Evaluating Style Embeddings for Machine-Generated Text Detection

Noé Durandard[♠,♣], **Saurabh Dhawan**[♢], **Thierry Poibeau**[♣]
[♠] Ecole Normale Supérieure - PSL, Paris, France
[♣] LATTICE, ENS, PSL, U. Sorbonne Nouvelle Paris 3, CNRS, Montrouge, France
[♢] Technical University of Munich, Munich, Germany
noe.durandard@psl.eu, saurabh.dhawan@tum.de, thierry.poibeau@ens.psl.eu

Abstract

In this paper, we evaluate the use of style embeddings for distinguishing machine-generated from human-written text. Style embeddings are particularly suited for this task as compared to semantic embeddings, they offer higher content-independence, and compared to feature-engineering approaches, they offer a richer and more holistic representation of writing style. We use a detection module in which texts are first embedded in high-dimensional stylistic spaces using a style encoder, and the resulting vector representations are classified using supervised methods. To optimize this detector, we evaluate the performance of a range of pre-trained public-domain style encoders paired with different supervised methods. When evaluated on MGTBench, a widely adopted benchmark, our approach matches or exceeds state-of-the-art performance metrics. It also generalizes well across various text domains and LLMs. Our findings highlight the potential, and would facilitate the use, of style embeddings as lightweight and effective components of machine-generated text detection systems.

Keywords: LLM, MGT-Detection, MGTBench, Stylistics, Style-Embeddings, Transferability

1. Introduction

The rapid advancement of large language models (LLM) has made it increasingly difficult to discern machine-generated text (MGT) from human-written text (HWT). To an unaided human reader, the text produced by contemporary LLMs is often indistinguishable from human writing. This has become a crucial concern for a variety of settings such as higher education, journalism and online disinformation, where knowing the true author of a text is crucial for trust and accountability. Accurate detection of AI-generated text has thus become an urgent challenge, motivating research into a range of reliable and generalizable detection techniques (Wu et al., 2025; Tang et al., 2024).

1.1. Why Style Embeddings?

We find that style embeddings present a principled and well-justified approach to MGT detection, as opposed to empirical trial-and-error approaches that just happen to work. There are a variety of reasons that make them particularly well suited. First, a range of studies that contrast the linguistic profiles of MGT and HWT with each other have found consistent and wide ranging stylistic differences between the two (Guo et al., 2023; McGovern et al., 2025; Reinhart et al., 2025; Zanotto and Aroyehun, 2025; Przystalski et al., 2026). Additionally, other studies have shown that LLMs (especially instruction-tuned versions) come to have a stylistically consistent persona that struggles to match human style (Durandard et al., 2025b,a; Reinhart et al., 2025). Linguistic style inherent to a text thus offers a promising point to discern machine-generated from human-written text.

Second, relative to the use of semantic properties, the stylistic properties of language capture the *how* of writing rather than the *what*, that is, they abstract away topical or semantic content. In doing so, they offer relative independence from the widely-varied semantic content of the text, and thus higher domain generalization. This can be tested by contrasting the efficacy of style embeddings to that of semantic embeddings as we present later.

Third, relative to approaches that combine hand-picked sets of stylistic features into feature engineered models (e.g. Kumarage et al., 2023; Opara, 2024; Przystalski et al., 2026), style embeddings jointly encode a much wider range of stylistic features in a high-dimensional space that would allow them to generalize beyond the specific features chosen by researchers. Thus, they are likely to be better suited for detecting the holistic stylistic signatures that separate human from machine-generated text.

Lastly, a range of pre-trained style encoders are available in the public domain e.g. LISA (Patel et al., 2023), Style-Embedding (Wegmann et al., 2022), styledistance (Patel et al., 2025) and Neurobiber (Alkiek et al., 2025), making them easily accessible as an addon to a modular MGT detector.

Hence, stylistic features of a text offer a well-grounded target for finding out if it's machine generated, and style embeddings offer a well-suited lightweight method for capturing this target. However, this potential remains underexplored.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 2619–2628
11–16 May 2026.

1.2. Evaluating Style Embeddings for MGT Detection

The present work addresses this gap by systematically testing the use of 4 major public-domain pre-trained style encoders for MGT detection (LISA, Style-Embedding, styledistance and Neurobiber), each of which is trained with specific methodology and objectives, and has important downstream differences. We also test a semantic encoder (GIST (Solatorio, 2024)) for further comparison.

We employ a detection module in which texts are first embedded in high-dimensional stylistic spaces using a style encoder, and the resulting vector representations are classified using different supervised methods. To optimize this detector, we evaluate the performance of these encoders paired with different supervised methods. In all, we report comparative performance of 15 encoder-classifier pairs (5 encoders $\times$ 3 classifiers) on MGTBench, a widely adopted benchmark (He et al., 2024)[1].

Our contribution is twofold:

1. The best overall detector pairs in our study (`styledistance+lr/mlp`) match or exceed state-of-the-art performance metrics on MGTBench.

2. We further evaluate the transferability of these detector pairs across text domains and models, and show an empirical trade-off between domain- and model-transferability. These transferability metrics offer empirical criteria for matching specific embedding methods to specific MGT detection tasks.

2. Related Works

Early approaches to the problem of MGT detection can be broadly categorized into metric-based and model-based methods (see Tang et al. (2024) and Wu et al. (2025) for extensive reviews of these approaches). Metric-based detectors use statistical cues from the text itself such as perplexity or log-likelihood, out-of-distribution token usage, or using heuristic patterns such as token entropy and rank. These methods don't require extensive training data (which is usually used solely to determine the optimal threshold index between MGT and HWT), but their effectiveness may degrade as MGT distribution patterns further converge toward human-like norms. Additionally, they can often be defeated by simple paraphrasing or slight text perturbations. Model-based detectors, on the other hand, train a classifier (often a transformer-based pre-trained language model such as BERT or RoBERTa, which is then fine-tuned) on labeled examples (e.g. Rojas-Simón et al., 2024; Adilazuarda, 2024). While these can capture more complex features and achieve high in-domain accuracy, they risk overfitting to specific datasets or models. However, the constant evolution of LLMs makes this an ever-changing problem that requires ever newer solutions.

Our approach to this problem is motivated by multiple reports of differences in various stylistic features in AI and Human generated language. For example, differences between HWT and MGT emerge in rhetorical patterns and discourse-level features. Instruction-tuned models adopt an informationally dense, noun-heavy style (Reinhart et al., 2025). On the contrary, Guo et al. (2023) note that HWT integrates more colloquialisms, humor and expressive markers (Guo et al., 2023). Distributional part-of-speech (POS) and morphosyntactic statistics expose important distinctions between MGT and HWT. Both tend to present different POS profiles (Guo et al., 2023; Rosenfeld and Lazebnik, 2024). McGovern et al. (2025) even demonstrates that such features capture persistent "fingerprints" of LLM families (McGovern et al., 2025), symptomatic of MGT's greater grammatical standardization (Przystalski et al., 2026). Structural features are likewise discriminative. HWT typically exhibit simpler dependency structures (Zanotto and Aroyehun, 2025), and, with PRDetect, Li et al. (2025) prove the efficacy of syntax tree representations for MGT detection, even under perturbations (Li et al., 2025).

Finally, several works combine heterogeneous stylistic features into feature-engineered models, including lexis, syntax, n-grams, word- or sentence-level measures (e.g., Kumarage et al., 2023; Opara, 2024; Przystalski et al., 2026), often achieving strong detection performance. Thus, a range of studies demonstrate differences in stylistic elements between HWT and MGT. However, these approaches employ limited sets of hand-crafted stylistic features to capture writing regularities. While effective within narrow domains, such limited feature sets are inherently low-dimensional, constraining their ability to model the full richness of human writing style.

In contrast, transformer-based style encoders are trained end-to-end to learn dense representations of writing style and jointly encode multiple stylistic dimensions. Their capacity to capture multi-dimensional stylistic cue interactions while being content-agnostic makes them a particularly promising candidate for MGT detection. To our knowledge, Soto et al. (2024) offer the only other application of this approach albeit with their own custom-trained style representations for MGT detection. However, while an increasing number of high quality pre-trained style embeddings are now easily available

[1] Source code is available on GitHub at `https://github.com/d-noe/mgt_style_detection`.

in the public domain, their potential as an effective, lightweight component for MGT detection systems has been largely ignored. This study is aimed at addressing this gap with a broad evaluation of the major style encoders by contrasting their performance, on MGT detection and generalization, to each other and to SOTA methods on the widely accepted MGTBench benchmark (He et al., 2024). Thereby, facilitating their use in building real-world MGT detection systems.

3. Experimental Framework

The experiments follow established benchmarking practices, situating our work within prior efforts to evaluate MGT detection under controlled settings. LLM-generated text detection is framed as a binary classification task. In accordance with the Machine-Generated Text detection Benchmark (MGTBench) (He et al., 2024), several evaluation scenarios, consisting of single domain-LLM pairs, are considered. Further, we compute a transferability score to assess the detectors' reliability when applied on texts parting from their training material.

The source code implementing the proposed experimental framework is available on GitHub at https://github.com/d-noe/mgt_style_detection.

3.1. Dataset: MGTBench

MGTBench (He et al., 2024) is built upon pre-existing human-written text datasets and covers three different domains: academic essays (Essay), creative writing (WP) and news articles (Reuters). The MGT counterparts were generated using standard prompt templates with six different LLMs (ChatGLM, Dolly, ChatGPT-turbo, GPT4All, StableLM, and Claude), based respectively on essay instructions, writing prompts, or article headlines.

Within each domain, MGTBench contains 1,000 samples of HWT, and as many[2] machine-generated texts for each LLM. Overall, MGTBench comprises 20,734 texts: 3,000 HWT and 17,734 MGT produced by six LLMs.

Considering each domain-LLM pair as standalone scenarios, MGTBench provides 18 different classification tasks. For the experiments, the data associated with each (domain-LLM) subsetting is split into training and testing samples, using 80% for training and the remaining 20% for evaluation.

[2]Some generation issues are present in approximately one third of the settings, leading to missing data. The largest loss occurs for StableLM in the essay domain (158 samples), while other losses remain marginal.

3.2. Reported Scores

The performances of the detectors are evaluated in terms of F1-score. In addition to in-domain performance, when detectors are trained and tested on the same domain-LLM setting, the methods' transferability is assessed across domains and across LLMs.

3.2.1. In-Domain Performance

The in-domain performance of the detectors are reported both per individual sub-setting and as an average over their performance for each domain-LLM scenario.

The scores are reported in subsection 5.1.

3.2.2. Transferability

The study of the generalization capacity of the detectors is systematized through summarising metrics measuring transferability across domains (T_D) and across LLMs (T_M), as averages of cross-setting F1-score performances. Formally, given a detector, we can compute its domain transferability for each LLM m as the average of non-diagonal elements of the domain transfer matrix (see examples of such matrices in Figure 1):

$$T_D(m) = \frac{1}{|D|(|D|-1)} \sum_{i,j \in D, i \neq j} \theta^m_{i \to j} \qquad (1)$$

where $\theta^m_{i \to j}$ is the performance when trained on domain i and tested on domain j for LLM m. The overall score averages over all LLMs:

$$T_D = \frac{1}{|M|} \sum_{m \in M} T_D(m) \qquad (2)$$

Analogously, swapping the roles of domains and LLMs yields the LLM transferability score T_M.

In other words, these scores aim to capture the average performance when the detectors are evaluated in MGTBench settings that diverge from their training samples.

The transferability scores are reported in subsection 5.2.

4. Detection Methods

We propose to evaluate the representative power of modern pre-trained stylistic encoders for MGT detection. Their performance is put into perspective through comparisons with more traditional semantic encoders, as well as standard baselines from the MGT detection literature.

4.1. Stylistic Encoder-based Detection

The stylistic encoder-based MGT detection modules are framed as combinations of a pre-trained encoder, that embeds input texts into dense representation spaces, with classifiers, that learn to discriminate MGT from HWT based on the vector representations.

4.1.1. Encoders

The experiments presented here rely on the hypothesis that stylistic features alone provide sufficient signal to discriminate MGT from HWT. Accordingly, the core component of the proposed detection modules is a set of pre-trained encoders specifically designed to capture stylistic properties of textual documents. While this line of research is still emerging, several such models have recently been proposed. Here, three representative encoders are selected to embed texts into linguistic style spaces (which then serve as input to the classification step): `neurobiber`[3] (Alkiek et al., 2025), `LISA`[4] (Patel et al., 2023), `Style-Embedding`[5] (Wegmann et al., 2022), and `styledistance`[6] (Patel et al., 2025).

On one hand, `neurobiber` and `LISA` are transformer-based models trained to encode specific stylistic attributes in each output dimensions. As such, `neurobiber` is trained to identify 96 different low-level stylistic features in texts (including Biber's features (Biber, 1991)), while `LISA` also encodes higher level discourse styles through the distillation of individual labelling tasks into an `EncT5` model (Raffel et al., 2020).

On the other hand, `Style-Embeddings` and `styledistance` were developed with a different training paradigm. Both employ contrastive learning objectives that encourage the model to construct representation spaces where texts with similar linguistic style cluster closely, regardless of their content. These methods have proved to be highly efficient to neurally represent style, however, contrary to earlier methods, their outputs lack interpretability in practice.

Finally, `GIST`[7] (Solatorio, 2024), a pre-trained encoder disclosing state-of-the-art performances on semantic tasks, such as MTEB (Muennighoff et al., 2022), is used as a 'control' encoder. In the context of these experiments, it allows to assess the benefits of using stylistic encoders, compared to more traditional semantic encoders, for MGT detection.

Each of these models produce 768-dimensional vectors (except for `neurobiber` which has 96 dimensions), that can readily be used in downstream tasks.

4.1.2. Classification

The vector representations produced by the encoders are then used to train supervised classifiers to distinguish HWT from MGT. In the context of this experiment, the results are reported for three classifiers: ridge logistic regression (`lr`), linear kernel-based support vector classifier (`svc`), and multi-layer perceptron (`mlp`).

The classifiers are trained independently for each sub-setting of MGTBench. However, none of the hyperparameters are specifically optimized, and all classifiers rest on the default parameters and implementations of `scikit-learn` Python library (Pedregosa et al., 2011).

4.2. Baselines

In line with MGTBench procedure, we adopt traditional MGT detection baselines: either based on metrics extracted from pre-trained language models, or using language models fine-tuned for the classification task.

For the first kind of detectors, we adopt and reproduce the classification tasks for three of the best performing metric-based detectors reported for MGTBench: log-likelihood, log-rank, and GLTR. Each of these methods are based on GPT-2[8] language modelling capabilities and derive a score for each text. Log-likelihood and log-rank average token-wise scores assigned respectively as the log probability, or (the log of) the absolute rank of the tokens given their left context. The rationale being that LLMs would produce more probable, or less surprising, texts than humans. GLTR produces a score based on the proportion of tokens that rank within the 10, 100, 1,000 or higher, most probable next-token. For these methods, the threshold to differentiate MGT from HWT is optimized on the individual training sets.

For the latter detector family, i.e. model-based, we report the results both with pre-trained detectors, and fine-tuned ones. For pre-trained detectors, we reproduce the results from MGTBench with the released MGT detectors OpenAI-D[9] (Solaiman et al.,

[3] https://huggingface.co/Blablablab/neurobiber

[4] https://drive.google.com/file/d/12oRf5lJBW6t943fW9jfFp_x9WA7_X9uB/view

[5] https://huggingface.co/AnnaWegmann/Style-Embedding

[6] https://huggingface.co/StyleDistance/styledistance

[7] https://huggingface.co/avsolatorio/GIST-Embedding-v0

[8] https://huggingface.co/openai-community/gpt2-medium

[9] openai-community/roberta-base-openai-detector

2019) and ChatGPT-D[10] (Guo et al., 2023), two RoBERTa-based models trained for MGT detection. Further, to provide fairer comparison standpoints, results for adapted versions of these models (denoted with †), i.e. fine-tuned on the individual training sets, are also reported. Yet, note that this procedure may disrupt the overall behavior of the original detectors and profoundly impact their native capabilities. Finally, in line with the literature, we reproduce the best performing approach on MGTBench: a BERT-like model[11] fine-tuned for MGT/HWT classification, following Ippolito et al. (2020), referred to as LM-D. The fine-tuned detectors LM-D, as well as adapted versions OpenAI-D† and ChatGPT-D†, are trained during three epochs using the AdamW optimizer.

5. Results

The performances of the stylistic encoder-based MGT detection modules are detailed and compared to standard baselines for in-domain performance and generalization capabilities, respectively in subsection 5.1 and subsection 5.2.

The findings outline the effectiveness of recent neural stylistic encoders for MGT detection within constrained in-domain settings. Additionally, best-performing methods achieve strong cross-domain transferability, surpassing alternative approaches, yet reveal lower cross-model performance, struggling to distinguish LLM-generated text when trained on data produced by different models.

5.1. In-Domain Performance

The in-domain evaluation framework outlines the strong performances achieved by style-encoder based MGT detection modules. Table 1 reports the average F1-score over the 18 domain-LLM settings of MGTBench and compares the outcome of stylistic-based methods with standard baselines. Detailed results for each sub-setting are disclosed in Table 2.

Overall, all four tested stylistic encoders, combined with any classifier method, achieve high average F1-scores. It can be observed that the type of classifier used has limited impact here, especially outlining that the non-linearity introduced by `mlp` does not particularly help improve the performance of any of these modules.

Table 1 and Table 2 disclose that the contrastive-based methods achieve the best results, consistently reaching ceiling-level performance. Indeed,

Method	Performance Mean	(std.)
Log-Likelihood	0.820	(0.148)
Log-Rank	0.827	(0.147)
GLTR	<u>0.839</u>	(0.115)
OpenAI-D*	0.659	(0.364)
OpenAI-D†	0.928	(0.030)
ChatGPT-D*	0.609	(0.302)
ChatGPT-D†	0.873	(0.083)
LM-D	<u>0.985</u>	(0.018)
`GIST + lr`	0.921	(0.044)
`GIST + svc`	0.944	(0.038)
`GIST + mlp`	0.948	(0.033)
`neurobiber + lr`	0.925	(0.043)
`neurobiber + svc`	0.921	(0.046)
`neurobiber + mlp`	0.917	(0.051)
`LISA + lr`	0.935	(0.030)
`LISA + svc`	0.940	(0.031)
`LISA + mlp`	0.938	(0.031)
`Style-Embedding + lr`	0.991	(0.008)
`Style-Embedding + svc`	0.992	(0.009)
`Style-Embedding + mlp`	0.990	(0.010)
`styledistance + lr`	**0.994**	(0.007)
`styledistance + svc`	0.992	(0.009)
`styledistance + mlp`	**0.994**	(0.007)

Table 1: In-domain F1-score per detector, averaged across the 18 domain-LLM settings of MGTBench. Best overall performance is presented in bold, best performance per detector family is underlined. * indicates pre-trained MGT detectors directly applied to MGTBench test sets without further training, while † means adapted version further fine-tuned on MGTBench settings.

`Style-Embedding` and `styledistance` yield classification F1-scores higher than 99%, on average over the 18 domain-LLM settings, with each of the tested classifiers. Considering their combination with `lr` classifier for instance, their performance drop below the 99% threshold respectively in only 4 and 3 scenarios out of 18, with lowest performance of 96.7 and 96.9% in WP-StableLM subsetting.

These methods reach better performances than the ones relying on `GIST` —which discloses average scores from 92.1% with `lr` to 94.8% with `mlp`—, outlining the efficacy of relying on stylistic, rather than semantic, features for MGT detection. However, less efficient style encoding methods, `neurobiber` and `LISA`, show low to no improvement compared to `GIST`. Yet, these methods still perform strongly, very rarely exhibiting scores below 90%, and, contrary to earlier methods, their representations could be useful for explainability and interpretability purposes.

In comparison, the baseline methods often display more variable performances with stronger devi-

[10] https://huggingface.co/Hello-SimpleAI/chatgpt-detector-roberta

[11] https://huggingface.co/distilbert/distilbert-base-uncased

Dataset	Method	ChatGLM	Dolly	ChatGPT-turbo	GPT4All	StableLM	Claude	Avg.
Essay	GIST + lr	0.898	0.840	0.920	0.889	0.890	0.856	0.882
	GIST + svc	0.944	0.881	0.939	0.927	0.924	0.905	0.920
	GIST + mlp	0.960	0.902	0.939	0.939	0.939	0.911	0.932
	neurobiber + lr	0.965	0.933	0.949	0.946	0.943	0.902	0.940
	neurobiber + svc	0.972	0.932	0.960	0.943	0.927	0.899	0.939
	neurobiber + mlp	0.963	0.923	0.944	0.951	0.936	0.889	0.934
	lisa + lr	0.978	0.926	0.948	0.944	0.927	0.894	0.936
	lisa + svc	0.985	0.917	0.953	0.959	0.934	0.894	0.940
	lisa + mlp	0.967	0.922	0.942	0.956	0.940	0.919	0.941
	Style-Embedding + lr	0.993	**1.000**	0.990	0.995	0.994	0.987	0.993
	Style-Embedding + svc	0.995	0.997	0.988	0.995	0.994	0.990	0.993
	Style-Embedding + mlp	**0.997**	0.989	0.988	0.995	0.997	0.990	0.993
	styledistance + lr	**0.997**	0.989	0.993	**1.000**	**1.000**	**0.998**	**0.996**
	styledistance + svc	0.995	0.992	**0.995**	**1.000**	0.997	0.993	0.995
	styledistance + mlp	0.993	0.992	**0.995**	**1.000**	**1.000**	**0.998**	**0.996**
WP	GIST + lr	0.966	0.892	0.960	0.943	0.873	0.887	0.920
	GIST + svc	0.983	0.899	0.980	0.965	0.880	0.894	0.933
	GIST + mlp	0.975	0.906	0.980	0.958	0.898	0.889	0.934
	neurobiber + lr	0.957	0.880	0.910	0.945	0.853	0.820	0.894
	neurobiber + svc	0.960	0.870	0.899	0.945	0.850	0.809	0.889
	neurobiber + mlp	0.955	0.860	0.900	0.949	0.839	0.787	0.882
	lisa + lr	0.983	0.941	0.961	0.967	0.899	0.889	0.940
	lisa + svc	0.980	0.922	0.958	0.975	0.910	0.880	0.938
	lisa + mlp	0.978	0.917	0.956	0.967	0.876	0.884	0.930
	Style-Embedding + lr	0.995	0.975	0.990	0.990	0.967	0.990	0.984
	Style-Embedding + svc	**1.000**	0.977	**0.998**	0.990	0.962	0.990	0.986
	Style-Embedding + mlp	0.990	0.975	0.988	**0.992**	0.956	0.992	0.982
	styledistance + lr	0.992	**0.988**	0.993	0.990	0.969	0.997	**0.988**
	styledistance + svc	0.992	0.972	0.990	0.990	0.964	**1.000**	0.985
	styledistance + mlp	0.992	0.977	0.990	**0.992**	**0.974**	0.997	0.987
Reuters	GIST + lr	0.977	0.970	0.947	0.993	0.916	0.956	0.960
	GIST + svc	0.987	0.978	0.960	0.993	0.967	0.985	0.978
	GIST + mlp	0.990	0.978	0.970	0.990	0.967	0.973	0.978
	neurobiber + lr	0.975	0.952	0.911	0.980	0.951	0.869	0.940
	neurobiber + svc	0.970	0.947	0.903	0.980	0.941	0.868	0.935
	neurobiber + mlp	0.980	0.933	0.905	0.980	0.954	0.865	0.936
	lisa + lr	0.956	0.908	0.938	0.973	0.908	0.895	0.930
	lisa + svc	0.975	0.920	0.940	0.978	0.934	0.913	0.943
	lisa + mlp	0.980	0.923	0.938	0.985	0.913	0.922	0.943
	Style-Embedding + lr	**1.000**	0.993	0.990	**0.998**	0.988	**1.000**	0.995
	Style-Embedding + svc	**1.000**	0.993	0.993	**0.998**	0.990	**1.000**	0.995
	Style-Embedding + mlp	0.997	0.988	0.995	0.997	0.985	**1.000**	0.994
	styledistance + lr	0.997	**0.995**	**1.000**	0.995	**0.993**	**1.000**	**0.997**
	styledistance + svc	0.995	**0.995**	**1.000**	0.997	0.990	**1.000**	0.996
	styledistance + mlp	0.997	**0.995**	**1.000**	0.997	0.992	**1.000**	**0.997**

Table 2: MGTBench Results. F1-score performances in binary HWT/MGT classification within different generation domains. Best performance per domain-LLM settings are presented in bold. The 'Avg.' column aggregates the performances across LLMs by presenting the average score for each detection method.

ations across contexts, and sometimes poor results, especially in the case of metric-based systems or non-adapted pre-trained classifiers. Nonetheless, LM-D consistently reaches near-perfect scores, performing on par with the introduced methods. Thus, while the showcased performances do not represent major breakthrough per se, stylistic encoders display a strong ability to discriminate MGT from HWT based on pre-trained encoders reliably (see also the lower performance variance exhibited by the best performing models).

Altogether, these results demonstrate that modern stylistic encoders alone suffice for highly reliable in-domain supervised LLM text detection on MGTBench. Yet, besides highlighting the effectiveness of these methods, it also points out the potential saturation of this benchmark under such settings.

5.2. Detectors Transferability

The transferability performances of the detectors, as described in subsubsection 3.2.2, are presented in Figure 2, which discloses the domain transferability T_D of each tested detectors against their model transferability T_M.

Interestingly, this figure seems to outline an empirical trade-off between domain- and model-transferability. For instance `styledistance + lr` exhibits the highest domain transferability

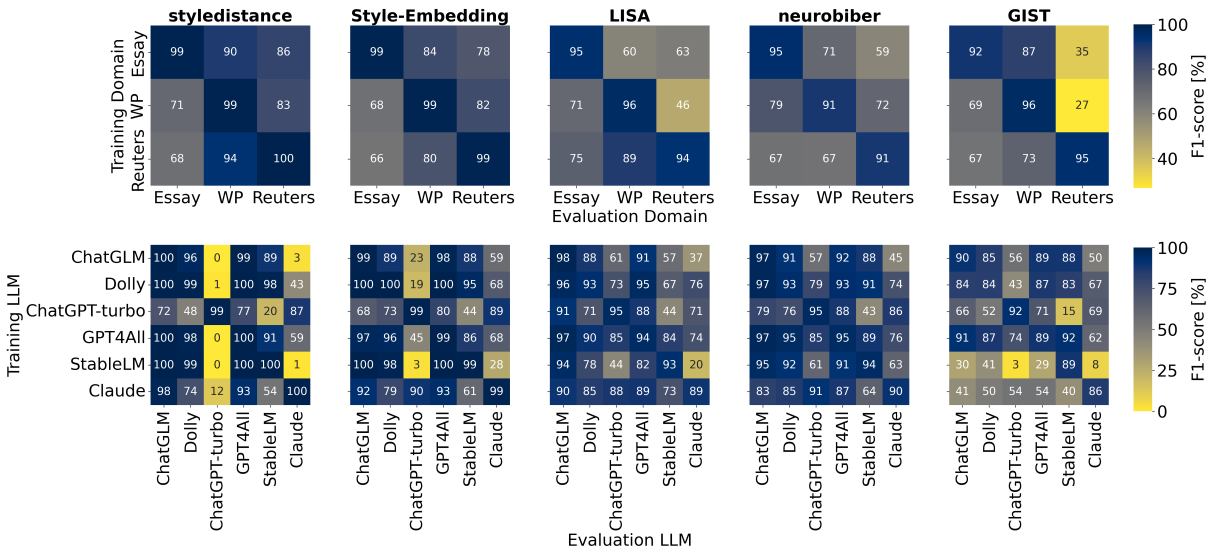

Figure 1: Detectors transferability case-study. F1-scores (%) of cross-setting training/evaluation experiments. Each column is associated with a detector (here, an encoder associated with `lr` classifier). The first row is associated with transfer domain for text generated with a single LLM (ChatGPT-turbo), and the second row illustrates LLM-transferability within the Essay domain. The y-axis represents the training setting, and the x-axis is the evaluation one.

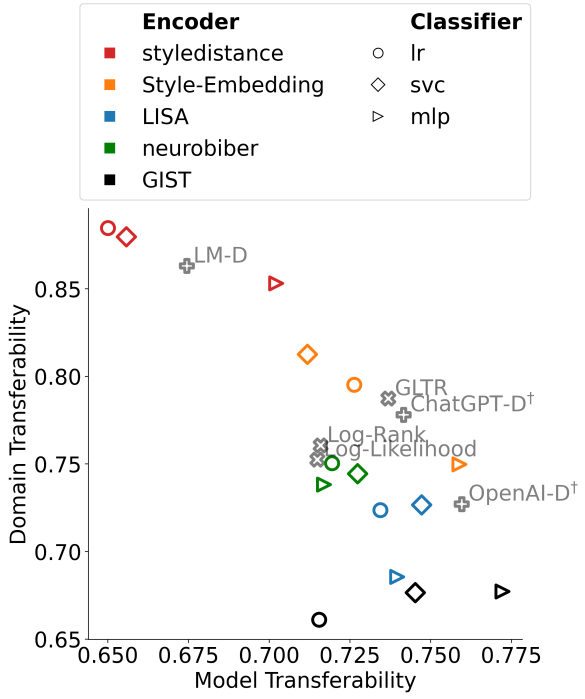

Figure 2: Domain Transferability plotted against Model Transferability for the different detectors. Colors refer to the encoder, while markers indicate the classifier. Baselines are annotated in transparency, crosses are used for metric-based baselines and pluses for PLM-based detectors, † indicate adapted methods.

(reaching a score of 0.885), but the lowest model-transferability (0.650). Reversely, `GIST + mlp` reaches the highest model transferability (0.772) but shows poor transferability across domains (0.677). Based on MGTBench data, these results imply that `styledistance`-based detectors can be operational when trained on texts generated by the same LLM but in different domains, while, to a lower extent, methods relying on `GIST + mlp`, or adapted OpenAI-D†, should be preferred when the domain is fixed but different LLMs could be used for generation. These results are further illustrated in the transfer matrices of the case-studies, with fixed LLM (ChatGPT-turbo), or fixed domain (Essay), shown in Figure 1 for encoders combined with `lr` classifier.

While further investigations would be required to validate the following hypotheses, these findings may reflect earlier findings: style-informed methods are able to capture LLMs 'fingerprints' which are stable across domains but different from one LLM family to another (McGovern et al., 2025), while semantic encoders are left clueless when changing domains, but may catch on different word choices or discourse framing that are specific to human voices (Guo et al., 2023).

6. Conclusion

In this study, we argue that stylistic properties of a text offer a well-grounded target for finding out if it's machine generated, and style embeddings offer a well-suited lightweight method for capturing

this target. We use a detection module in which texts are first embedded in high-dimensional stylistic spaces using a style encoder, and the resulting vector representations are classified using supervised methods. To optimize this detector, we evaluate the performance of a range of pre-trained public-domain style encoders on MGTBench, a widely adopted benchmark. This framework allows us to do a comparative study of different style encoders to each other, and to other SOTA methods. The best detector pair (`styledistance+lr/mlp`) in our approach matches or exceeds state-of-the-art performance metrics on MGTBench. We also find an empirical trade-off between domain- and model-transferability, which offers empirical criteria for matching specific embedding methods to specific MGT detection tasks. Altogether, these results show that modern style encoders alone suffice for highly reliable (as measured on MGTBench) supervised LLM text detection. However, given the fast evolving domain of LLMs, and the repertoire of tricks available for evading detection, we recommend that real-world MGT detection systems should be built in a modular manner using multiple approaches. The present study offers data and metrics to show that style embeddings can certainly form a lightweight and effective component of such a modular system.

Limitations

While using MGTBench and the binary MGT/HWT classification as a testbed offers a widely adopted framework and easily comparable evaluation framework, relying solely on this setup, and on a deliberately controlled scope, inevitably introduces several limitations that may affect ecological validity.

First, the evaluation protocol remains largely artificial. It relies on fully separated human-written and LLM-generated samples. It does not consider adversarial editing, mixed authorship, nor model evolution, factors that are likely to be present in real-world scenarios. Second, the laboratory conditions afforded by the supervised classification framework may not entirely reflect practical detection scenarios. Indeed, the generated texts may lack diversity, considering, for instance, that all MGT texts were generated from a single prompt template per domain. Thus, it would be informative to study how performance would be impacted under distributional shifts, in additions and complementary to the transferability analysis.

The combination of these factors (and potentially others) may contribute to the ceiling-level performances consistently achieved by the best performing detectors, suggesting that the benchmark is saturated under this classification paradigm. While our results demonstrate strong within-domain, within-LLM performance, we suggest that style embeddings are best viewed as lightweight components to be included in modular, multi-method MGT detection systems. We doubt that any single benchmarked approach can maintain such performance by their own in real-world settings given the complexity and variability of actual use cases.

Finally, beyond more benchmark-related aspects, the use of transformer-based style encoders limits interpretability compared to earlier methods that leveraged handcrafted stylistic features. Except for `LISA`, which is specifically interpretability-oriented, and `neurobiber`, the representations produced by `Style-Embedding` and `styledistance` (which also yields the best performances) remain opaque, making it difficult to characterize the stylistic cues used by the detectors, and whether they reflect general stylistic traits or dataset-specific regularities.

Acknowledgements

Parts of this research were carried within the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 945304 - Cofund AI4theSciences hosted by PSL University.

This work benefited from funding from the French State, managed by the Agence Nationale de la Recherche, under the France 2030 program (grant reference ANR-23-IACL-0008).

SD was supported by a grant from IEAI-TUM.

7. Bibliographical References

Muhammad Adilazuarda. 2024. Beyond Turing: A comparative analysis of approaches for detecting machine-generated text. In *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*, pages 1–12, Mexico City, Mexico. Association for Computational Linguistics.

Kenan Alkiek, Anna Wegmann, Jian Zhu, and David Jurgens. 2025. Neurobiber: Fast and interpretable stylistic feature extraction.

Douglas Biber. 1991. *Variation across speech and writing*. Cambridge university press.

Noé Durandard, Saurabh Dhawan, and Thierry Poibeau. 2025a. Language style matching in large language models. In *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 620–636, Avignon, France. Association for Computational Linguistics.

Noé Durandard, Saurabh Dhawan, and Thierry Poibeau. 2025b. LLMs stick to the point, humans to style: Semantic and stylistic alignment in human and LLM communication. In *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 206–213, Avignon, France. Association for Computational Linguistics.

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

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. Mgtbench: Benchmarking machine-generated text detection. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, page 2251–2265, New York, NY, USA. Association for Computing Machinery.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Tharindu Kumarage, Joshua Garland, Amrita Bhattacharjee, Kirill Trapeznikov, Scott Ruston, and Huan Liu. 2023. Stylometric detection of ai-generated text in twitter timelines.

Xiang Li, Zhiyi Yin, Hexiang Tan, Shaoling Jing, Du Su, Yi Cheng, Huawei Shen, and Fei Sun. 2025. PRDetect: Perturbation-robust LLM-generated text detection based on syntax tree. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 8290–8301, Albuquerque, New Mexico. Association for Computational Linguistics.

Hope Elizabeth McGovern, Rickard Stureborg, Yoshi Suhara, and Dimitris Alikaniotis. 2025. Your large language models are leaving fingerprints. In *Proceedings of the 1stWorkshop on GenAI Content Detection (GenAIDetect)*, pages 85–95, Abu Dhabi, UAE. International Conference on Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Chidimma Opara. 2024. Styloai: Distinguishing ai-generated content with stylometric analysis. In *International conference on artificial intelligence in education*, pages 105–114. Springer.

Ajay Patel, Delip Rao, Ansh Kothary, Kathleen McKeown, and Chris Callison-Burch. 2023. Learning interpretable style embeddings via prompting LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15270–15290, Singapore. Association for Computational Linguistics.

Ajay Patel, Jiacheng Zhu, Justin Qiu, Zachary Horvitz, Marianna Apidianaki, Kathleen McKeown, and Chris Callison-Burch. 2025. StyleDistance: Stronger content-independent style embeddings with synthetic parallel examples. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8662–8685, Albuquerque, New Mexico. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Karol Przystalski, Jan K. Argasiński, Iwona Grabska-Gradzińska, and Jeremi K. Ochab. 2026. Stylometry recognizes human and llm-generated texts in short samples. *Expert Systems with Applications*, 296:129001.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Alex Reinhart, Ben Markey, Michael Laudenbach, Kachatad Pantusen, Ronald Yurko, Gordon Weinberg, and David West Brown. 2025. Do llms write like humans? variation in grammatical and rhetorical styles. *Proceedings of the National Academy of Sciences*, 122(8):e2422455122.

Jonathan Rojas-Simón, Yulia Ledeneva, and René Arnulfo García-Hernández. 2024. Classification of human and machine-generated texts using lexical features and supervised/unsupervised machine learning algorithms. In *Pattern Recognition*, pages 331–341, Cham. Springer Nature Switzerland.

Ariel Rosenfeld and Teddy Lazebnik. 2024. Whose llm is it anyway? linguistic comparison and llm attribution for gpt-3.5, gpt-4 and bard. *arXiv preprint arXiv:2402.14533*.

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. 2019. Release strategies and the social impacts of language models.

Aivin V. Solatorio. 2024. Gistembed: Guided in-sample selection of training negatives for text embedding fine-tuning.

Rafael Alberto Rivera Soto, Kailin Koch, Aleem Khan, Barry Y. Chen, Marcus Bishop, and Nicholas Andrews. 2024. Few-shot detection of machine-generated text using style representations. In *The Twelfth International Conference on Learning Representations*.

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

Anna Wegmann, Marijn Schraagen, and Dong Nguyen. 2022. Same author or just same topic? towards content-independent style representations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 249–268, Dublin, Ireland. Association for Computational Linguistics.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025. A survey on LLM-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338.

Sergio E. Zanotto and Segun Aroyehun. 2025. Linguistic and embedding-based profiling of texts generated by humans and large language models.

8. Language Resource References

Kenan Alkiek and Anna Wegmann and Jian Zhu and David Jurgens. 2025. *Neurobiber: Fast and Interpretable Stylistic Feature Extraction*. PID https://huggingface.co/Blablablab/neurobiber. No PID or ISLRN available; resource accessed on October 2025.

He, Xinlei and Shen, Xinyue and Chen, Zeyuan and Backes, Michael and Zhang, Yang. 2024. *MGTBench: Benchmarking Machine-Generated Text Detection*. Association for Computing Machinery, CCS '24. PID https://github.com/xinleihe/MGTBench. No PID or ISLRN available; resource accessed on August 2025.

Patel, Ajay and Rao, Delip and Kothary, Ansh and McKeown, Kathleen and Callison-Burch, Chris. 2023. *Learning Interpretable Style Embeddings via Prompting LLMs*. Association for Computational Linguistics. PID https://drive.google.com/file/d/12oRf5lJBW6t943fW9jfFp_x9WA7_X9uB/view. No PID or ISLRN available; resource accessed on September 2025.

Patel, Ajay and Zhu, Jiacheng and Qiu, Justin and Horvitz, Zachary and Apidianaki, Marianna and McKeown, Kathleen and Callison-Burch, Chris. 2025. *StyleDistance: Stronger Content-Independent Style Embeddings with Synthetic Parallel Examples*. Association for Computational Linguistics. PID https://huggingface.co/StyleDistance/styledistance. No PID or ISLRN available; resource accessed on October 2025.

Aivin V. Solatorio. 2024. *GISTEmbed: Guided In-sample Selection of Training Negatives for Text Embedding Fine-tuning*. PID https://huggingface.co/avsolatorio/GIST-Embedding-v0. No PID or ISLRN available; resource accessed on October 2025.

Wegmann, Anna and Schraagen, Marijn and Nguyen, Dong. 2022. *Same Author or Just Same Topic? Towards Content-Independent Style Representations*. Association for Computational Linguistics. PID https://huggingface.co/AnnaWegmann/Style-Embedding. No PID or ISLRN available; resource accessed on October 2025.