Team DACTYL at PAN 2026: Bayesian Data Mixing and Empirical X-risk Minimization for AI-text Detection

Notebook for the PAN Lab at CLEF 2026

Shantanu Thorat[1,*]

[1]College Station, Texas, US

Abstract

Existing research shows that AI-generated text detection classifiers achieve strong in-distribution (ID) performance but do not maintain the same performance on out-of-distribution (OOD) texts, suggesting overfitting to dataset-specific features. However, combining different training datasets doesn't always improve performance and, in some cases, can even encourage shortcut learning. To address this issue, we fine-tune BERT-tiny models with Bayesian classification heads to select texts across three different datasets to use as a consolidated training set. We trained three different classifiers: fine-tuned DeBERTa-V3-large and ModernBERT-large classifiers via empirical X-risk minimization, and an MCGrad model that calibrates the predictions from the ModernBERT-large classifier. The DeBERTa-V3-large-large classifier achieves a mean score of 0.882 on the PAN 2026 test set across five metrics: AUROC, F_1, C@1, Brier score, and $F_{0.5u}$. ModernBERT-large achieves a score of 0.96 while MCGrad achieves the best score of the three with a mean score of 0.974, ranking second on the leaderboard. Our results highlight that careful dataset curation can lead to strong OOD performance.

Keywords

Bayesian neural networks, empirical X-risk minimization, multicalibration, AI-generated text detection

1. Introduction

The Voight-Kampff 2026 Task at PAN at CLEF 2026 focused on binary AI-generated (AIG) text detection: identifying if a text came from a large language model (LLM) or a human [1, 2, 3, 4]. Despite many classifiers achieving high performance when faced with texts that mirror their training distributions, AIG text detectors struggle to generalize to out-of-distribution (OOD) texts [5]. We suspect that AIG text detection datasets may contain spurious correlations — e.g., texts that contain unintentional artifacts that coincidentally map to AI-generated content or human-written content. Empirically, for example, we observe that for the RAID benchmark test set, a 4 million parameter (BERT-tiny) model achieved a competitive AUROC (0.967) score against larger classifiers[6]. On another AIG text detection dataset, the DACTYL test set, BERT-tiny achieves an AUROC score of 0.99 [5]. These results suggest that AIG text detection datasets may contain superficial features that allow for smaller models to obtain misleadingly high scores. A naive way to combat dataset-specific issues is to combine datasets. However, this approach doesn't always help machine-learning models and in some cases *reinforces* a model's dependency on these spurious correlations [7].

Previous work to address these issues include hard mining — retraining classifiers on texts they struggled with — has yielded some success in improving robustness [8]. However, this technique doesn't necessarily distinguish between texts that are genuinely difficult or simply noisy texts (e.g., a corrupted/hallucinated generation). This lack of distinguishability leads to another issue — classifier's scores (scaled from 0 to 1, where 0 means human-written and 1 means AIG) aren't necessarily calibrated, meaning the score itself may not be a reliable signal for evaluating a classifier's confidence on an example. Additionally, a single (even calibrated) score doesn't indicate a model's uncertainty on an

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

*Work completed independently of employment at Texas A&M University.

st980[at]cantab.ac.uk (S. Thorat)

https://shantanuthorat.io (S. Thorat)

0009-0004-8470-0713 (S. Thorat)

input. Finally, a majority of BERT-based classifiers use empirical risk minimization (ERM): the training loop optimizes cross-entropy loss or its variants, such as focal loss. Empirical risk minimization, while successful in many classification tasks and domains, tends to encourage shortcut-learning, leading to classifiers struggling to generalize to OOD texts [9].

To address these issues, we contribute the following parts of our system:

- We construct DACTYLv2, an expansion of the original DACTYL dataset.
- To better select texts for training, we train weak BERT-tiny models with a Bayesian neural network (BNN) classifier head (we refer to this architecture as Bayesian BERT-tiny), which can output different scores for a given text, allowing us to compute μ (mean score) and σ (standard deviation). Using these features, we can filter out texts from different datasets to use for training. These filtering steps allow us to blend different datasets together for one final training set.
- Instead of traditional ERM, we use empirical X-risk minimization (EXM), specifically, two-way partial AUROC optimization via the LibAUC library [10]. EXM has shown promising results in generalizability, especially with larger models [5].
- We produce three different classifiers: a ModernBERT-large and a DeBERTa-V3-large classifier [11, 12]. For our third classifier, we use the weak Bayesian BERT-tiny classifiers' uncertainties (σ) as features to calibrate ModernBERT-large via MCGrad [13].

2. Datasets Used

2.1. Training

2.1.1. DACTYLv2

Building upon the work in [5], we construct the second version of DACTYL, which we denote as DACTYLv2 — we refer to DACTYL-complete as DACTYLv1 and DACTYLv2 combined. We create all AIG texts via one-shot prompting; we gave the LLM a human-written example to base its generation from. We included the following domains in DACTYLv2:

- blogs [14]
- web articles [15]
- Medium articles [16]
- movie scripts [17]
- NBER economic paper abstracts [18]
- Yelp reviews [19]

We use the following models to construct the dataset.

- Claude Haiku 4.5 [20]
- Claude Sonnet 4.6 [21]
- GPT 5.4 [22]
- GLM 5.1 [23]
- MiniMax M2.5 [24]
- Mercury M2 [25]

We used mirror prompting to create LLM prompts as used in [8] for the blogs, web articles, and Medium articles. We provided the GPT 5.4 Nano model with the human article and asked it to generate a possible prompt that could have inspired the article [26]. For movie scripts, given two randomly selected movie scripts, we take a random excerpt of 4,096 characters from each script. We prompt the LLM to rewrite the first random excerpt by mimicking the style of the second random excerpt. For NBER abstracts, we have the LLM copy the style of a random abstract and generate a new abstract given only the title of another randomly selected paper title. For Yelp reviews, we take a randomly

selected business, its industry, and one random review and ask the LLM to generate another review using the selected review as a style reference. For Yelp reviews only, we also included texts generated by Kimi-K2.5 [27]. After manual inspection of Kimi-K2.5's generations, we observed that the specific model struggled with style mimicking compared to other LLMs, so we did not use the model for other domains. We report aggregated counts by generator (LLM or human) and domain in Table 1 after deduplication.

For the open-source LLMs, we used the models provided by the TogetherAI and FireworksAI platforms [28, 29].

Table 1
DACTYLv2 Text Counts by Domain/Model across all splits

Model	NBER Abstracts	Yelp	Web	Blogs	Medium Articles	Scripts (Movie)	Total Texts
GLM 5.1	1,400	1,400	160	1,150	5,996	275	10,381
GPT 5.4	1,400	1,400	160	1,150	6,000	275	10,385
Sonnet 4.6	1,399	1,400	158	1,143	5,994	274	10,368
Haiku 4.5	1,400	1,400	160	1,150	6,000	275	10,385
Mercury 2	1,400	1,397	159	1,076	5,985	274	10,291
Kimi-K2.5	0	1,387	0	0	0	0	1,387
Qwen3.5-397B-A17B	1,400	1,337	160	1,150	5,986	275	10,308
MiniMax-M2.5	1,388	1,396	160	1,133	5,954	229	10,260
Human	30,600	90,078	21,123	97,727	156,362	2,200	398,090
Total	**40,387**	**101,198**	**22,240**	**105,679**	**198,277**	**4,077**	**471,858**

2.1.2. External Datasets

Since DACTYL-complete only includes one-shot generations, we included texts from other existing datasets. Augmenting our training set also ensures that our final classifiers don't overfit to DACTYL-specific artifacts unintentionally. We selected LLMTrace (English only) [30], Personagen and Stack Exchange [31, 32], MAGA-Bench (English only) [33], and DetectRL [34] as additional *candidate* training datasets to complement DACTYL-complete. LLMTrace's English split includes multiple prompt types, such as updating a human-written text as well as mixed texts. Personagen includes a specific prompt style — LLMs act as a specific human persona before generating their responses. MAGA-Bench includes generations from different LLM "alignments" (e.g., role-playing, black-box prompt optimization, etc.). DetectRL includes mixed LLM-generated texts where a single AIG text may have contributions from different LLMs. Since Personagen only includes AIG texts, we complement it with a subset of Stack Exchange posts. For each community (that wasn't a meta community), we randomly sampled up to 10,000 posts before November 1, 2022.

2.2. Calibration

We perform an additional layer of calibration for one of our models (ModernBERT-large), given that the scoring metrics for PAN CLEF 2026 include Brier score loss. We use the following datasets listed in Table 2. For each dataset, we take a random sample of 2,000 texts, followed by deduplication, to avoid a larger dataset from dominating our calibration set. Due to an imbalance of AIG texts over human texts, we include samples from human-only datasets: CC-News and Stack Exchange. We did not perform calibration on DeBERTa-V3-large since it already obtained strong performance on the calibration set.

2.3. Testing

We evaluate our models on numerous test sets to evaluate generalizability since an updated validation set was not released this year, in contrast to previous years [37]. We use the following datasets listed in

Table 2
Calibration counts by dataset and generator.

Dataset	Total Texts	AI Texts	Human Texts
DACTYLv1 (validation)	1,997	738	1,259
Personagen	2,000	2,000	0
AIGC Text Bank [35]	2,000	1,913	87
AuthorAware	2,000	2,000	0
OpenTuringBench [36]	1,995	1,995	0
DACTYLv2 (validation)	2,000	469	1,531
PAN-CLEF 2025 (training)[37]	2,000	1,252	748
CC News [38]	1,961	0	1,961
NY Times AI [39]	2,000	1,713	287
Stack Exchange	2,000	0	2,000
SAGE [40]	1,985	1,523	462
LLMTrace (English, validation)	2,000	1,210	790
Detect AI [41]	2,000	1,346	654
Total	**25,938**	**16,159**	**9,779**

Table 3. We included ID test sets (DACTYL-complete, MAGA-Bench, LLMTrace) as well as OOD test sets.

Table 3
External test sets used.

Dataset	Domain Count	LLMs Used
BEEMO[42]	5	10
CoCONUTS[43]	6	7
DACTYL-complete (DACTYLv2 and DACTYLv1 [5])	12	25
DetectRL	4	4
Dolly-Cosmopedia [44, 45]	21	1
LLMTrace (English)	9	19
MAGA-Bench (English)	10	12
OriginalityAI [46]	Not specified	4
PAN 2025 Val [37]	3	22
Personagen + StackExchange	4	6
RealDet (English)[47]	15	22
UChicago [48]	6	4

3. Data Filtering

3.1. Bayesian BERT-Tiny Classifiers

To get an estimate on dataset generalizability, we train small classifiers on a sample of these datasets. Each smaller classifier is a BERT-tiny model with a Bayesian variant of a linear layer as the classification head [49, 50, 51]. The Bayesian head's input dimensions are equivalent to the hidden size of the BERT-tiny model, and the output dimension size is 1. Bayesian neural networks (BNNs) allow for the estimation of σ (standard deviation) of a prediction on a specific input. This uncertainty modeling comes from how a BNN represents its internal weights — rather than having individual scalars, the BNN replaces each scalar with a distribution [52]. During inference time, we sample a set of parameters to use for a prediction. Upon doing n samples, we can compute the (μ, σ), or the mean score and standard deviation.

Training a Bayesian BERT-tiny for classification is nearly identical to a standard BERT model's

training, except it optimizes the ELBO (evidence lower bound) loss. We can define ELBO loss as

$$\text{ELBO} = L + \frac{\beta}{b}\text{KL}\left[q(w|\theta)||P(w)\right] \tag{1}$$

where L is our loss function of choice, $q(w|\theta)$ is our variational approximation to the posterior, and $P(w)$ is our prior. θ are the parameters that characterize the distribution of weights w for our model. The KL divergence term prevents the model's weights from moving too far away from our prior. The fraction β/b is a scaling factor to avoid the KL term from dominating the ELBO loss. β is a positive value (for our experiments, we set it to 1) and b is the mini-batch size during training.

For each of the five training sets, we train a Bayesian BERT-tiny classifier on a random sample of the training set. We use the two-way partial AUROC loss function as L (the DRO-KL variant also used in [5]), a batch size of 64, a learning rate of 1×10^{-4}, and a weight decay of 0.01. We train each classifier for one epoch. Since the two-way partial AUROC loss needs at least one positive sample per mini-batch, we set the proportion of positive samples equal to 0.5 for each mini-batch. Due to each of the five datasets possessing different distributions, we manually tuned the hyperparameters so that the classifier's uncertainty increased on OOD texts. We observed that the random sample size (defined as fraction f) and the initial posterior (ρ, which can be transformed into the standard deviation of weights via $\sigma = \log\left(1 + \exp\rho\right)$) influenced the model's uncertainty significantly. For each Bayesian BERT-tiny model, we assign some of the classification head hyperparameters to fixed values, according to the BLiTZ documentation [51]:

- prior σ_1 = 0.1
- prior σ_2 = 1.5
- prior π = 0.5
- initial posterior μ = 0

Table 4
Bayesian BERT-tiny training configurations.

Dataset	Initial Posterior ρ	f	Training Sample Count
DACTYL-complete	-3	1/5	167,099
DetectRL	-3	1	205,914
MAGA-Bench	-3	1/5	71,903
Personagen+SE	-3	1/80	25,565
LLMTrace	-2	1	173,511

We report initial AUROC scores by classifier and test set to evaluate discrimination ability between AI and human texts in Table 5.

Table 5
AUROC scores by Bayesian BERT-tiny classifier.

Train Dataset	DACTYL-complete	DetectRL	LLMTrace	MAGA-Bench	Personagen+SE
DACTYL-complete	0.9419	0.7725	0.8692	0.9429	0.9819
DetectRL	0.7014	0.9839	0.7324	0.6438	0.9702
LLMTrace	0.8361	0.8553	0.9633	0.9541	0.9757
MAGA-Bench	0.8205	0.7881	0.8761	0.9906	0.9978
Personagen+SE	0.6943	0.7597	0.8302	0.8745	0.9999

3.2. Filtering Process

We observe in Table 5 that DACTYL-complete, LLMTrace, and MAGA-Bench are among the best in cross-dataset generalizability. We focus on those datasets and use our classifiers to either "keep" or

"reject" (or "abstain" from a decision) texts to use for our final training set. There are four "quadrants" of classification outcomes for our Bayesian BERT-tiny models:

- correct but uncertain
- correct and certain
- incorrect but uncertain
- incorrect and certain

The first three outcomes are acceptable for a model to train on — a correct classification indicates that a classifier has the *capacity* to learn that sample, regardless of the certainty. An incorrect but uncertain classification for a sample suggests a difficult case — the classifier is aware that its prediction is possibly wrong. We discard samples in the fourth and final case, incorrect and certain. The intuition comes from dataset cartography: Swayamdipta et al. [53] demonstrated that some samples exist in the "hard-to-learn" space, which are samples that the model assigns low probability scores to the true class consistently — these "hard-to-learn" samples map roughly to the "incorrect and certain" quadrant. While this space can include mislabeled data points, it also includes samples that may be genuine outliers. Given that the two-way partial AUROC loss function focuses on the harder positive (AI) and negative (human) samples, this makes the loss function more sensitive to texts that have high loss and increases the probability of overfitting to outliers [54, 10, 55].

We define the voting process for an individual classifier as follows. For a given text t, we compute the standard deviation (σ_t) and mean score (μ_t, number of runs = 5 to minimize inference time) for the Bayesian BERT-tiny classifier C_D, where D is the training dataset. We refer to the distribution of standard deviations on the *test* set for D for classifier C_D as S_D. For example, with DACTYL-complete, we compute $S_{\text{DACTYL-complete}}$ by computing the individual σ_t values using the Bayesian BERT-tiny classifier trained on DACTYL-complete's subset — since we want a stable distribution, we do 30 runs for each text in the test set. We can compute the percentile of σ_t relative to S_D as P_t. A higher P_t indicates a larger value of σ_t, which implies the model is more uncertain — conversely, a lower P_t signals high confidence. We convert μ_t to label $\hat{y}$ (1 or AI-generated if $\mu_t > 0.5$ else 0) to compare to the true label y. Using this information, we can then see how C_D votes on t, as described in Table 6.

Table 6
Criteria for voting on text t using classifier C_D.

Criteria	Decision
(1) $P_t < 5$ or $P_t > 95$	Abstain
(2) $5 \leq P_t \leq 95$ and $P_t \leq 50$ and $\hat{y} \neq y$	Reject t
(3) Criteria (1) and (2) are not met	Keep t

Criterion 1 is a case where the classifier's uncertainty signal appears to be struggling — extremely high confidence (σ_t is less than the 5th percentile) or extreme uncertainty (σ_t is greater than the 95th percentile). A classifier abstaining from a text with significantly high uncertainty is reasonable; it hasn't seen this text before to make a reliable vote. However, exceptional overconfidence implies that the classifier's output may be miscalibrated.

Criterion 2 is the "incorrect and certain" sample — this is a sample where the classifier is relatively confident (P_t is less than the 50th percentile) but makes an incorrect prediction. Criterion 3 is the "keep" vote and occurs when the first two criteria aren't met.

We use the ensemble of classifiers' votes to decide whether or not to keep t. t is ultimately kept if at least three of the five classifiers vote to keep and no more than one classifier rejects t. For example, if three classifiers vote to keep t, but the remaining two classifiers reject it, then we discard t. However, if three classifiers vote to keep t, and one of the remaining classifiers abstains while the other rejects, we choose to keep t for the final training set. This rejection mechanism helps to minimize the chances of an unusual outlier text making it to the training set.

We report final dataset sizes in Table 7. The "Rejected" column refers to instances where a text received three "keep" votes and two "reject" votes.

Table 7
Filtering results per dataset. Δ denotes the change (percentage points) in the proportion of AIG texts after filtering. The rejected column denotes the number of texts that received three keep votes and two reject votes.

Train Dataset	Initial N	Kept N	% Kept	Rejected	AIG text% (Δ)
DACTYL-complete	826,891	727,235	87.9	1,802	$17.0\% \rightarrow 15.4\%(-1.6)$
LLMTrace	173,511	146,908	84.7	734	$60.8\% \rightarrow 57.5\%(-3.3)$
MAGA-Bench	359,518	297,994	82.9	1,387	$83.4\% \rightarrow 81.7\%(-1.7)$

4. Model Training

4.1. Pre-trained Model Fine-tuning

We train a (non-Bayesian) DeBERTa-V3-large and ModernBERT-large model on our filtered superset, which consists of all kept texts from the three datasets. We optimize the two-way partial AUROC loss function directly (we don't use ELBO as our objective, as the classification head is not a BNN). We use a learning rate of 1×10^{-5}, a batch size of 16, a sampling rate of 0.5, and one epoch for both models.

4.2. Multicalibration with MCGrad

For ModernBERT-large, we also perform post-hoc multicalibration with MCGrad using our calibration set [13]. Our MCGrad model takes in four inputs — the transformed score s from the ModernBERT-large model and the individual uncertainties (σ, 5 runs) for three Bayesian BERT-tiny classifiers (DACTYL, LLMTrace, MAGA-Bench).

We compute s via temperature scaling by transforming the raw probability score s_r into a logit from ModernBERT-large given temperature T in Equation 2.

$$\text{logit}(s_r) = \log \frac{\text{clip}(s_r,\, 10^{-6},\, 1 - 10^{-6})}{1 - \text{clip}(s_r,\, 10^{-6},\, 1 - 10^{-6})} \tag{2}$$

$$s = \frac{1}{1 + \exp\left(\frac{-\text{logit}(s_r)}{T}\right)} \tag{3}$$

We determine T (which we found $T = 2.14555$) by minimizing the log loss between the logits values and the target y values using SciPy's `minimize_scalar` function on the calibration set [56].

We note that MCGrad's algorithm allows for correction, meaning that individual predictions might change significantly.

5. Evaluation & Analysis

For testing on external datasets, we use the same approach described by the workshop organizers [1]. We use the following five metrics and the mean of all five metrics (which we denote as the overall score):

- AUROC score
- micro-F_1 score: the harmonic mean between precision and recall
- Brier Score: 1 minus the mean squared error between the predicted probability and true label
- C@1: modified accuracy that ignores cases where predicted probability = 0.5
- $F_{0.5u}$: modified $F_{0.5}$ where abstained cases (predicted probability = 0.5) are considered as false negatives

Table 8
ModernBERT-large test set results.

dataset	AUROC	F_1	C@1	Brier Score	$F_{0.5u}$	Overall
BEEMO	0.8560	0.9329	0.8806	0.9032	0.9322	0.9010
CoCONUTS	0.9792	0.9703	0.9473	0.9602	0.9721	0.9658
DACTYL-complete	0.9903	0.9603	0.9747	0.9793	0.9761	0.9761
DetectRL	0.9370	0.8910	0.8852	0.9054	0.8992	0.9036
Dolly-Cosmopedia	0.9958	0.9161	0.8953	0.9202	0.8726	0.9200
LLMTrace	0.9909	0.9730	0.9675	0.9735	0.9797	0.9769
MAGA-Bench	0.9992	0.9967	0.9945	0.9955	0.9964	0.9965
OriginalityAI	0.9322	0.7465	0.7820	0.8371	0.8273	0.8250
PAN 2025-Validation	0.9984	0.9159	0.8819	0.9174	0.8727	0.9172
Personagen+SE	1.0000	0.9825	0.9799	0.9822	0.9723	0.9834
RealDet	0.9540	0.9158	0.9178	0.9314	0.9284	0.9295
UChicago	0.9784	0.9365	0.9181	0.9365	0.9653	0.9470
PAN 2025 Test	0.984	0.925	0.89	0.921	0.899	0.924
PAN 2026 Test	0.994	0.962	0.941	0.958	0.946	0.96
PAN 2026 ELOQUENT Test	0.927	0.929	0.873	0.91	0.961	0.92

Table 9
ModernBERT-large with MCGrad post-hoc calibration test set results.

dataset	AUROC	F_1	C@1	Brier Score	$F_{0.5u}$	Overall
BEEMO	0.8619	0.9337	0.8826	0.9112	0.9350	0.9049
CoCONUTS	0.9797	0.9667	0.9420	0.9593	0.9785	0.9652
DACTYL-complete	0.9893	0.9509	0.9688	0.9757	0.9685	0.9707
DetectRL	0.9384	0.8869	0.8830	0.9085	0.9054	0.9044
Dolly-Cosmopedia	0.9931	0.9053	0.8804	0.9110	0.8573	0.9094
LLMTrace	0.9906	0.9723	0.9666	0.9722	0.9771	0.9758
MAGA-Bench	0.9988	0.9955	0.9925	0.9938	0.9949	0.9951
OriginalityAI	0.9303	0.7422	0.7860	0.8424	0.8462	0.8294
PAN 2025-Validation	0.9986	0.9559	0.9407	0.9551	0.9325	0.9566
Personagen+SE	1.0000	0.9909	0.9897	0.9887	0.9856	0.9910
RealDet	0.9586	0.9120	0.9152	0.9353	0.9327	0.9308
UChicago	0.9781	0.9332	0.9140	0.9394	0.9629	0.9455
PAN 2025 Test	0.984	0.95	0.928	0.946	0.939	0.949
PAN 2026 Test	0.993	0.975	0.962	0.97	0.969	0.974
PAN 2026 ELOQUENT Test	0.943	0.911	0.844	0.899	0.959	0.911

Both uncalibrated models (ModernBERT-large and DeBERTa-V3-large) have relatively high AUROC scores, even across OOD test sets — for example, we excluded DetectRL during training, but both models achieved an AUROC score of at least 0.93, highlighting some discrimination capacity. ModernBERT-large has an advantage in the overall score; on the PAN test sets, it achieves an overall score of at least 0.92. However, DeBERTa-V3-large shows more robustness in terms of AUROC score — on the ELOQUENT test set, it achieves an AUROC score of 0.982 in contrast with ModernBERT-large's 0.927. Our results indicate that a high discrimination capacity can hide miscalibrated scores — a model that achieves high AUROC is not guaranteed to have outputs that are calibrated. This discrepancy highlights the primary advantage of evaluating across multiple metrics.

ModernBERT-large with MCGrad yields improvements in the overall score in the PAN 2025 and 2026 test sets compared to the base model. However, it fails to give an improvement on the ELOQUENT test set, suggesting that the multicalibration approach is sensitive to distribution shifts. We confirm this on the 12 external test sets: we see that using the MCGrad and Bayesian BERT-tiny models shows an increase in the overall score for six of the test sets. The increase in score for the PAN test sets (excluding

Table 10
DeBERTa-V3-large test set results.

dataset	AUROC	F_1	C@1	Brier Score	$F_{0.5u}$	Overall
BEEMO	0.8780	0.9302	0.8785	0.9041	0.9418	0.9065
CoCONUTS	0.9819	0.9518	0.9180	0.9296	0.9785	0.9519
DACTYL-complete	0.9846	0.9524	0.9698	0.9747	0.9702	0.9703
DetectRL	0.9465	0.8759	0.8756	0.8940	0.9124	0.9009
Dolly-Cosmopedia	0.9952	0.9274	0.9107	0.9386	0.8905	0.9325
LLMTrace	0.9903	0.9738	0.9686	0.9739	0.9842	0.9782
MAGA-Bench	0.9992	0.9966	0.9944	0.9956	0.9974	0.9967
OriginalityAI	0.9213	0.6551	0.7425	0.7741	0.8227	0.7831
PAN 2025-Validation	0.9985	0.9917	0.9894	0.9919	0.9951	0.9933
Personagen+SE	1.0000	0.9973	0.9970	0.9976	0.9958	0.9975
RealDet	0.9810	0.9389	0.9418	0.9518	0.9670	0.9561
UChicago	0.9817	0.8989	0.8756	0.8981	0.9563	0.9221
PAN 2025 Test	0.987	0.975	0.965	0.969	0.987	0.977
PAN 2026 Test	0.989	0.843	0.797	0.852	0.93	0.882
PAN 2026 ELOQUENT Test	0.982	0.865	0.774	0.822	0.941	0.877

Table 11
Final leaderboard results for the PAN 2026 test set for teams that achieved higher overall scores compared to the zero-knowledge baseline.

Rank	Team	AUROC	F_1	C@1	Brier Score	$F_{0.5u}$	Overall
1	suspiciously-coherent	0.989	0.976	0.965	0.965	0.980	0.975
2	**DACTYL (ModernBERT-large + MCGrad)**	0.993	0.975	0.962	0.970	0.969	0.974
-	*Baseline mdok (PAN 2025 Winner)*	0.995	0.963	0.946	0.951	0.984	0.968
3	plagiarism-detector-com-reg2	0.996	0.899	0.935	0.941	0.957	0.946
4	dsgt-pan-26	0.981	0.907	0.873	0.885	0.961	0.921
5	ydong	0.930	0.928	0.886	0.915	0.896	0.911
6	turingteam	0.957	0.851	0.846	0.882	0.925	0.892
7	writerslogic-inc	0.891	0.902	0.853	0.891	0.899	0.887
8	a-a-detection	0.852	0.885	0.826	0.848	0.880	0.858
-	*Baseline PPMd*	0.783	0.865	0.783	0.810	0.828	0.814
9	ak-ys	0.911	0.774	0.723	0.754	0.891	0.811
10	ikr3	0.814	0.856	0.749	0.779	0.788	0.797
11	radar	0.772	0.800	0.722	0.805	0.839	0.788
12	sinai	0.624	0.856	0.748	0.771	0.788	0.757
13	egrantha-ai	0.500	0.856	0.749	0.749	0.788	0.728
-	*Baseline Zero-knowledge*	0.500	0.856	0.749	0.749	0.788	0.728

ELOQUENT) might indicate that those test sets represent a distribution shift that our MCGrad model covers.

In comparison to other classifiers, our ModernBERT-large and MCGrad classifier ranks second in overall score and is one of two models to beat the PAN 2025 winning system. The MCGrad system ranks second in AUROC, F_1, C@1, $F_{0.5u}$, and first in Brier score among competing teams.

6. Conclusion

Our work applies Bayesian neural networks (as a classification head) to help select candidate texts to use for downstream training. Additionally, these same classifiers can help improve calibration

of larger, non-Bayesian models via their uncertainties (standard deviations). We also point out that both ModernBERT-large and DeBERTa-V3-large have strong generalization capacities given their high AUROC scores across OOD test sets. Using the Bayesian BERT-tiny models' uncertainties as inputs for an MCGrad model demonstrates mixed results on external test sets but an increase in the overall score on the PAN test sets. Note that our Bayesian BERT-tiny hyperparameters for the BNN layer were manually tuned — we suspect that hyperparameter optimization via some proxy uncertainty metric could lead to better uncertainty estimates, which could improve the MCGrad model.

Future work might explore applying this Bayesian data filtering technique and EXM on other text classification tasks, such as prompt safety detection.

Resources Used

Training Bayesian BERT-tiny models and inference for the classifiers was done on a NVIDIA GB10. We used Lambda's GH200 instances for training ModernBERT-large and DeBERTa-V3-large [57].

Declaration on Generative AI

The author has used Grammarly to perform grammar checking in preparing this paper.

References

[1] J. Bevendorff, M. Fröbe, A. Greiner-Petter, A. Jakoby, M. Mayerl, P. Nakov, H. Plutz, M. Ta, Y. Wang, E. Zangerle, Overview of PAN 2026: Voight-Kampff Generative AI Detection, Text Watermarking, Multi-author Writing Style Analysis, Generative Plagiarism Detection, and Reasoning Trajectory Detection – Extended Abstract, in: R. Campos, A. Jatowt, Y. Lan, M. Aliannejadi, C. Bauer, S. MacAvaney, Z. Ren, S. Verberne, N. Bai, M. Mansoury (Eds.), Advances in Information Retrieval. 48th European Conference on IR Research (ECIR 2026), volume 16485 of *Lecture Notes in Computer Science*, Springer Nature, Cham, Switzerland, 2026, pp. 225–232. doi:10.1007/978-3-032-21321-1_32.

[2] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241. URL: https://link.springer.com/chapter/10.1007/978-3-031-28241-6_20. doi:10.1007/978-3-031-28241-6_20.

[3] J. Bevendorff, M. Fröbe, A. Greiner-Petter, A. Jakoby, M. Mayerl, P. Nakov, H. Plutz, M. Potthast, B. Stein, M. N. Ta, Y. Wang, E. Zangerle, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, A. Barrón-Cedeño, A. G. S. de Herrera, E. S. Salido, S. MacAvaney, J. M. Struß (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2026.

[4] J. Bevendorff, R. R. Gunti, J. Karlgren, M. Fröbe, M. Potthast, B. Stein, Overview of the third "Voight-Kampff" Generative AI / LLM Detection Task at PAN and ELOQUENT 2026, in: A. Barrón-Cedeño, A. G. S. de Herrera, E. S. Salido, S. MacAvaney, J. M. Struß (Eds.), Working Notes of CLEF 2026 – Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2026.

[5] S. Thorat, A. Caines, Dactyl: Diverse adversarial corpus of texts yielded from large language models, arXiv preprint arXiv:2508.00619 (2025).

[6] L. Dugan, A. Hwang, F. Trhlík, A. Zhu, J. M. Ludan, H. Xu, D. Ippolito, C. Callison-Burch, 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), 2024, pp. 12463–12492.

[7] R. Compton, L. Zhang, A. Puli, R. Ranganath, When more is less: Incorporating additional datasets can hurt performance by introducing spurious correlations, in: Machine learning for healthcare conference, PMLR, 2023, pp. 110–127.
[8] B. Emi, M. Spero, Technical report on the pangram ai-generated text classifier, arXiv preprint arXiv:2402.14873 (2024).
[9] M. Korakakis, A. Vlachos, A. Weller, Mitigating shortcut learning with interpolated learning, in: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 9191–9206.
[10] Z. Yuan, D. Zhu, Z.-H. Qiu, G. Li, X. Wang, T. Yang, Libauc: A deep learning library for x-risk optimization, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '23, ACM, 2023, p. 5487–5499. URL: http://dx.doi.org/10.1145/3580305.3599861. doi:10.1145/3580305.3599861.
[11] B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen, G. T. Adams, J. Howard, I. Poli, Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar (Eds.), Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 2526–2547. URL: https://aclanthology.org/2025.acl-long.127/. doi:10.18653/v1/2025.acl-long.127.
[12] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled attention, in: International Conference on Learning Representations, 2021. URL: https://openreview.net/forum?id=XPZIaotutsD.
[13] N. Tax, L. Perini, F. Linder, D. Haimovich, D. Karamshuk, N. Okati, M. Vojnovic, P. A. Apostolopoulos, MCGrad: Multicalibration at Web Scale, in: Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1 (KDD 2026), 2026. doi:10.1145/3770854.3783954.
[14] J. Schler, M. Koppel, S. Argamon, J. W. Pennebaker, Effects of age and gender on blogging., in: AAAI spring symposium: Computational approaches to analyzing weblogs, volume 6, Stanford, CA, USA, 2006, pp. 199–205.
[15] G. Zhang, X. Du, Z. Yu, Z. Wang, Z. Wang, S. Guo, T. Zheng, K. Zhu, J. Liu, S. Yue, B. Liu, Z. Peng, Y. Yao, J. Yang, Z. Li, B. Zhang, M. Liu, T. Liu, Y. Gao, W. Chen, X. Zhou, Q. Liu, T. Wang, W. Huang, Finefineweb: A comprehensive study on fine-grained domain web corpus, 2024. URL: https://huggingface.co/datasets/m-a-p/FineFineWeb, version v0.1.0.
[16] F. Chiusano, 190k+ Medium Articles — kaggle.com, https://www.kaggle.com/datasets/fabiochiusano/medium-articles/data, 2022. [Accessed 17-05-2026].
[17] R. Saxena, F. Keller, MovieSum: An abstractive summarization dataset for movie screenplays, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 4043–4050. URL: https://aclanthology.org/2024.findings-acl.239/. doi:10.18653/v1/2024.findings-acl.239.
[18] L. Edwindra, GitHub - ledwindra/nber: Scrape and analyzie NBER working papers. — github.com, https://github.com/ledwindra/nber, 2025. [Accessed 17-05-2026].
[19] Yelp, Yelp Dataset — kaggle.com, https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset, 2022. [Accessed 17-05-2026].
[20] Anthropic, Introducing Claude Haiku 4.5 — anthropic.com, https://www.anthropic.com/news/claude-haiku-4-5, 2025. [Accessed 28-05-2026].
[21] Anthropic, Introducing Sonnet 4.6 — anthropic.com, https://www.anthropic.com/news/claude-sonnet-4-6, 2026. [Accessed 28-05-2026].
[22] OpenAI, Introducing GPT-5.4, https://openai.com/index/introducing-gpt-5-4/, 2026. [Accessed 28-05-2026].
[23] GLM-5-Team, :, A. Zeng, X. Lv, Z. Hou, Z. Du, Q. Zheng, B. Chen, D. Yin, C. Ge, C. Huang, C. Xie, C. Zhu, C. Yin, C. Wang, G. Pan, H. Zeng, H. Zhang, H. Wang, H. Chen, J. Zhang, J. Jiao, J. Guo, J. Wang, J. Du, J. Wu, K. Wang, L. Li, L. Fan, L. Zhong, M. Liu, M. Zhao, P. Du, Q. Dong, R. Lu,

Shuang-Li, S. Cao, S. Liu, T. Jiang, X. Chen, X. Zhang, X. Huang, X. Dong, Y. Xu, Y. Wei, Y. An, Y. Niu, Y. Zhu, Y. Wen, Y. Cen, Y. Bai, Z. Qiao, Z. Wang, Z. Wang, Z. Zhu, Z. Liu, Z. Li, B. Wang, B. Wen, C. Huang, C. Cai, C. Yu, C. Li, C. Hu, C. Zhang, D. Zhang, D. Lin, D. Yang, D. Wang, D. Ai, E. Zhu, F. Yi, F. Chen, G. Wen, H. Sun, H. Zhao, H. Hu, H. Zhang, H. Liu, H. Zhang, H. Peng, H. Tai, H. Zhang, H. Liu, H. Wang, H. Yan, H. Ge, H. Liu, H. Chu, J. Zhao, J. Wang, J. Zhao, J. Ren, J. Wang, J. Zhang, J. Gui, J. Zhao, J. Li, J. An, J. Li, J. Yuan, J. Du, J. Liu, J. Zhi, J. Duan, K. Zhou, K. Wei, K. Wang, K. Luo, L. Zhang, L. Sha, L. Xu, L. Wu, L. Ding, L. Chen, M. Li, N. Lin, P. Ta, Q. Zou, R. Song, R. Yang, S. Tu, S. Yang, S. Wu, S. Zhang, S. Li, S. Li, S. Fan, W. Qin, W. Tian, W. Zhang, W. Yu, W. Liang, X. Kuang, X. Cheng, X. Li, X. Yan, X. Hu, X. Ling, X. Fan, X. Xia, X. Zhang, X. Zhang, X. Pan, X. Zou, X. Zhang, Y. Liu, Y. Wu, Y. Li, Y. Wang, Y. Zhu, Y. Tan, Y. Zhou, Y. Pan, Y. Zhang, Y. Su, Y. Geng, Y. Yan, Y. Tan, Y. Bi, Y. Shen, Y. Yang, Y. Li, Y. Liu, Y. Wang, Y. Li, Y. Wu, Y. Zhang, Y. Duan, Y. Zhang, Z. Liu, Z. Jiang, Z. Yan, Z. Zhang, Z. Wei, Z. Chen, Z. Feng, Z. Yao, Z. Chai, Z. Wang, Z. Zhang, B. Xu, M. Huang, H. Wang, J. Li, Y. Dong, J. Tang, Glm-5: from vibe coding to agentic engineering, 2026. URL: https://arxiv.org/abs/2602.15763. arXiv:2602.15763.

[24] MiniMax, :, A. Chen, A. Li, B. Zhou, B. Gong, B. Jiang, B. Dan, C. Yu, C. Wang, C. Ma, C. Zhong, C. Zhu, C. Xiao, C. Yang, C. Du, C. Zhang, C. Zhang, C. Huang, C. Zhang, C. Du, C. Zhao, C. Guo, D. Chen, D. Ding, D. Sun, D. Zhang, E. Yang, F. Yu, G. Zheng, G. Zheng, G. Li, H. Zhu, H. Zhou, H. Zhang, H. Ding, H. Zhang, H. Sun, H. Lyu, H. Lu, H. Wang, H. Shi, H. Li, J. Chen, J. Zhang, J. Zhuang, J. Cai, J. Pan, J. Li, J. Song, J. Zhang, J. Wang, J. Gu, J. Zhu, J. Dong, J. Li, J. Zhang, J. Zhuang, J. Tian, J. Liu, J. Hu, J. Tao, J. Zhang, J. Ruan, J. Xu, J. Yan, J. Liu, J. He, K. Xu, K. Ji, K. Yang, K. Xiao, K. Duan, K. Li, L. Han, L. Ruan, L. Yuan, L. Yu, L. Feng, L. Mo, L. Li, L. Bao, L. Yang, L. Zhou, Loki, L. Chen, L. Ceng, M. Li, M. Zhong, M. Tao, M. Chi, M. Lin, N. Hu, N. Chen, P. Zhu, P. Gao, P. Gao, P. Li, P. Li, P. Zhao, Q. Ren, Q. Xu, Q. Ren, Q. Li, Q. Wang, Q. Chen, Q. Ceng, R. Tian, R. Dong, R. Leng, R. Zhang, S. Liu, S. Chen, S. Jia, S. Yao, S. Zhao, S. Yu, S. Li, S. Pan, S. Zhu, T. Li, T. Xie, T. Qin, T. Liang, W. Liu, W. Xu, W. Li, W. Chen, W. Cheng, W. Zhang, W. Chen, W. Zhao, X. Chen, X. Song, X. Wang, X. Luo, X. Su, X. Li, X. Han, X. Wu, X. Song, X. Han, X. Guan, X. Lu, X. Zou, X. Lai, X. Li, Y. Gong, Y. Wang, Y. Xu, Y. Wang, Y. Tang, Y. Chen, Y. Qiu, Y. Shi, Y. Guo, Y. Huang, Y. Wang, Y. Hu, Y. Gao, Y. Zhang, Y. Ying, Y. Zhang, Y. Wang, Y. Song, Y. Yang, Y. Meng, Y. Miao, Y. Li, Y. Liu, Y. Hu, Y. Huang, Y. Li, Y. Huang, Y. Zhang, Y. Hong, Y. Xie, Y. Zhang, Y. Liao, Y. Shi, Y. Wenren, Z. Li, Z. Li, Z. Luo, Z. Jin, Z. Sun, Z. Zhou, Z. Su, Z. Li, Z. Zhu, Z. Peng, Z. Fan, Z. Zhang, Z. Xu, Z. Lv, Z. Xu, Z. He, Z. He, Z. Li, Z. Gao, Z. Wu, Z. Song, Z. Zhou, Z. Sun, Z. Huang, Z. Chen, Z. Ge, The minimax-m2 series: Mini activations unleashing max real-world intelligence, 2026. URL: https://arxiv.org/abs/2605.26494. arXiv:2605.26494.

[25] I. Labs, S. Khanna, S. Kharbanda, S. Li, H. Varma, E. Wang, S. Birnbaum, Z. Luo, Y. Miraoui, A. Palrecha, S. Ermon, A. Grover, V. Kuleshov, Mercury: Ultra-fast language models based on diffusion, 2025. URL: https://arxiv.org/abs/2506.17298. arXiv:2506.17298.

[26] OpenAI, Introducing GPT-5.4 mini and nano — openai.com, https://openai.com/index/introducing-gpt-5-4-mini-and-nano/, 2026. [Accessed 28-05-2026].

[27] K. Team, T. Bai, Y. Bai, Y. Bao, S. H. Cai, Y. Cao, Y. Charles, H. S. Che, C. Chen, G. Chen, H. Chen, J. Chen, J. Chen, J. Chen, J. Chen, K. Chen, L. Chen, R. Chen, X. Chen, Y. Chen, Y. Chen, Y. Chen, Y. Chen, Y. Chen, Y. Chen, Y. Chen, Z. Chen, Z. Chen, D. Cheng, M. Chu, J. Cui, J. Deng, M. Diao, H. Ding, M. Dong, M. Dong, Y. Dong, Y. Dong, A. Du, C. Du, D. Du, L. Du, Y. Du, Y. Fan, S. Fang, Q. Feng, Y. Feng, G. Fu, K. Fu, H. Gao, T. Gao, Y. Ge, S. Geng, C. Gong, X. Gong, Z. Gongque, Q. Gu, X. Gu, Y. Gu, L. Guan, Y. Guo, X. Hao, W. He, W. He, Y. He, C. Hong, H. Hu, J. Hu, Y. Hu, Z. Hu, K. Huang, R. Huang, W. Huang, Z. Huang, T. Jiang, Z. Jiang, X. Jin, Y. Jing, G. Lai, A. Li, C. Li, C. Li, F. Li, G. Li, G. Li, H. Li, H. Li, J. Li, J. Li, J. Li, L. Li, M. Li, W. Li, W. Li, X. Li, X. Li, Y. Li, Y. Li, Y. Li, Y. Li, Z. Li, Z. Li, W. Liao, J. Lin, X. Lin, Z. Lin, Z. Lin, C. Liu, C. Liu, H. Liu, L. Liu, S. Liu, S. Liu, S. Liu, T. Liu, T. Liu, W. Liu, X. Liu, Y. Liu, Y. Liu, Y. Liu, Y. Liu, Y. Liu, Z. Liu, Z. Liu, E. Lu, H. Lu, Z. Lu, J. Luo, T. Luo, Y. Luo, L. Ma, Y. Ma, S. Mao, Y. Mei, X. Men, F. Meng, Z. Meng, Y. Miao, M. Ni, K. Ouyang, S. Pan, B. Pang, Y. Qian, R. Qin, Z. Qin, J. Qiu, B. Qu, Z. Shang, Y. Shao, T. Shen, Z. Shen, J. Shi, L. Shi, S. Shi, F. Song, P. Song, T. Song, X. Song, H. Su, J. Su, Z. Su, L. Sui, J. Sun, J. Sun, T. Sun, F. Sung, Y. Tai, C. Tang, H. Tang, X. Tang, Z. Tang, J. Tao, S. Teng, C. Tian,

P. Tian, A. Wang, B. Wang, C. Wang, C. Wang, C. Wang, D. Wang, D. Wang, D. Wang, F. Wang, H. Wang, H. Wang, H. Wang, H. Wang, H. Wang, J. Wang, J. Wang, J. Wang, K. Wang, L. Wang, Q. Wang, S. Wang, S. Wang, S. Wang, W. Wang, X. Wang, X. Wang, Y. Wang, Y. Wang, Y. Wang, Y. Wang, Y. Wang, Y. Wang, Z. Wang, Z. Wang, Z. Wang, Z. Wang, Z. Wang, Z. Wang, C. Wei, M. Wei, C. Wen, Z. Wen, C. Wu, H. Wu, J. Wu, R. Wu, W. Wu, Y. Wu, Y. Wu, Y. Wu, Z. Wu, C. Xiao, J. Xie, X. Xie, Y. Xie, Y. Xin, B. Xing, B. Xu, J. Xu, J. Xu, J. Xu, L. H. Xu, L. Xu, S. Xu, W. Xu, X. Xu, X. Xu, Y. Xu, Y. Xu, Y. Xu, Z. Xu, Z. Xu, J. Yan, Y. Yan, G. Yang, H. Yang, J. Yang, K. Yang, N. Yang, R. Yang, X. Yang, X. Yang, Y. Yang, Y. Yang, Y. Yang, Z. Yang, Z. Yang, Z. Yang, H. Yao, D. Ye, W. Ye, Z. Ye, B. Yin, C. Yu, L. Yu, T. Yu, T. Yu, E. Yuan, M. Yuan, X. Yuan, Y. Yue, W. Zeng, D. Zha, H. Zhan, D. Zhang, H. Zhang, J. Zhang, P. Zhang, Q. Zhang, R. Zhang, X. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, Z. Zhang, C. Zhao, F. Zhao, J. Zhao, S. Zhao, X. Zhao, Y. Zhao, Z. Zhao, H. Zheng, R. Zheng, S. Zheng, T. Zheng, J. Zhong, L. Zhong, W. Zhong, M. Zhou, R. Zhou, X. Zhou, Z. Zhou, J. Zhu, L. Zhu, X. Zhu, Y. Zhu, Z. Zhu, J. Zhuang, W. Zhuang, Y. Zou, X. Zu, Kimi k2.5: Visual agentic intelligence, 2026. URL: https://arxiv.org/abs/2602.02276. arXiv:2602.02276.

[28] TogetherAI, Together AI | The AI Native Cloud — together.ai, https://www.together.ai/, 2026. [Accessed 28-05-2026].

[29] FireworksAI, Fireworks AI - Fastest Inference for Generative AI — fireworks.ai, https://fireworks.ai/, 2026. [Accessed 28-05-2026].

[30] I. Tolstykh, A. Tsybina, S. Yakubson, M. Kuprashevich, Llmtrace: A corpus for classification and fine-grained localization of ai-written text, arXiv preprint arXiv:2509.21269 (2025).

[31] C. Gugliotta, L. La Cava, A. Tagarelli, Personagen: A persona-driven open-ended machine-generated text dataset, in: Proceedings of the 34th ACM International Conference on Information and Knowledge Management, 2025, pp. 6397–6401.

[32] HuggingFaceGECLM, HuggingFaceGECLM/StackExchange_Mar2023 Â· Datasets at Hugging Face — huggingface.co, https://huggingface.co/datasets/{H}ugging{F}ace{G}{E}{C}{L}{M}/{S}tack{E}xchange_{M}ar2023, 2023. [Accessed 17-05-2026].

[33] A. Song, Y. Cheng, Y. Xu, R. Feng, Maga-bench: Machine-augment-generated text via alignment detection benchmark, arXiv preprint arXiv:2601.04633 (2026).

[34] J. Wu, R. Zhan, D. F. Wong, S. Yang, X. Yang, Y. Yuan, L. S. Chao, Detectrl: Benchmarking llm-generated text detection in real-world scenarios, Advances in Neural Information Processing Systems 37 (2024) 100369–100401.

[35] Z. Wang, M. Xiong, J. Lian, Z. Dou, Reasoning-aware aigc detection via alignment and reinforcement, 2026. URL: https://arxiv.org/abs/2604.19172. arXiv:2604.19172.

[36] L. L. Cava, D. Costa, A. Tagarelli, Is contrasting all you need? contrastive learning for the detection and attribution of ai-generated text, in: U. Endriss, F. S. Melo, K. Bach, A. J. B. Diz, J. M. Alonso-Moral, S. Barro, F. Heintz (Eds.), ECAI 2024 - 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024), volume 392 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2024, pp. 3179–3186. URL: https://doi.org/10.3233/FAIA240862. doi:10.3233/FAIA240862.

[37] J. Bevendorff, D. Dementieva, M. Fröbe, B. Gipp, A. Greiner-Petter, J. Karlgren, M. Mayerl, P. Nakov, A. Panchenko, M. Potthast, et al., Overview of pan 2025: Generative ai detection, multilingual text detoxification, multi-author writing style analysis, and generative plagiarism detection, in: European Conference on Information Retrieval, Springer, 2025, pp. 434–441.

[38] F. Hamborg, N. Meuschke, C. Breitinger, B. Gipp, news-please: A generic news crawler and extractor, in: Proceedings of the 15th International Symposium of Information Science, 2017, pp. 218–223. doi:10.5281/zenodo.4120316.

[39] R. Roy, N. Imanpour, A. Aziz, S. Bajpai, G. Singh, S. Biswas, K. Wanaskar, P. Patwa, S. Ghosh, S. Dixit, N. R. Pal, V. Rawte, R. Garimella, G. Jena, A. Sheth, V. Sharma, A. N. Reganti, V. Jain, A. Chadha, A. Das, A comprehensive dataset for human vs. ai generated text detection, 2026. URL: https://arxiv.org/abs/2510.22874. arXiv:2510.22874.

[40] A. Tripathi, EndLessTime/SAGE · Datasets at Hugging Face — huggingface.co, https://huggingface.co/datasets/{E}nd{L}ess{T}ime/{S}{A}{G}{E}, 2025. [Accessed 17-05-2026].
[41] M. Gromadzki, A. Wróblewska, A. Kaliska, On the effectiveness of llm-specific fine-tuning for detecting ai-generated text, 2026. URL: https://arxiv.org/abs/2601.20006. arXiv:2601.20006.
[42] E. Artemova, J. Lucas, S. Venkatraman, J. Lee, S. Tilga, A. Uchendu, V. Mikhailov, Beemo: Benchmark of expert-edited machine-generated outputs, 2025. URL: https://arxiv.org/abs/2411.04032. arXiv:2411.04032.
[43] Y. Chen, J. Chen, G. Mo, X. Chen, B. He, X. Han, L. Sun, CoCoNUTS: Concentrating on Content while Neglecting Uninformative Textual Styles for AI-Generated Peer Review Detection, 2025. URL: https://arxiv.org/abs/2509.04460. arXiv:2509.04460.
[44] M. Conover, M. Hayes, A. Mathur, J. Xie, J. Wan, S. Shah, A. Ghodsi, P. Wendell, M. Zaharia, R. Xin, Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL: https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.
[45] L. Ben Allal, A. Lozhkov, G. Penedo, T. Wolf, L. von Werra, Cosmopedia, 2024. URL: https://huggingface.co/datasets/HuggingFaceTB/cosmopedia.
[46] OriginalityAI, GitHub - OriginalityAI/AI-detector-research-tool — github.com, https://github.com/{O}riginality{A}{I}/{A}{I}-detector-research-tool, 2023. [Accessed 18-05-2026].
[47] X. Zhu, Y. Ren, Y. Cao, X. Lin, F. Fang, Y. Li, Reliably bounding false positives: A zero-shot machine-generated text detection framework via multiscaled conformal prediction, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar (Eds.), Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 12298–12319. URL: https://aclanthology.org/2025.acl-long.601/. doi:10.18653/v1/2025.acl-long.601.
[48] B. Jabarian, A. Imas, Artificial writing and automated detection, Technical Report, National Bureau of Economic Research, 2025.
[49] P. Bhargava, A. Drozd, A. Rogers, Generalization in nli: Ways (not) to go beyond simple heuristics, 2021. arXiv:2110.01518.
[50] I. Turc, M. Chang, K. Lee, K. Toutanova, Well-read students learn better: The impact of student initialization on knowledge distillation, CoRR abs/1908.08962 (2019). URL: http://arxiv.org/abs/1908.08962. arXiv:1908.08962.
[51] P. Esposito, Blitz - bayesian layers in torch zoo (a bayesian deep learing library for torch), https://github.com/piEsposito/blitz-bayesian-deep-learning/, 2020.
[52] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight uncertainty in neural networks, 2015. URL: https://arxiv.org/abs/1505.05424. arXiv:1505.05424.
[53] S. Swayamdipta, R. Schwartz, N. Lourie, Y. Wang, H. Hajishirzi, N. A. Smith, Y. Choi, Dataset cartography: Mapping and diagnosing datasets with training dynamics, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 9275–9293. URL: https://aclanthology.org/2020.emnlp-main.746/. doi:10.18653/v1/2020.emnlp-main.746.
[54] LibAUC, libauc.losses — libauc 1.0.0 documentation — docs.libauc.org, https://docs.libauc.org/api/libauc.losses.html#libauc.losses.auc.tp{A}{U}{C}_{K}{L}_{L}oss, ???? [Accessed 28-06-2026].
[55] D. Zhu, G. Li, B. Wang, X. Wu, T. Yang, When auc meets dro: Optimizing partial auc for deep learning with non-convex convergence guarantee, in: International Conference on Machine Learning, PMLR, 2022, pp. 27548–27573.
[56] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272. doi:10.1038/s41592-019-0686-2.
[57] Lambda, The Superintelligence Cloud | Lambda — lambda.ai, https://lambda.ai/, 2026. [Accessed

29-05-2026].