SENTRA: A General Purpose Encoder for LLM Text Detection

Anonymous ACL submission

Abstract

LLMs are becoming increasingly capable and widespread. Consequently, the potential and reality of their misuse is also growing. In this work, we address the problem of detecting LLM-generated text that is not explicitly declared as such. We present a novel, general-purpose, and supervised LLM text detector, *SElected-Next-Token tRAnsformer (SENTRA)*. SENTRA is a Transformer-based encoder leveraging selected-next-token-probability sequences and utilizing contrastive pre-training on large amounts of unlabeled data. Our experiments on three popular public datasets across 24 domains of text demonstrate SENTRA is a general-purpose classifier that significantly outperforms several popular baselines in the out-of-domain setting.

1 Introduction

The problem of determining whether a text has been generated by an LLM or written by a human has been widely studied in both academia (Tang et al., 2024) and industry. Several commercial-level automated text detection systems have been developed, including GPTZero (Tian and Cui, 2023), Originality (Originality.AI, 2025), Sapling (Sapling AI, 2025), and Reality Defender (Reality Defender, 2025). Although significant progress has been made in detecting LLM-generated text over the past several years, these systems remain far from perfect and are often unreliable. A major limitation is their brittleness: they can perform well on certain types of LLM-generated text but fail catastrophically in other cases (Dugan et al., 2024). This issue is particularly pronounced when operating in a real world scenario, where models must handle out-of-domain (OOD) data, different LLM generators, or various LLM "attacks" (Dugan et al., 2024; Zhou et al., 2024). Therefore, it is crucial to develop more generalizable methods that deliver reliable performance across these settings. Because the space of possible domains is much larger than the number of known LLM generators or attacks, this work focuses on generalization to unseen domains since this type of generalization constitutes one of the most crucial issues facing the LLM text detectors.

The probability assigned by an LLM to a document can be measured by auto-regressively feeding the document's tokens into the LLM and observing the predicted probabilities for each token. This process produces a sequence of values that we denote as selected-next-token-probabilities (SNTP). SNTP have been extensively used in prior work on LLM-generated text detection (Guo et al., 2023; Hans et al., 2024; Verma et al., 2024). These prior works primarily rely on either heuristics (handcrafted functions) applied to SNTP sequences or linear models trained on expert-derived features (Hans et al., 2024; Verma et al., 2024). In contrast, our approach encodes SNTP sequences using a Transformer model pre-trained on unlabeled data, leveraging the expressivity of Transformers to directly learn a representation of the probability that a single or a pair of LLMs assign to tokens in a document. In this paper, we propose *SElected-Next-Token tRAnsformer (SENTRA)*, a method for detecting LLM-generated text that directly learns a detection function in a supervised manner from SNTP sequences. This method utilizes a novel Transformer-based architecture with a contrastive pre-training mechanism. The learned representation can be fine-tuned on labeled data to create a supervised model that distinguishes LLM-generated texts from human-written texts.

For the LLM-text-detection task, supervised detectors have been shown to generalize poorly outside the training distribution (Dugan et al., 2024). Our SENTRA network addresses this issue by learning generalizable functions on SNTP. We show empirically that the supervised method presented in this paper generalizes to unseen do-

mains better than both supervised and unsupervised baselines by leveraging our proposed Transformer-based architecture, thus demonstrating greater generalization to distribution shifts.

In this paper, we demonstrate:

- Detectors utilizing SENTRA as their encoder *generalize* well to domains outside of the training distribution(s).
- Contrastive pre-training of SENTRA leads to *improved generalization* results on new domains.
- SENTRA outperforms all studied baselines in out-of-domain evaluations on three widely used benchmark datasets.

2 Related Work

With the rise of LLMs, significant research has been conducted on accurately detecting text generated by these models (Tang et al., 2024). At a high level, these detectors can be categorized into three approaches: watermarking, unsupervised (or zero-shot) detection, and supervised detection. Watermarking generally relies on the LLM deliberately embedding identifiable traces in its output (Liu et al., 2025). In this work, we focus on the general detection problem, including cases involving non-cooperative LLMs; therefore, we do not consider watermarking as a point of comparison. Unsupervised methods typically leverage metrics computed by an LLM on the target document. These methods can be further divided into white-box detection, where the candidate LLM is known (Mitchell et al., 2023), and black-box detection, where the candidate LLM is unknown (Tang et al., 2024). Given our focus on the general detection problem, we prioritize black-box detection methods. Supervised methods, on the other hand, involve collecting a corpus of human-written and LLM-generated text samples, which are then used to train the detection models (Verma et al., 2024; Soto et al., 2024).

Selected-next-token-probabilities (SNTP) have been widely used for LLM detection in both white and black box settings (Guo et al., 2023; Hans et al., 2024; Verma et al., 2024). Perplexity (Jelinek et al., 1977) is a commonly used metric to evaluate an LLM's ability to model a given dataset. In the context of AI detection, a lower perplexity score on a document indicates an LLM "fits" a document and this indicates a higher likelihood the document was LLM-generated. Conversely, a higher perplexity score suggests the LLM's probability model does not fit or accurately represent the candidate text, implying a lower likelihood that the text was generated by the LLM (Guo et al., 2023).

Some detectors use multiple sequences of STNP for the detection task (Verma et al., 2024; Hans et al., 2024). Verma et al. (2024) leveraged SNTPs from two Markov models, along with an LLM's SNTP, extracted features, and a forward feature selection scheme as inputs to a linear classifier. In contrast to Guo et al. (2023), Hans et al. (2024) argued that relying solely on the perplexity score for LLM-generated content detection can be misleading. Although human-authored text generally results in higher perplexity, prompts can significantly influence perplexity values. The authors highlighted the "capybara problem", where the absence of a prompt can cause an LLM-generated response to have higher perplexity, leading to false detections. They addressed this issue by introducing *cross-perplexity* as a normalizing factor to calibrate for prompts that yield high perplexity. GLTR (Gehrmann et al., 2019) is a detection method that leverages SNTP along with other metrics, such as the rank of the selected word within the next-token distribution and the entropy of the next-token distribution (Gehrmann et al., 2019). These metrics target LLM decoding strategies, including greedy decoding, top-k sampling, and beam search.

DetectGPT is an unsupervised method based on the idea that texts generated by LLMs tend to "occupy negative curvature regions of the model's log probability function" (Mitchell et al., 2023). The method generates perturbations of the sample text using a smaller model and compares the log probability of the sample text to that of the perturbations. Fast-DetectGPT replaces the perturbations in DetectGPT with a more efficient sampling step (Bao et al., 2024). Nguyen-Son et al. (2024) observed that the similarity between a sample and its counterpart generation is notably higher than the similarity between the counterpart and another independent regeneration. They demonstrated that this difference in similarity is useful for detection. Other works (Hao et al., 2024) have also explored the idea of "rewriting" text using LLMs to aid detection methods. In their study, they trained an LLM to maximize the edit distance from rewriting human-written texts while minimizing the edit distance from rewriting LLM-generated texts.

The most common supervised baseline for LLM-generated text detection is a RoBERTa classifier (Liu et al., 2019) trained on a corpus of labeled

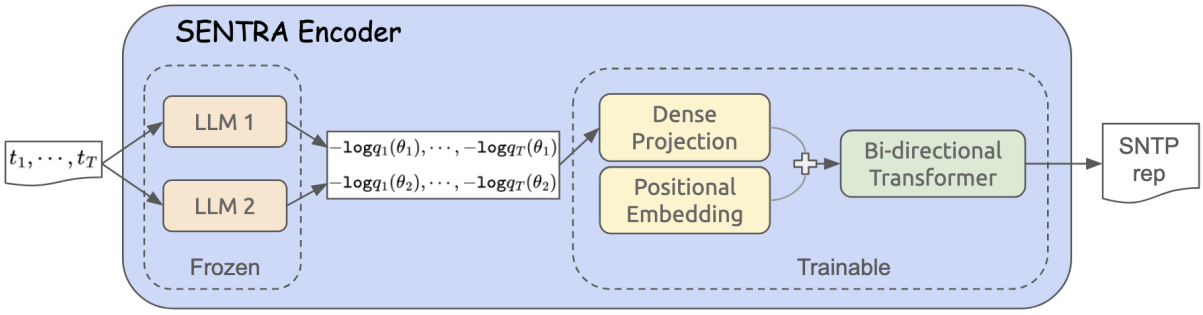

Figure 1: SENTRA leverages the selected-next-token-probabilities from two frozen LLMs. These two sequences of logits are concatenated into a vector. Each of these vectors are projected to the dimension of the bi-directional transformer.

text, where each document is marked as either human-written or LLM-generated. Several studies have expanded on this approach to supervised text-based classification. Yu et al. (2024) trained a feed-forward classifier with two hidden layers using intrinsic features derived from Transformer hidden states, determined via KL-divergence. Tian et al. (2024) address the challenge of detecting short texts by treating short samples in the training corpus as partially "unlabeled". Hu et al. (2023) employed adversarial learning to enhance the robustness of their RoBERTa-based classifier against paraphrase attacks.

Several publications have explored contrastive training for the LLM detection task (Bhattacharjee et al., 2023, 2024; Soto et al., 2024; Guo et al., 2024). These studies use contrastive pre-training for a text transformer, which is chosen to be RoBERTa (Liu et al., 2019) in many cases, to guide the network toward a representation more useful for LLM-generated text detection. Furthermore, many prior contrastive training strategies focus on identifying stylometric features (Soto et al., 2024; Guo et al., 2024), while other studies extract stylometric features directly and train classifiers using those features (Kumarage et al., 2023a). Rather than focusing on text representations, our method is primarily designed to produce useful SNTP representations and, thus, proposes a different contrastive pre-training scheme, one that compares textual representations with those of the SNTP transformer.

However, SNTP and supervised methods have been shown, both intuitively and empirically, to struggle with generalization to unseen domains (Li et al., 2024; Roussinov et al., 2025). This challenge has led to a series of studies aiming at improving generalization. For instance, Lai et al. (2024) applied adaptive ensemble algorithms to enhance model performance in OOD scenario. Meanwhile, Guo et al. (2024) and Soto et al. (2024), recognizing the limited number of widely adopted general-purpose AI assistants, proposed to train an embedding model to learn the writing style of LLMs, and thereby improving the detection accuracy.

Prior work has shown SNTP to be an effective input for identifying LLM generated text (Guo et al., 2023; Hans et al., 2024; Verma et al., 2024), but they rely on relatively simple metrics or heuristics. In this work, we show Transformer networks, specifically SENTRA, can learn a representation of SNTP sequences that can be used to train detection models that better generalize to unseen domains.

3 Methodology

3.1 Overview of Our Method: SENTRA

Consider a document t consisting of an input sequence of T tokens $t = (t_1, t_2, \cdots, t_T)$. Assuming an LLM has parameters θ, the probability of document t given this LLM can be specified as

$$P(t_1, t_2, \cdots, t_T|\theta) = \prod_{t=1}^{T} q_i(\theta), \qquad (1)$$

where

$$q_i(\theta) = P(t_i \mid t_1, t_2, \cdots, t_{i-1}; \theta) \qquad (2)$$

is the probability of token t_i, given the preceding tokens $(t_1, t_2, \cdots, t_{i-1})$. We denote the observed sequence of selected-next-token-probabilities (SNTP) as

$$q(\theta) = (q_1(\theta), q_2(\theta), \cdots, q_T(\theta)). \qquad (3)$$

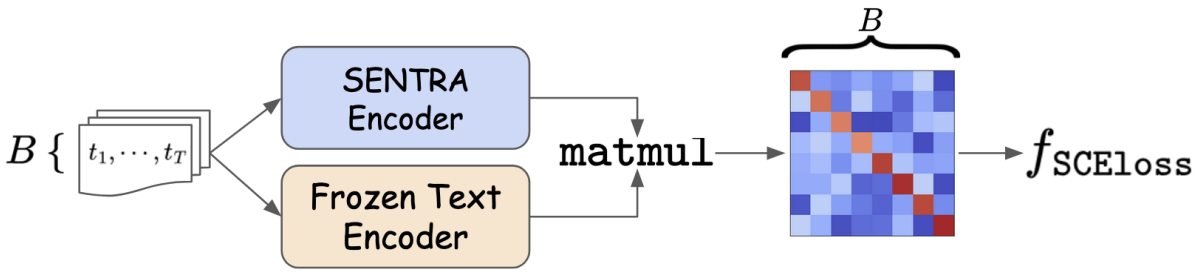

Figure 2: Pre-training: the outputs of SENTRA and a frozen text encoder go through linear layers, (W_s and W_t) respectively, and normalization before a matrix multiplication (`matmul`) operation to produce the similarity matrix M.

Prior work has crafted or discovered heuristic functions on these sequences that are useful in detecting LLM-generated text (Guo et al., 2023; Hans et al., 2024). SENTRA replaces these heuristic functions on SNTP sequence(s) with a neural network.

Figure 1 illustrates our proposed method. We leverage two LLMs with parameters θ_1 and θ_2 to produce SNTP sequences q_1 and q_2 for a candidate document with tokens t. The two sequences are concatenated to form a tensor of size $[T, 2]$.

Instead of token embeddings often seen in Transformer architectures (Devlin et al., 2019), each tensor slice of size two is independently projected to an embedding dimension D using a fully connected layer. This transformation results in a tensor of size $[T, D]$ for a single document. We then insert a learned `[CLS]` tensor at the first position, extending the sequence to size $[T+1, D]$. Learned positional embeddings are added to each vector before passing the embedded sequence through a bi-directional Transformer (Devlin et al., 2019), producing a representation of size $[T+1, D]$.

The output of SENTRA is a learned representation over SNTP, capturing the probability assigned by two LLMs to the tokens in a document. For classification, we use the representation at the `[CLS]` token and append a classification head.

In summary, our approach employs a Transformer-based encoder to systematically learn a useful representation of SNTP sequences. Similar to many Transformer-based approaches (Devlin et al., 2019; Radford et al., 2021), we demonstrate in Section 3.2 that our method can leverage large quantities of unlabeled data to enhance this learned representation.

3.2 SENTRA Contrastive Pre-Training

We further explore the pre-training of SENTRA using unlabeled text data and find that it significantly improves detection performance, see Section 4.3. Figure 2 illustrates our concept for pre-training SENTRA. We elected to leverage a mode of information with many available pre-trained representations, text, to help pre-train SENTRA which leverages a new mode of information, SNTP. A document is encoded using both a pre-trained text encoder (Devlin et al., 2019; Liu et al., 2019) and our SENTRA network, producing representations R_t and R_s. These representations are projected to a joint embedding space, U_e and S_e, using fully connected layers W_t and W_s for the text and SNTP representations respectively.

$$\begin{aligned} U_e &= W_t(R_t) \\ S_e &= W_s(R_s) \end{aligned} \tag{4}$$

After applying L2 normalization to U_e and S_e to control for scaling, we then compute a comparison matrix M

$$M = (U_e S_e^T) e^r \tag{5}$$

where r is learned temperature scalar.

The two encoders learn to match representations of the same document within a batch B. Employing the contrastive learning objective, we then minimize the cross-entropy loss over the columns (text-to-SNTP), and rows (SNTP-to-text) of the comparison matrix M, using the ground truth text-SNTP pairings in the batch, $y = 0, 1, ...B$.

The pre-training scheme effectively enables SENTRA to produce representations that align with those generated by the frozen text encoder, thereby yielding more useful representations of the q_1 and q_2 sequences.

Notably, this pre-training scheme is reminiscent of CLIP (Radford et al., 2021). In their work, the

Dataset	Size	Domains	LLMs	Attks	A.Tokens	% LLM-Gen	A.Train	A.Val	A.Test
RAID	500,000	8	11	11	712	97.16%	22,398	2,488	62,500
M4GT	267,863	6	14	0	471	67.6%	97,584	10,893	33,482
MAGE	430,630	10	27	0	267	34.86%	167,972	50,387	5,682

Table 1: Overview of datasets used in the study. Attks is the number of attacks included in the dataset. A.Tokens is the average token length using the Falcon 1 tokenizer. A.Train, A.Val, and A.Test are the average train, validation, test set sizes across all domain splits. The train and validation datasets are class balanced.

authors jointly trained text and image encoders from scratch. Unlike CLIP, which deals with text and images, we focus solely on text and on pre-training only the SENTRA SNTP encoder. To do this, we freeze a pre-trained text encoder and train only SENTRA and the contrastive embedding projections.

3.3 Implementation

We implement our SENTRA model with eight attention heads, eight layers, and a hidden dimension of 768 for a total of 57M parameters. The Transformer architecture and positional embeddings follow the same definitions as in BERT (Devlin et al., 2019). Before pre-training, the SENTRA parameters are randomly initialized. The frozen text encoder used for contrastive pre-training is initialized from RoBERTa-base (Liu et al., 2019). SENTRA is pre-trained on a 600K sample of Common Crawl data from RedPajama (Weber et al., 2024). Pre-training is conducted for 20 epochs with a batch size of 256 and a maximum token length of 64. We then continue contrastive training for 10 epochs with a batch size of 128 and a maximum token length of 512 to pre-train the later position embeddings. The peak learning rate was set to $1e-4$ for both phases. We use the AdamW (Loshchilov and Hutter, 2019) optimizer with a weight decay of $1e-2$ and set the contrastive learning temperature to 0.007 (Chen et al., 2020). During fine-tuning, we initialize SENTRA from the pre-trained model, use a learning rate of $1e-4$, a weight decay of $1e-2$, and apply early stopping with a patience of two epochs on a validation dataset.

The SENTRA encoder leverages two frozen LLMs to produce sequences of SNTP. Following Binoculars (Hans et al., 2024), we use Falcon-7B [1] and Falcon-7B-Instruct [2] (Almazrouei et al., 2023) to produce these sequences. We used a sequence of two SNTP because Binoculars showed it is useful for the detector to compare both SNTP, and we used the Falcon models specifically because Binoculars showed they worked well (Hans et al., 2024).Thus far, we have described SENTRA's inputs as sequences of selected-next-token-probabilities (SNTP). More precisely, we use sequences of cross-entropy loss values produced by the LLMs for a given candidate text t. The probabilities can be recovered from those loss values as $q_i(\theta) = exp(-l_i(\theta))$, where l_i is the loss value for token t_i. During SENTRA training, the SNTP sequences are precomputed and cached. At inference, the computational complexity is dominated by the Falcon models. Because we use the same LLMs as Binoculars (Hans et al., 2024) and our SENTRA encoder is small, our method has the same order of complexity as Binoculars. See Appendix B for additional details.

4 Experiments

4.1 Datasets

If we define text similar to the training data distribution as in-domain and text that is dissimilar as out-of-domain, it is well established supervised LLM detection methods perform significantly better in-domain than out-of-domain (Dugan et al., 2024). However, a model designed for LLM-generated text detection in real world scenarios will inevitably encounter out-of-domain texts. For this reason, this work focuses on *out-of-domain experiments*, where key subsets of data are withheld from the training dataset.

To evaluate the effectiveness of our proposed method, we used three publicly available datasets: RAID (Dugan et al., 2024), M4GT (Wang et al., 2024a) and MAGE (Li et al., 2024), focusing exclusively on English-language data.

RAID The full RAID dataset contains over 6 million human- and LLM-generated texts spanning 8 domains, 11 LLM models, multiple decoding strategies, penalties, and 11 adversarial attack types. We down-sampled it to 500K instances before per-

[1] https://huggingface.co/tiiuae/falcon-7b

[2] https://huggingface.co/tiiuae/falcon-7b-instruct

forming out-of-domain split sampling. With the included attacks, the RAID dataset also assesses the effectiveness of different supervised baseline methods against adversarial attacks under the in-attack setup.

M4GT An extension of M4 (Wang et al., 2024b), the M4GT dataset is a multi-domain and multi-LLM-generator corpus comprising data from 6 domains, 9 LLMs, and 3 different detection tasks.

MAGE The MAGE dataset covers 10 content domains, with data generated by 27 LLMs using 3 different prompting strategies. It is specifically designed to assess out-of-distribution generalization capability. We use the "Unseen Domains" evaluation from (Li et al., 2024).

Each dataset is further split into training, validation and test sets. For MAGE, we used the published split. To mitigate the label imbalance problem, the train and validation splits are balance-sampled to ensure an equal number of human- and LLM-generated texts. This was achieved by down-sampling the majority class to match the size of the minority class within split. Addressing this imbalance is crucial for two reasons: 1) the percentage of LLM-generated text is over 97% in the RAID dataset by design; 2) across the three datasets, the proportion of LLM-generated text varies significantly. The average train and validation set sizes show how much data went into the training of the supervised methods while ensuring class balance, providing a clear comparison to the total dataset size. The MAGE dataset has significantly shorter texts and this adds difficulties in the detection task (Tian et al., 2024; Fraser et al., 2024). Table 1 contains detailed statistics on the evaluation datasets.

We use the first 512 tokens from the datasets across all methods and baselines.

4.2 Baseline Methods

We evaluated and compared the performance of our approach against multiple existing methods, including zero-shot, embedding-based, and supervised detectors. For zero-shot detectors, we selected **perplexity** (Guo et al., 2023), **Fast-DetectGPT** (Bao et al., 2024), and **Binoculars** (Hans et al., 2024). For embedding-based detectors, we selected **UAR** (Soto et al., 2024) and evaluated both its Multi-LLM and Multi-domain models. For supervised detectors, we chose **RoBERTa-base** (Liu et al., 2019) with direct fine-tuning, **Ghostbuster** (Verma et al., 2024) which trains a logistic regression classifier on forward-selected crafted log-probability features, and **Text Fluoroscopy** (Yu et al., 2024) which utilizes intrinsic features. For RoBERTa, we used the same settings as the fine-tuning of SENTRA: a learning rate of $1e-4$, a weight decay of $1e-2$, and a patience of two epochs.

We used Falcon-7B and Falcon-7B-Instruct across all baseline methods that required LLMs, except for Fast-DetectGPT where we followed its black-box setting. Appendix C provides a detailed description of the setup, assumptions and modifications made for each baseline method.

We compared aforementioned baseline methods with our proposed methods. We present results from two SENTRA encoder variations, R-SENTRA and SENTRA. R-SENTRA has all non-LLM weights in SENTRA encoder initialized at random (without pre-training), whereas the full SENTRA model has those weights pre-trained on RedPajama data (Weber et al., 2024), as described in Section 3.3.

4.3 Results

We measured performance of all methods on three out-of-domain evaluations. For the supervised methods, these evaluations assess how well the LLM text detectors perform in real world scenarios, where data distributions differ from the training distribution. Detectors that remain more invariant across these evaluations are considered more robust to changes and variations in data, thus showing better generalization to unseen domains. The results for each domain split are presented in Table 2, 3 and 4, while the summary of overall relevant findings is presented in Table 5. Note the data listed in the column name in all these tables is *withheld* from the training dataset, meaning the test dataset consists *entirely* of data from the specified column name.

Methods that are not zero-shot or linear models are inherently more stochastic; therefore, the UAR, RoBERTa, and SENTRA methods were ran over three random seeds. The main results in Tables 2, 3, 4, 5 show the means over these seeds. Additional details are shown in Appendix A.

Tables 2, 3 and 4 present performance of different baselines, measured with the AUROC metric, across different OOD test data for the RAID, M4GT and MAGE datasets respectively. The MEAN and WORST columns represent the average and the worst performance results of the baselines taken across the OOD test data, and the bold numbers indicate the best-performing models (on

	MEAN	WORST	Abstracts	Books	News	Poetry	Recipes	Reddit	Reviews	Wiki
RoBERTa [22]	90.9	84.4	93.1	87.0	91.4*	95.2*	84.4	93.9*	90.2	91.8
Text-Fluor. [40]	76.4	70.6	71.4	82.4	74.9	70.6	76.1	79.2	73.9	82.6
UAR-D [31]	81.7	71.4	71.4	85.2	84.5	73.2	89.5*	82.4	84.9	82.3
UAR-L [31]	87.3	76.3	89.6	91.1	89.8	76.3	85.3	88.8	88.1	89.3
PPL [10]	72.9	69.4	69.7	76.8	69.4	73.9	69.6	76.6	75.8	71.3
Binoculars [12]	82.0	79.4	83.2	84.3	80.2	83.5	79.4	83.2	82.1	80.2
F-DetectGPT [2]	78.6	75.6	80.0	80.1	77.9	77.1	75.6	78.8	80.0	79.4
Ghostbuster [35]	84.7	74.1	88.0	91.4	81.6	88.2	74.1	85.0	81.7	87.8
R-SENTRA	90.9	85.5	94.6	95.1*	88.4	92.5	85.5	91.7	87.8	91.8
SENTRA	**92.5**	**87.0**	95.1*	94.1	91.3	95.0	87.0	93.7	90.4*	93.2*

Table 2: AUROC Metric Performance for for the RAID OOD evaluation. The best mean and worst-case performance are in bold. The best result in each domain are marked by *.

	MEAN	WORST	arXiv	OUTFOX	PeerRead	Reddit	wikiHow	Wikipedia
RoBERTa [22]	88.2	82.8	97.8*	84.9	82.8	89.6	85.5	88.5
Text-Fluor. [40]	83.2	78.1	84.7	84.8	89.2	83.9	78.1	78.3
UAR-D [31]	75.3	63.9	73.3	83.9	65.7	86.1	63.9	78.9
UAR-L [31]	84.7	71.0	93.8	87.6	87.1	80.3	71.0	88.4
PPL [10]	87.0	81.7	83.6	85.7	94.2	89.7	81.7	87.1
Binoculars [12]	89.1	79.0	93.1	82.6	90.5	93.8	79.0	95.4
F-DetectGPT [2]	87.4	79.1	91.9	80.3	88.2	91.0	79.1	93.7
Ghostbuster [35]	87.8	73.3	94.3	87.3	81.9	95.4	73.3	94.5
R-SENTRA	92.8	83.9	94.6	88.4*	94.9	97.7*	83.9	97.4
SENTRA	**93.0**	**87.1**	92.3	88.0	95.0*	97.7*	87.1*	97.7*

Table 3: AUROC Metric Performance for the M4GT OOD evaluation. The best mean and worst-case performance are in bold. The best result in each domain are marked by *.

average and in the worst case) in these tables. Also, the asterisks (*) indicate the best-performing models for each test case.

As Tables 2, 3 and 4 show, SENTRA outperformed all the baselines on average and in the worst case across the three datasets RAID, M4GT and MAGE. Also, SENTRA and R-SENTRA models outperformed the baselines in most of the test cases (across the specific columns since most of the asterisks are associated with the SENTRA and R-SENTRA models in the columns of these tables). In a few specific domain splits where SENTRA/R-SENTRA lost to other baselines (usually RoBERTa), the performance loss was marginal (e.g., 91.3 vs. 91.4 for News, 95.0 vs. 95.2 for Poetry and 93.7 vs. 93.9 for Reddit for RAID evaluations - see Table 2).

Table 5 summarizes the AUROC OOD performance results taken directly from the MEAN columns of Tables 2, 3 and 4. It demonstrates SENTRA outperforms all other baselines for the three datasets RAID, M4GT and MAGE by 1.8%, 5.4% and 6.7% respectively, as compared to the second-best performing baseline.

All these results show SENTRA serves as a generalizable encoder for LLM detection models when one considers likely OOD distribution shifts. As Table 5 also shows, SENTRA's performance improves after pre-training: it is 92.5 vs. 90.9 on the RAID dataset, 93.0 vs. 92.8 on M4GT, and 94.2 vs. 93.8 on the MAGE dataset. The improved OOD performance indicates pre-training helps SENTRA learn a more generalizable representation to shifts in the data and demonstrates the effectiveness of our contrastive pre-training method for SENTRA.

Since LLMs became increasingly more available and their usage has surged, interest in detection tools, such as those presented in this paper, has grown (Wu et al., 2023). At the same time, countermeasures have emerged to attack these LLM text detectors, typically by altering LLM-generated text to elicit false negatives (Koike et al., 2024). Dugan et al. (2024) demonstrated many attacks can significantly degrade detector performance. In that study, the best open-source tool, Binoculars (Hans et al., 2024), exhibited much stronger performance on non-attacked data than on attacked data. For unsupervised methods, (Guo et al., 2023; Hans et al.,

	MEAN	WORST	CMV	ELI5	HSWAG	ROCT	SciGen	SQuAD	TL;DR	WP	XSum	Yelp
RoBERTa [22]	88.3	74.4	94.8	92.9	87.4*	88.8*	84.3	93.3	85.7	90.3	74.4	91.3
Text-Fluor. [40]	63.9	47.8	62.1	61.9	69.5	71.6	79.1	53.3	73.2	56.5	47.8	64.3
UAR-D [31]	63.4	40.5	80.2	74.4	63.5	61.5	56.5	59.6	60.1	67.8	40.5	70.3
UAR-L [31]	76.4	61.2	90.1	81.9	61.2	73.5	80.6	76.1	66.3	88.2	69.0	77.5
PPL [10]	57.2	45.7	57.9	61.4	73.8	61.2	49.4	48.3	62.9	59.4	45.7	51.9
Binoculars [12]	61.7	52.9	71.0	70.2	59.3	52.9	59.7	55.3	63.4	67.2	57.6	60.5
F-DetectGPT [2]	63.0	54.9	71.3	70.1	66.1	60.5	56.4	57.4	66.2	64.5	54.9	62.1
Ghostbuster [35]	79.2	65.0	90.5	86.0	66.2	65.0	83.6	78.8	74.0	94.1	72.4	80.9
R-SENTRA	93.8	84.6	98.5	95.2	84.6	87.3	97.9*	94.1*	93.4	98.6	93.8	94.4
SENTRA	**94.2**	**86.0**	98.6*	95.4*	86.0	88.2	97.6	93.9	94.1*	98.9*	94.4*	95.1*

Table 4: AUROC Metric Performance for the MAGE OOD evaluation. The best mean and worst-case performance are in bold. The best result in each domain are marked by *.

	RAID	M4GT	MAGE
RoBERTa [22]	90.9	88.2	88.3
Text-Fluor. [40]	76.4	83.2	63.9
UAR-D [31]	81.7	75.3	63.4
UAR-L [31]	87.3	84.7	76.4
PPL [10]	72.9	87.0	57.2
Binoculars [12]	82.0	89.1	61.7
F-DetectGPT [2]	78.6	87.4	63.0
Ghostbuster [35]	84.7	87.8	79.2
R-SENTRA	90.9	92.8	93.8
SENTRA	**92.5**(+1.8)*	**93.0**(+5.4)*	**94.2**(+6.7)*

Table 5: Evaluation Summary: Expected performance results (mean AUROC) across domains for our three evaluations. The best results are marked in bold. The percentage change of the best model over the best baseline is shown in parenthesis.

2024; Bao et al., 2024), it is not immediately clear how to adapt the approach to a known attack. In contrast, for supervised methods, the adaptation strategy is straightforward: train on attacked data. The results on the RAID dataset in Table 2 include 11 forms of attack. When the attack type is known and models are trained on the attacked data, Table 2 suggests SENTRA is the most effective method at adapting to those attacks.

5 Conclusions

In this paper, we proposed a novel general purpose supervised LLM text detector method SENTRA that is a transformer-based encoder leveraging SNTP sequences and utilizing contrastive pre-training on large amounts of unlabeled data. Since, supervised detectors tend to perform better on data that is similar to their training distributions (Dugan et al., 2024), it is essential to include a wide variety of domains when testing such general-purpose detectors. Therefore, we tested the performance of SENTRA on three public datasets RAID, M4GT and MAGE containing a broad range of different domains (24 in total) across various experimental settings and compared its performance with eight popular baselines.

We empirically demonstrated SENTRA significantly outperformed all baselines in most of the experimental settings: it achieved AUROC performance improvements of 1.8%, 5.4% and 6.7% for RAID (Dugan et al., 2024), M4GT (Wang et al., 2024a) and MAGE (Li et al., 2024) out-of-domain datasets respectively, as compared to the second-best performing baseline. On our three evaluation datasets, SENTRA outperformed all eight popular baselines in expected and worst-case out-of-domain performance, and SENTRA/R-SENTRA was also the best model in 17 out of 24 of the domain specific experiments. Even in the few cases when SENTRA/R-SENTRA (SENTRA without pre-training) lost to particular baselines (mostly RoBERTa), the performance loss was usually marginal (e.g., 91.3 vs. 91.4 for News, 95.0 vs. 95.2 for Poetry and 93.7 vs. 93.9 for Reddit domains for RAID evaluations).

This shows SENTRA is a strong method for training LLM text detectors that can generalize to unseen domains. We also demonstrated our contrastive pre-training strategy increased the performance of SENTRA on these out-of-domain evaluations. Domain generalization is one of the most critical issues for LLM text detectors. These results demonstrate that SENTRA is a general purpose encoder that can serve as a foundation for LLM text detector models.

6 LLM Acknowledgment

We used ChatGPT for generating first iterations of some software snippets. We also consulted ChatGPT on the phrasing of some points in the paper and for catching some grammatical errors.

7 Limitations

In this work, we studied the effects of domain shifts on detection models. While these have significant impacts on detector performance, other factors can also influence results. Notably, prompt variation can have a large effect on detectors (Kumarage et al., 2023b). Many LLM detection benchmark datasets use prompt templates (Dugan et al., 2024) to generate their samples. However, these templates exhibit significantly less prompt variety than what a real-world detector is likely to encounter. Benchmark datasets with a broader range of prompting strategies are needed to further assess the robustness of detection methods.

In this work, we followed Binoculars (Hans et al., 2024) in choosing Falcon (Almazrouei et al., 2023) models as the SNTP generators. This decision was primarily based on Binoculars' strong performance, allowing for a direct and fair comparison. However, it is important to note SENTRA is a general methodology, and other SNTP generators may perform better or more efficiently than Falcon models.

We pre-trained our model on a relatively small sample of Common Crawl data. The volume of data and the amount of compute used for pre-training were small relative to what is typically used for foundation models (Liu et al., 2019; Radford et al., 2021). It is very likely SENTRA could be significantly improved with additional pre-training on larger datasets.

8 Ethical Considerations

In this study, we did not observe any detector achieving perfect performance on any slice of data. Therefore, any detector will inherently make tradeoffs between false positives and false negatives when deployed in real-world scenarios, such as plagiarism detection. Users of LLM detection technology should be aware that these detectors are not perfect.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The Falcon Series of Open Language Models. *arXiv preprint*. ArXiv:2311.16867 [cs].

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations*.

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. ConDA: Contrastive domain adaptation for AI-generated text detection. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 598–610, Nusa Dua, Bali. Association for Computational Linguistics.

Amrita Bhattacharjee, Raha Moraffah, Joshua Garland, and Huan Liu. 2024. Eagle: A domain generalization framework for ai-generated text detection. *arXiv preprint arXiv:2403.15690*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12463–12492, Bangkok, Thailand. Association for Computational Linguistics.

Kathleen C. Fraser, Hillary Dawkins, and Svetlana Kiritchenko. 2024. Detecting ai-generated text: Factors influencing detectability with current methods. *Preprint*, arXiv:2406.15583.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. 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. *Preprint*, arXiv:2301.07597.

Xun Guo, Yongxin He, Shan Zhang, Ting Zhang, Wanquan Feng, Haibin Huang, and Chongyang Ma. 2024. Detective: Detecting AI-generated text via multi-level contrastive learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Wei Hao, Ran Li, Weiliang Zhao, Junfeng Yang, and Chengzhi Mao. 2024. Learning to Rewrite: Generalized LLM-Generated Text Detection. *arXiv preprint*. ArXiv:2408.04237 [cs].

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *Advances in Neural Information Processing Systems*, 36:15077–15095.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21258–21266.

Tharindu Kumarage, Joshua Garland, Amrita Bhattacharjee, Kirill Trapeznikov, Scott Ruston, and Huan Liu. 2023a. Stylometric Detection of AI-Generated Text in Twitter Timelines. *arXiv preprint*. ArXiv:2303.03697 [cs].

Tharindu Kumarage, Paras Sheth, Raha Moraffah, Joshua Garland, and Huan Liu. 2023b. How reliable are AI-generated-text detectors? an assessment framework using evasive soft prompts. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1337–1349, Singapore. Association for Computational Linguistics.

Zhixin Lai, Xuesheng Zhang, and Suiyao Chen. 2024. Adaptive Ensembles of Fine-Tuned Transformers for LLM-Generated Text Detection. *arXiv preprint*. ArXiv:2403.13335 [cs].

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53, Bangkok, Thailand. Association for Computational Linguistics.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong, and Philip Yu. 2025. A Survey of Text Watermarking in the Era of Large Language Models. *ACM Computing Surveys*, 57(2):1–36.

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 preprint*. ArXiv:1907.11692 [cs].

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

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*, ICML'23. JMLR.org.

Hoang-Quoc Nguyen-Son, Minh-Son Dao, and Koji Zettsu. 2024. SimLLM: Detecting Sentences Generated by Large Language Models Using Similarity between the Generation and its Re-generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22340–22352, Miami, Florida, USA. Association for Computational Linguistics.

Originality.AI. 2025. Originality.ai - ai plagiarism and fact checker.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Reality Defender. 2025. Reality defender.

Dmitri Roussinov, Serge Sharoff, and Nadezhda Puchnina. 2025. Controlling out-of-domain gaps in LLMs for genre classification and generated text detection. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3329–3344, Abu Dhabi, UAE. Association for Computational Linguistics.

Sapling AI. 2025. Ai detector.

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. *Commun. ACM*, 67(4):50–59.

Edward Tian and Alexander Cui. 2023. Gptzero: Towards detection of ai-generated text using zero-shot and supervised methods".

Yuchuan Tian, Hanting Chen, Xutao Wang, Zheyuan Bai, QINGHUA ZHANG, Ruifeng Li, Chao Xu, and Yunhe Wang. 2024. Multiscale positive-unlabeled detection of AI-generated texts. In *The Twelfth International Conference on Learning Representations*.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2024. Ghostbuster: Detecting text ghostwritten by large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1702–1717, Mexico City, Mexico. Association for Computational Linguistics.

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. 2024a. M4GT-bench: Evaluation benchmark for black-box machine-generated text detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3964–3992, Bangkok, Thailand. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. *NeurIPS Datasets and Benchmarks Track*.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F. Wong, and Lidia S. Chao. 2023. A Survey on LLM-generated Text Detection: Necessity, Methods, and Future Directions. *arXiv preprint*. ArXiv:2310.14724 [cs].

Xiao Yu, Kejiang Chen, Qi Yang, Weiming Zhang, and Nenghai Yu. 2024. Text fluoroscopy: Detecting LLM-generated text through intrinsic features. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15838–15846, Miami, Florida, USA. Association for Computational Linguistics.

Ying Zhou, Ben He, and Le Sun. 2024. Humanizing machine-generated content: Evading ai-text detection through adversarial attack. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation.*

A Additional Results and Experimental Notes

The datasets used in this work were used for research purposes. This aligns with their intended use and licenses.

Here we show the mean and standard deviation across three runs, (random seeds 42,43,44) for the methods that are not zero shot or logistic regression based. Note there were three M4GT and four RAID samples where Ghostbuster could not make an inference due to the low number of tokens in the document. For this documents, we infilled a low prediction score indicating human prediction. For the RAID dataset, we used the Binoculars for each document released by (Dugan et al., 2024).

B Computational Complexity

LLM generators are computationally expensive. Unfortunately, methods that rely on SNTP inputs depend on LLM inference, making it the most costly component of all detection methods studied in this work. However, SENTRA is a relatively small model with only eight Transformer layers, meaning that computational costs at inference are dominated by the production of SNTP inputs. During training, we cache the SNTP sequences so that the LLMs are run only once per sample. SENTRA uses the same LLMs as Binoculars (Hans et al., 2024), and because the cost of the SENTRA encoder is minimal compared to LLM inference, the overall computational complexity of SENTRA is roughly equivalent to that of the Binoculars method. Refer to Table 9 for detailed number of parameters.

Pre-training took approximately 36 hours on a GH200 GPU. We also fine-tuned RoBERTa and SENTRA models on GH200 instances. Fine-tuning for each data split too between .5 and 12 hours.

C Baseline Assumptions and Setups

This section details the assumptions and setups for all baseline methods if we have made modifications.

For UAR, the original paper compares the distance between the input query and the closest

	abstracts	books	news	poetry	recipes	reddit	reviews	wiki
RoBERTa	93.1±1.2	87.0±2.1	91.4±3.4*	95.2±1.3*	84.4±16.9	93.9±1.2*	90.2±2.3	91.8±2.8
Text-Fluor.	71.4±0.0	82.4±0.0	74.9±0.0	70.6±0.0	76.1±0.0	79.2±0.0	73.9±0.0	82.6±0.0
UAR-D	71.4±4.4	85.2±0.8	84.5±1.2	73.2±0.5	89.5±0.8*	82.4±0.3	84.9±1.1	82.3±0.2
UAR-L	89.6±2.0	91.1±0.2	89.8±0.4	76.3±2.6	85.3±1.2	88.8±0.7	88.1±0.4	89.3±0.5
PPL	69.7±0.0	76.8±0.0	69.4±0.0	73.9±0.0	69.6±0.0	76.6±0.0	75.8±0.0	71.3±0.0
Binoculars	83.2±0.0	84.3±0.0	80.2±0.0	83.5±0.0	79.4±0.0	83.2±0.0	82.1±0.0	80.2±0.0
Fast-DetectGPT	80.0±0.0	80.1±0.0	77.9±0.0	77.1±0.0	75.6±0.0	78.8±0.0	80.0±0.0	79.4±0.0
Ghostbuster	88.0±0.0	91.4±0.0	81.6±0.0	88.2±0.0	74.1±0.0	85.0±0.0	81.7±0.0	87.8±0.0
R-SENTRA	94.6±0.3	95.1±0.3*	88.4±0.5	92.5±2.2	85.5±0.9	91.7±0.1	87.8±0.5	91.8±0.3
SENTRA	95.1±0.1*	94.1±1.6	91.3±0.5	95.0±0.8	87.0±1.5	93.7±0.5	90.4±0.9*	93.2±0.7*

Table 6: Mean and standard deviation of the AUROC across random seeds on the RAID dataset.

	arxiv	outfox	peerread	reddit	wikihow	wikipedia
RoBERTa	97.8±0.3*	84.9±2.2	82.8±18.6	89.6±3.9	85.5±2.3	88.5±3.9
Text-Fluor.	84.7±0.0	84.8±0.0	89.2±0.0	83.9±0.0	78.1±0.0	78.3±0.0
UAR-D	73.3±6.7	83.9±0.2	65.7±1.0	86.1±1.0	63.9±0.6	78.9±2.2
UAR-L	93.8±1.2	87.6±0.6	87.1±0.4	80.3±1.1	71.0±2.4	88.4±0.7
PPL	83.6±0.0	85.7±0.0	94.2±0.0	89.7±0.0	81.7±0.0	87.1±0.0
Binoculars	93.1±0.0	82.6±0.0	90.5±0.0	93.8±0.0	79.0±0.0	95.4±0.0
Fast-DetectGPT	91.9±0.0	80.3±0.0	88.2±0.0	91.0±0.0	79.1±0.0	93.7±0.0
Ghostbuster	94.3±0.0	87.3±0.0	81.9±0.0	95.4±0.0	73.3±0.0	94.5±0.0
R-SENTRA	94.6±0.5	88.4±0.4*	94.9±0.2	97.7±0.3*	83.9±1.3	97.4±0.3
SENTRA	92.3±1.0	88.0±0.1	95.0±0.3*	97.7±0.2	87.1±1.7*	97.7±0.3*

Table 7: Mean and standard deviation of the AUROC across random seeds on the M4GT dataset.

	cmv	eli5	hswag	roct	sci_gen	squad	tldr	wp	xsum	yelp
RoBERTa	94.8±1.0	92.9±0.7	87.4±4.2*	88.8±1.0*	84.3±6.5	93.3±1.0	85.7±5.1	90.3±1.5	74.4±3.4	91.3±1.6
Text-Fluoroscopy	62.1±0.0	61.9±0.0	69.5±0.0	71.6±0.0	79.1±0.0	53.3±0.0	73.2±0.0	56.5±0.0	47.8±0.0	64.3±0.0
UAR-D	80.2±1.8	74.4±1.7	63.5±2.3	61.5±2.5	56.5±4.7	59.6±3.4	60.1±1.7	67.8±3.3	40.5±0.9	70.3±0.4
UAR-L	90.1±0.7	81.9±0.7	61.2±2.4	73.5±1.0	80.6±1.7	76.1±0.8	66.3±2.8	88.2±0.9	69.0±1.9	77.5±1.3
PPL	57.9±0.0	61.4±0.0	73.8±0.0	61.2±0.0	49.4±0.0	48.3±0.0	62.9±0.0	59.4±0.0	45.7±0.0	51.9±0.0
Binoculars	71.0±0.0	70.2±0.0	59.3±0.0	52.9±0.0	59.7±0.0	55.3±0.0	63.4±0.0	67.2±0.0	57.6±0.0	60.5±0.0
Fast-DetectGPT	71.3±0.0	70.1±0.0	66.1±0.0	60.5±0.0	56.4±0.0	57.4±0.0	66.2±0.0	64.5±0.0	54.9±0.0	62.1±0.0
Ghostbuster	90.5±0.0	86.0±0.0	66.2±0.0	65.0±0.0	83.6±0.0	78.8±0.0	74.0±0.0	94.1±0.0	72.4±0.0	80.9±0.0
R-SENTRA	98.5±0.2	95.2±0.7	84.6±0.6	87.3±0.6	97.9±0.1*	94.1±0.3*	93.4±0.3	98.6±0.3	93.8±1.7	94.4±0.2
SENTRA	98.6±0.2*	95.4±0.4*	86.0±0.3	88.2±0.5	97.6±0.8	93.9±0.6	94.1±0.4*	98.9±0.1*	94.4±1.0*	95.1±0.2*

Table 8: Mean and standard deviation of the AUROC across random seeds on the MAGE dataset.

Method	Parameter Count
RoBERTa-base	124M
Text Fluoroscopy	7B (LLM) + 5.1M (FCN) $\approx$ 7B
UAR	82M
Perplexity	7B (LLM)
Binoculars	14B (2 LLMs)
Fast-DetectGPT	2.7B + 6B (2 LLMs) = 8.7B
Ghostbuster	7B (LLM) + N (LR, N « 7B) $\approx$ 7B
SENTRA	57M (training), 14B (inference)
R-SENTRA	57M (training), 14B (inference)

Table 9: Parameter count of all methods with the actual LLM(s) used in evaluation. LR stands for logistic regression, FCN stands for fully connected network. For Ghostbuster, we observed N to be between 20 to 40.

machine support query against the distance between the closest machine support query and the closest human support query. Mathematically speaking, given Q the input query, H the closest human support query, and $\mathbf{M}$ is the seeded machine support queries, the distance $d_{\mathtt{Q}} = \min_{m \in \mathbf{M}}[d(Q, m), d(H, m)]$ is used as the prediction. Though this allows d_Q to be directly usable for metric calculation, this is less trivial than a simple nearest neighbor classification where we calculate the percentage of machine support queries among k as the prediction. in our baseline, we employed the simple nearest neighbor approach with $k = 10$ and cosine similarity distance measure. For each domain, we randomly sampled 1,000 human and machine texts respectively to form the kNN seed corpus. We did not group texts into episodes and kept episode size of 1 due to the generally longer text lengths compared to twitter posts.

For Text Fluoroscopy, we switched the model from gte-Qwen1.5-7B-instruct to Falcon-7B-Instruct to better align with other baselines by eliminating the effect of model selection. With this change, we modified the input dimension to the feed forward network from 4096 to 4454 due to falcon models hidden state sizes. Despite the possibilities of under-training, we followed their implementation and sampled 160 data points for training, and 20 for validation (during training). The test set metric at the earliest highest validation accuracy was reported. We also optimized the feature selection script for more efficient batch processing.

For Ghostbuster, we included a minimum accuracy score improvement threshold of $1\mathrm{e}{-4}$ to avoid over-fitting and allow early stopping for MAGE dataset where we observed significantly more feature selection iterations compared to the other two datasets. In the case of least square convergence failure (`max_iter=1000`) in Logistic Regression fitting, the current feature list is taken as the best features for evaluation.

D Hyper-parameter Selection

For RoBERTa, we chose one domain from the MAGE dataset to tune the learning rate. With this learning rate, the RoBERTa diverged before the first epoch on one MAGE split and one RAID split. We then turned down the learning rate for these two splits and reran RoBERTa, but the models still diverged. It is possible with additional tuning, RoBERTa could better fit these datasets, but we did not want to pay special attention to the fine-tuning any one method.

For SENTRA, we did a small search over the number of layers, {2,4,8}, for the CMV-MAGE data split by looking at the in-domain development loss. We found four layers to work best. We later found SENTRA had trouble fitting the in-distribution validation data of a data. We found that varying the LR and batch size on this dataset had no significant effect, so we kept the defaults of a LR of $1\mathrm{e}-4$ and a batch size of 128 which were the defaults from RoBERTa. We then manually tuned the pre-training model while looking at this in-distribution loss. We ultimately found that eight layers and and two pre-training phases produced the best performance on this in distribution validation dataset.