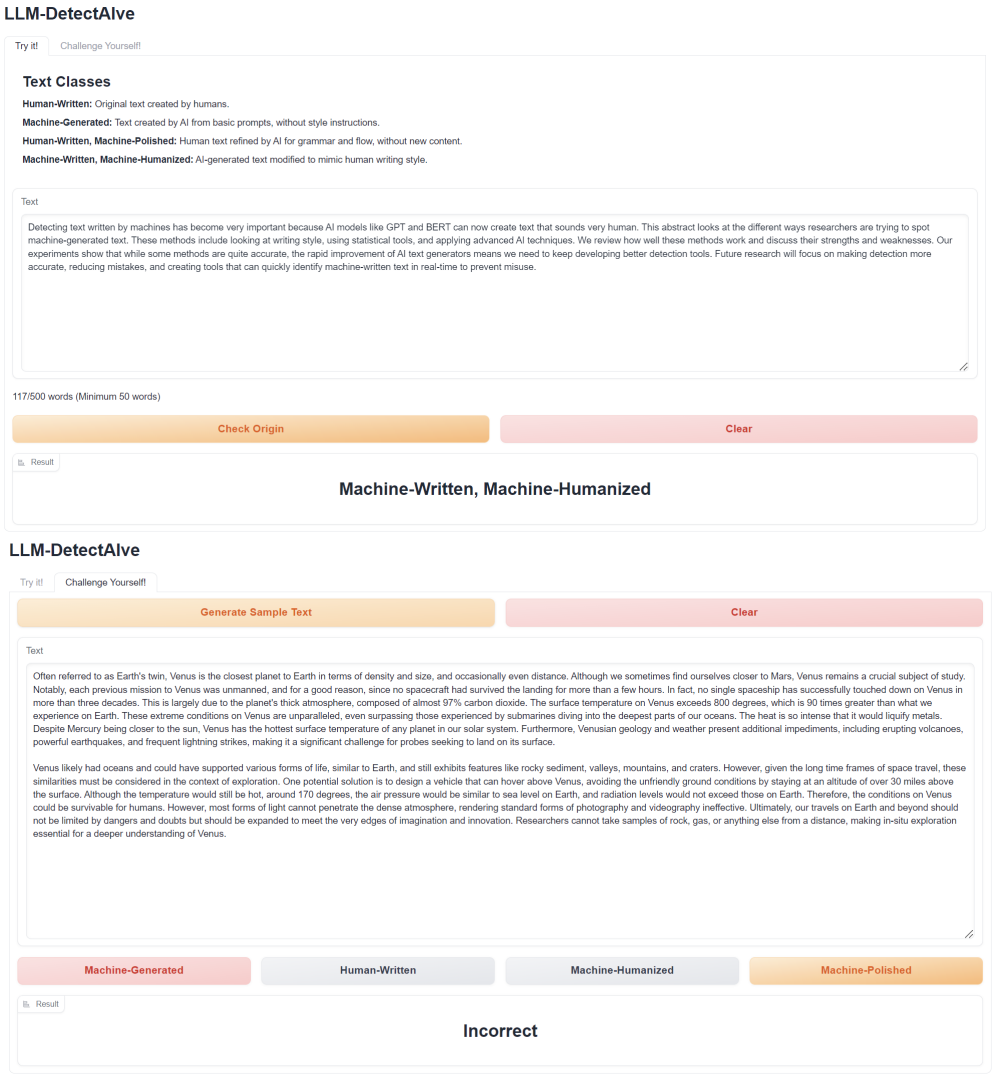

Figure 1: **LLM-DetectAIve interface:** automatic text detection (top) and human detector playground (bottom).

In education, using LLMs to complete entire assignments or even to polish human-written essays is typically prohibited (Susnjak, 2022). Therefore, it is important to perform fine-grained text classification. For example, detecting the use of LLMs in text humanization and refinement becomes critical to ensure the fair assessment of students' genuine knowledge and abilities. Fine-grained human/machine identification is also important for authorship detection in digital forensics.

To address this problem, we propose a new formulation of problem, as multi-way classification with the following labels:

I. **Human-Written**: the ext is created solely by a human author without GenAI assistance.

II. **Machine-Generated**: the text is entirely produced by a machine based on input prompts without any human intervention.

III. **Machine-Written Machine-Humanized**: the text is initially generated by a machine and then subtly modified to appear more human-like. This involves automatically tweaking the LLM to make the output appear more human.

IV. **Human-Written Machine-Polished**: the text is written by a human and then is refined or polished by a machine, e.g., to correct grammar, improve style, and/or optimize readability while trying to preserve the meaning of the original human text.

We further develop **LLM-DetectAIve**, a system that accurately distinguishes between different types of text generation and editing. With this, we aim to uphold academic integrity and ensure a fair evaluation process for both students and researchers.

Text Class	Generator	OUTFOX	Wikipedia	Wikihow	Reddit ELI5	arXiv abstract	PeerRead
			M4GT-Bench				
I	**Human**	14,043	14,333	15,999	16,000	15,998	2,847
II	**davinci-003**	3,000	3,000	3,000	3,000	3,000	2,340
	gpt-3.5-turbo	3,000	2,995	3,000	3,000	3,000	2,340
	cohere	3,000	2,336	3,000	3,000	3,000	2,342
	dolly-v2	3,000	2,702	3,000	3,000	3,000	2,344
	BLOOMz	3,000	2,999	3,000	2,999	3,000	2,334
	gpt4	3,000	3,000	3,000	3,000	3,000	2,344
			New Generations				
II + III + IV	**gpt-4o**	8,966	8,995	9,000	9,000	9,000	7,527
	gemma-7b	8,280	8,985	9,000	9,000	9,000	0
	llama3-8b	8,271	8,985	9,000	9,000	9,000	0
	llama3-70b	8,577	8,985	9,000	9,000	9,000	0
	mixtral-8x7b	17,001	8,985	9,000	9,000	9,000	0
	gemma2-9b	0	8,985	9,000	9,000	9,000	0
III	**gemini1.5**	0	1,652	1,601	904	0	0
	mistral-7b	0	2,993	3,000	0	0	2,344
IV	**gemini1.5**	0	1,652	1,601	904	2,994	586
	mistral-7b	0	2,993	3,000	0	0	2,344

Table 1: **Statistics about our datasets** across LLMs over the four classes: I. Human-Written, II. Machine-Generated, III. Machine-Written Machine-Humanized and IV. Human-Written Machine-Polished. For row II + III + IV, the data is approximately uniformly distributed across the three classes.

Our contributions are as follows:

- We reformulate the task as fine-grained multi-way classification.
- We collect a dataset for this reformulation using generations from a variety of LLMs.
- We build, evaluate, and compare several machine-generated text detectors on our new fine-grained dataset.
- We develop a Web-based demo that (*i*) allows users to input text and to obtain fine-grained classification prediction, and (*ii*) offers a playground for users to test their ability to detect texts with varying degrees of LLM involvement, according to the above 4-way fine-grained schema.

2 Dataset

To collect the dataset for our multi-way fine-grained detector, we first gathered datasets that were curated for binary machine-generated text detection from previous work, and then we extended the data into our four labels by introducing new corresponding generations. Sections 2.2 and 2.3 discuss the prompts we used for generation and data cleaning, respectively.

2.1 Data Overview

We build the new dataset by extending the M4GT-Bench (Wang et al., 2024b), which is an benchmark dataset for evaluating machine-generation text detectors that encompasses multiple generators and domains, including arXiv, Wikihow, Wikipedia, Reddit, student essays (OUTFOX), and peer reviews (PeerRead). From these sources, we sampled a subset comprising 79,220 human-written texts and 103,075 machine-generated texts.

Next, we expanded this dataset by (*i*) collecting additional machine-generated texts produced by new LLMs (e.g., GPT-4o), (*ii*) generating machine-written then machine-humanized texts, and (*iii*) polishing human-written texts using various LLMs. This resulted in 91,358 fully-MGTs, 103,852 machine-written then machine-humanized texts, and 107,900 human-written then machine-polished texts. Table 1 gives detailed statistics about the dataset.

For data generation, we used a variety of LLMs, including Llama3-8b, Llama3-70b (Llama, 2024), Mixtral 8x7b (Jiang et al., 2024), Gemma-7b, Gemma2-9b (Team, 2024), GPT-4o (OpenAI, 2023), Gemini-1.5-pro (Gemini, 2023), and Mistral-7b (Jiang et al., 2023). By incorporating a diverse array of LLMs and domains, we aim to enhance the detection accuracy within actual domains and generators, as well as improve generalization.

2.2 Generation Prompts

For the *Machine-Written Machine-Humanized* class, examples of prompts include *Rewrite this text to make it sound more natural and human-written* or "*Rephrase this text to be easy to understand and personable.*" For the *Human-Written Machine-Polished* class, we used prompts such as "*Paraphrase the provided text.*" or "*Rewrite this text so that it is grammatically correct and flows nicely.*" Additionally, we introduced a trailing prompt appended to each randomly selected prompt to prevent undesirable text that the LLM may prepend to its output, e.g., "*Only output the text in double quotes with no text before or after it. Text: {} Your response:*". We used 5-6 prompts per domain to generate data for the *Machine-Written Machine-Humanized* and *Human-Written Machine-Polished* classes. In addition to the *Machine-Generated* class, we used the original prompts from the M4GT-Bench dataset.

2.3 API Tools & Data Cleaning

For data generation, we used multiple APIs from OpenAI, Gemini, Groq, and DeepInfra, to generate a total of 303,110 texts for the three LLM-dependent classes. For each of the two new class generations, we limited the text length to 1,500 words in order to accommodate the context length restrictions of some smaller LLMs and to efficiently manage time and costs.

The output of the LLMs occasionally included formatting such as "Here is the paraphrased text:" and "Sure!" despite instructions in the trailing prompt to exclude any additional output. We removed these phrases in the post-processing with two considerations. On the one hand, this naturally occurs in real-world applications, i.e., humans will remove these irrelevant phrases when they use the target content. Moreover, the presence of these indicative artifacts could impact the detectors' generalization and the quality of the dataset, given that they are potentially unique for a specific text class.

3 Detection Models

We trained three detectors by fine-tuning RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2021), and DistilBERT (Sanh et al., 2019). DeBERTa is built upon BERT and RoBERTa by incorporating disentangled attention mechanisms and an enhanced mask decoder, which improves word representation.

Dataset	Detector	Learning rate	Weight Decay	Epochs	Batch Size
arXiv	RoBERTa	2e-5	0.01	10	16
	DistilBERT	2e-5	0.01	10	16
OUTFOX	RoBERTa	2e-5	0.01	10	16
	DistilBERT	2e-5	0.01	10	16
Full Dataset	RoBERTa	5e-5	0.01	10	32
	DeBERTa	5e-5	0.01	10	32

Table 2: **Hyper-parameter values** across the models.

Eventually, in the demo, we used DistilBERT, which is a compact and fast variant of BERT: 60% faster and 40% smaller, while retaining 97% of BERT's language understanding capabilities.

Table 2 shows the values of the hyper-parameters for each model. We used RoBERTa and DistilBERT in our domain-specific experiments. However, due to the inferior performance of DistilBERT to RoBERTa in our preliminary trials, we substituted DistilBERT with DeBERTa in the following experiments (DeBERTa is superior to RoBERTa).

4 Experiments and Evaluation

The previous studies have shown that the accuracy of detectors drops substantially when testing on out-of-domain examples (Wang et al., 2024b). To alleviate this, we propose three strategies: (*i*) train multiple domain-specific detectors, each specifically responsible for detecting inputs from one domain, (*ii*) train one universal detector using more training data across various domains, and (*iii*) leverage domain-adversarial neural network (DANN) for domain adaption.

4.1 Domain-Specific Detectors

We fine-tuned RoBERTa and DistilBERT using the data from arXiv and OUTFOX, using a ratio of training, validation, and test sets of 70%:15%:15%. The results are shown in Table 3. We can see that both RoBERTa and DistillBERT performed well on OUTFOX. Overall, RoBERTa is more robust over diverse domains, with accuracy greater than 95% on both domains, with a small number of mis-classifications occurring between classes with overlapping features, such as Machine-Generated vs. Human-Written, vs. Machine-Polished classes, as the confusion matrices in Figure 2 show.

However, in this setup, the users need to first specify the domain of the input text, which is an extra effort. To mitigate this, we further trained a universal model that does not require the user to select the domain.

Detector	Test Domain	Prec	Recall	F1-macro	Acc
RoBERTa	arXiv	95.82	95.79	95.79	95.79
	OUTFOX	95.67	95.43	95.53	95.65
DistilBERT	arXiv	88.98	87.97	87.93	87.79
	OUTFOX	96.66	96.65	96.65	96.65

Table 3: **Domain-specific performance** for RoBERTa and DistilBERT on arXiv and OUTFOX.

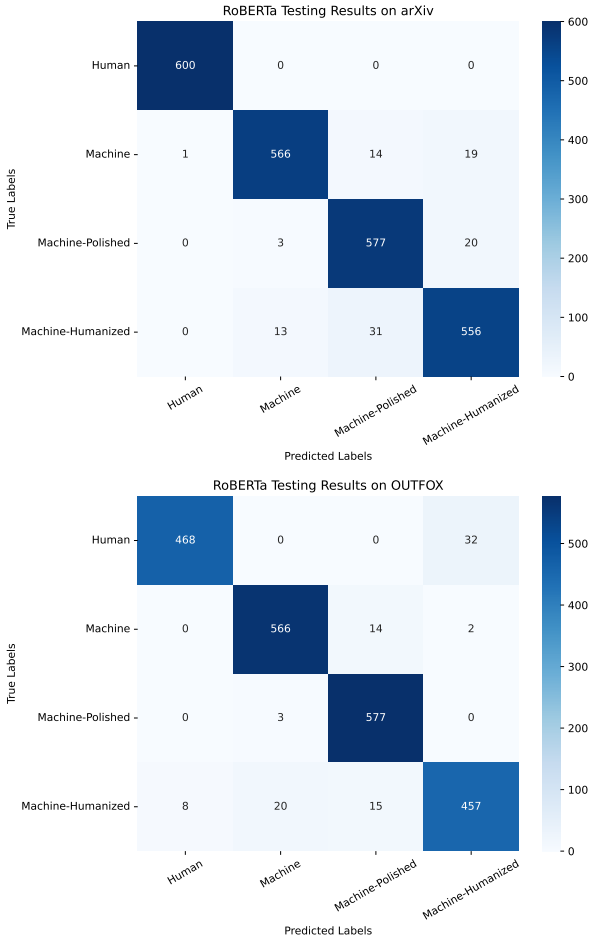

Figure 2: Domain-specific confusion matrix for RoBERTa on arXiv (top) and on OUTFOX (bottom).

4.2 Universal Detectors

We fine-tuned RoBERTa and DeBERTa using the full dataset; the data distribution for this is shown in Table 4. To reduce data imbalance and prevent the detector from favoring any particular class, we excluded some of the original data. The evaluation results in Table 5 indicate that DeBERTa consistently outperforms RoBERTa across all evaluation measures we use. Therefore, we deployed the fine-tuned DeBERTa as the backend detection model for our demo.

Domain	Human	Machine-Generated	Machine-Polished	Machine-Humanized
arXiv	15,998	18,000	18,000	18,000
Reddit	16,000	18,904	18,904	18,904
wikiHow	15,999	22,601	22,601	22,601
Wikipedia	14,333	22,615	22,615	22,615
PeerRead	2,847	4,684	4,684	4,684
Outfox	14,043	17,000	17,000	17,000

Table 4: **Distribution** of the data used for fine-tuning **universal detectors** based on RoBERTa and DeBERTa.

Detector	Prec	Recall	F1-Macro	Acc
RoBERTa	94.79	94.63	94.65	94.62
DeBERTa	95.71	95.78	95.72	95.71

Table 5: **Detector performance** on the full dataset.

4.3 DANN-Based Detector

In our domain-specific experiments above, we achieved strong performance when the domain of the text was provided. However, in cross-domain evaluation, the performance is sub-optimal as previous work has suggested (Wang et al., 2024b,c). In real-world scenarios, the domain would not always be specified, and thus we need a classifier that is as domain-independent as possible.. Thus, we investigated the use of *domain adversarial neural networks* (Ganin et al., 2017) to train a domain-robust detector.

DANN was initially designed to achieve domain adaptation by aligning representations across different domains with three major components:

- **Representation Extractor:** which builds a representation of the input data; here, we use RoBERTa.
- **Label Predictor:** to predict the class labels based on the representation; it is trained using labeled data from the source domain.
- **Domain Classifier:** connected to the representation via a *gradient reversal layer (GRL)*, it distinguishes between the source and the target domains. It multiplies the gradient by a negative constant during back-propagation, promoting domain-invariant representation.

The network is trained using standard back-propagation and stochastic gradient descent, optimizing the label classification loss while intentionally confusing the model regarding the domain by reversing the gradient from the domain classifier. This reduces the label classification loss while increasing the domain classification one.

Detector	Prec	Recall	F1-macro	Acc
RoBERTa	94.79	94.63	94.65	94.62
DANN+RoBERTa	**96.30**	**95.54**	**96.06**	**95.24**

Table 6: Comparing domain-specific RoBERTa vs. DANN+RoBERTa. The latter outperforms the former across all measures, indicating that decoupling the model from domain-specific representation is beneficial.

As a result, the Domain-Adversarial Neural Network (DANN) yields a representation that is independent of the domain. In our experiments, we trained the DANN to predict our four classes and to be as confused as possible when predicting the six sources/domains. The results are shown in Table 6. We can see that using domain adversarial training on top of RoBERTa-enhances the overall performance compared to just fine-tuning RoBERTa as in Section 4.2. This suggests that decoupling the model from domain-specific representation leads to an improvement in its overall performance.

4.4 Comparison to Existing Systems

There are several previously proposed systems for detecting machine-generated text, such as GPTZero,[2] ZeroGPT,[3] and Sapling AI detector,[4] but none of them supports four classes. GPTZero is the only one that goes beyond binary classification: it adds a *mixed text*; however, it limits users to only 40 free runs per day or 10,000 words per month for registered accounts. Thus, we could not perform comparison on our entire test dataset. Instead, we randomly sampled 60 machine-generated texts and 60 human texts (10 per source) per source. In this binary classification setting, LLM-DetectAIve achieved 97.50% accuracy, outperforming GPTZero, ZeroGPT, and Sapling AI, with 87.50%, 69.17%, and 88.33%, respectively.

4.5 Generalization Evaluation

To evaluate the generalization ability of our detector on unseen domains and generators, we experimented with testing on two additional datasets: MixSet (Ji et al., 2024) and IELTS essays written by individuals for whom English is a second language.[5]

[2] https://gptzero.me/

[3] https://www.zerogpt.com/

[4] https://sapling.ai/ai-content-detector

[5] https://huggingface.co/datasets/chillies/IELTS_essay_human_feedback

Dataset	Prec	Recall	F1-macro	Acc
IELTS	63.74	66.91	66.55	66.91
MixSet	59.18	64.25	54.95	60.08

Table 7: **Cross-domain evaluation** of our detector on unseen domains and generators: IELTS and MixSet.

For the IELTS essays, after deduplication, we randomly sampled 300 (essay problem statement, human-written essay) pairs, and then we produced the corresponding machine-written essays using the problem statements based on Llama3.1-70B. We further generated Machine-Written Machine-Humanized and Human-Written Machine-Polished. For MixSet, the original dataset contains a total of 3,600 examples, with 300, 300, 600, and 2,400 examples for Human-Written, Machine-Generated, Machine-Written Machine-Humanized and Human-Written Machine-Polished, respectively. It involves models such as Llama2-70B and GPT-4, and text covering domains of email content, news, game reviews, and so on.

The results are shown in Table 7, where we can see that the detector performs much worse on unseen domains and generators, compared to in-domain and in-generator cases. The performance on IELTS is better than on MixSet. This can be attributed to the inclusion of the OUTFOX data (English native-speaker student essays) in the training data, while the domains and the generators in MixSet are not in the training set. The low generalization performance suggests challenges in adapting black-box detectors to the diverse domains and generators in real-world applications.

5 Demo Web Application

Our demo web application has two interfaces: (*i*) an interface for fine-grained MGT detection, and (*ii*) a playground for users.

5.1 Automatic Detection

The automatic detection interface is shown in Figure 1 (top). It allows users to input a text, and then the system responds with the class that the text belongs to. To ensure the prediction accuracy, the length of the submitted text is constrained to 50-500 words since the performance of our detectors drops significantly for shorter texts. Longer texts will be truncated, as we are limited by the context window size of the BERT-like transformers we use.

5.2 Human Detector Playground

The demo further offers a human detector playground as an interactive interface, which allows users to test their capability to distinguish between the four text categories. Figure 1 (bottom) shows a snapshot of the playground interface where the users can try the system, gaining insights into the subtle differences between various types of human-written and machine-generated texts.

5.3 Deployment and Implementation

Our demo is deployed on Hugging Face Spaces, which allows seamless integration with transformer models, ease of use, and robust support for hosting machine learning applications. For implementing the user interface, we used Gradio. The code is publicly available under an MIT license.

6 Conclusion and Future Work

In an era of advanced large language models, maintaining the integrity of text poses significant challenges. We presented a system that aims to identify the use of machine-generated text, accurately differentiating human-written text from various types of automatically generated text. Unlike previous work, we use a fine-grained classification schema (Human-Written, Machine-Generated, Machine-Written Machine-Humanized, and Human-Written Machine-Polished), which offers insights into the origins of the text, thus enabling trustworthiness.

In future work, we plan to improve the Domain Adversarial Neural Network (DANN) to improve the results even further. We further plan to explore the possibility of using a DANN on the text's generator instead of the text's domain to generalize detection across different text generators. Using a DANN on both the domain and the generator could potentially lead to a truly universal detector. We also aim to expand the classification to include a fifth category: machine-written and human-edited text, enhancing detection capabilities and providing a more comprehensive analysis of text origins. To further improve the system, we also plan to address potential biases in the dataset caused by formatting styles linked to specific domains, such as Wikihow and PeerRead, to ensure better robustness across a broader range of human-written content. Last but not least, we want to expand the dataset to encompass a diverse set of languages, enabling the development of a robust multilingual detection model.

Limitations

We acknowledge certain limitations of our work, which we plan to address in future work. First, although our work has explored more fine-grained machine-generated text scenarios beyond conventional binary classification, we did not consider a complex scenario where the text is first generated by a machine and then is manually edited by humans to suit their personal needs. This is primarily due to the high costs associated with collecting data that requires human editing.

Moreover, we identified some issues with the dataset. Specifically, some LLMs associate specific domains with particular formatting styles, such as markdown for lists, bullet points, and headers. This issue was particularly noticeable in the Wikihow and PeerRead domains, where the LLMs frequently applied these formatting styles, potentially skewing the data and impacting the accuracy of our classifications. It also remains uncertain whether our system can generalize to detecting models or languages not included in our English-only dataset.

Ethical Statement and Broad Impact

Data License A primary ethical consideration is the data license. We reused pre-existing corpora, such as OUTFOX and Wikipedia, which have been publicly released and approved for research purposes. Moreover, we generated new data on top of the original data, thereby mitigating concerns regarding data licensing.

Biased and Offensive Language Considering that our data is generated by large language models, it might contain offensive or biased language; we did not try to control for this, replying on the inetnal safety mechanisms of the LLMs we used.

Positive Impact of Fine-Grained Detection LLM-DetectAIve expands the conventional binary classification in machine-generated text detection to more fine-grained levels, which is more aligned with real-life scenarios. We believe this approach could be applied in various scenarios, e.g., for students' essays to ensure the originality of their work. Moreover, LLM usage detection may find applications in authorship detection as well as in digital forensics.

References

Anthropic. 2024. The Claude 3 model family: Opus, Sonnet, Haiku.

Evan N Crothers, Nathalie Japkowicz, and Herna L Viktor. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*, 11:70977–71002.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. 2017. Domain-adversarial training of neural networks. In Gabriela Csurka, editor, *Domain Adaptation in Computer Vision Applications*, Advances in Computer Vision and Pattern Recognition, pages 189–209. Springer.

Team Gemini. 2023. Gemini: A family of highly capable multimodal models. *arXiv:2312.11805*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *Proceedings of the International Conference on Learning Representations*.

Jiazhou Ji, Ruizhe Li, Shujun Li, Jie Guo, Weidong Qiu, Zheng Huang, Chiyu Chen, Xiaoyu Jiang, and Xinru Lu. 2024. Detecting machine-generated texts: Not just "AI vs Humans" and explainability is complicated. *arXiv:2406.18259*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv:2310.06825*.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *arXiv:2401.04088*.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. OUTFOX: LLM-generated essay detection through in-context learning with adversarially generated examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21258–21266.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv:1907.11692*.

Team Llama. 2024. The Llama 3 herd of models. *arXiv:2407.21783*.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. MULTITuDE: Large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9960–9987, Singapore.

Dominik Macko, Robert Moro, Adaku Uchendu, Ivan Srba, Jason Samuel Lucas, Michiharu Yamashita, Nafis Irtiza Tripto, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2024. Authorship obfuscation in multilingual machine-generated text detection. *arXiv:2401.07867*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 24950–24962. PMLR.

OpenAI. 2023. GPT-4 technical report. *arXiv:2303.08774*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv:1910.01108*.

Teo Susnjak. 2022. ChatGPT: The end of online exam integrity? *arXiv:2212.09292*.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2024. The science of detecting LLM-generated text. *Communications of the ACM*, 67(4):50–59.

Gemma Team. 2024. Gemma 2: Improving open language models at a practical size. *arXiv:2408.00118*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, and Thomas Arnold. 2024a. SemEval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, pages 2057–2079, Mexico City, Mexico.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4GT-bench: Evaluation benchmark for black-box machine-generated text detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 3964–3992, Bangkok, Thailand.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024c. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1369–1407, St. Julian's, Malta.