Uncertainty Estimation Methods for Countering Attacks on Machine-Generated Text Detectors

Valeriy Levanov[1], **Anastasia Voznyuk**[1], and **Andrey Grabovoy**[1]

[1]Moscow Institute of Physics and Technology, Moscow

April 20, 2025

Abstract

In this work, we explore the use of uncertainty estimation methods to enhance the quality of machine-generated text detectors against various attacks, such as homoglyphs, paraphrasing and noise injection. These attacks are not only used to bypass detection but also to test the resilience of detectors. We will test the hypothesis that uncertainty estimation can provide a more sustainable approach, eliminating the need for constant retraining. The research will evaluate this hypothesis in two scenarios: when only the text is available and when access to the model's internal states is also provided. The planned experiments aim to validate the results and compare them with current state-of-the-art solutions.

1 Introduction

Recent advancements in large language models (LLMs) allow for the easy creation of coherent texts that are almost have not diffence from those written by humans. Despite the wide range of efficient applications of generative models for society, there is also a high risk of their misuse for spreading misinformation or solving students' homework. Therefore, there is a need for effective methods to discern machine-generated text from human-written text. This task is highly challenging due to the diversity of models, their generation styles and different texts domains. Also various types of attacks pose particular difficulties for detection, and even the simplest of these attacks can significantly reduce the accuracy of effective detectors.

The main approach to finding differences between texts will be **Uncertainty Estimation** (UE). Uncertainty estimation is a metric computed over text that represents a model's understanding about the given text. It can be calculated based on the logits of the context, hidden layers, or the output of the model's prediction for the text. This method has already been investigated for various NLP tasks such as machine translation (MT), text summarization (TS), and question answering (QA)[1]. Uncertainty estimation is also proven effective for detecting generated images by analyzing the distributions of natural images [2]. The use of uncertainty estimation for detecting machine-generated text has not been explored in the literature, which further motivates investigating this direction.

In this paper, we aim to combine an approach with uncertainty estimation for the task of detecting machine-generated text. We will examine whether the representations of the machine and human text models differ from each other. This research could serve as a foundation for developing attack-resistant detectors capable of identifying machine-generated text with high accuracy.

2 Problem statement

Formally, the problem can be described as follows:

Given a set of texts $\{x_n\}$ with binary labels $\{y_n\}$, where the labels indicate whether the text is machine-generated or human-written. The research will explore two approaches to calculating uncertainty.

2.1 White-box methods

In this case, the model $F(x)$ can be represented as a composition of mappings $f_1(x)$ and $f_2(h)$, where $f_1(x)$ maps the text to some internal representation of the model such as context logits or hidden states, and $f_2(h)$ maps this representation to the model's prediction. In the white-box method, for a specific text, we can observe both $f_1(x)$ and $F(x)$, which means that the model allows us to examine its internal states during prediction. The goal is to use uncertainty estimation methods to study whether texts with different labels cluster together. In this case, uncertainty can be estimated in various ways[1], which require some knowledge of the internal workings of the model.

Typically, three types of such methods are distinguished. The first type comprises **information-based** methods, which are computed from the model's context logits and rely on word prediction probabilities for each token. Another category is **ensemble-based methods**, requiring multiple distinct models to aggregate their output metrics; we exclude this approach due to prohibitive memory and computational costs. Finally, **density-based** methods analyze the distribution of hidden states h(x): by estimating h(x) distributions for labeled texts, these methods evaluate whether new texts align with (implying matching labels) or deviate from (indicating label mismatch) the learned distributions.

2.2 Black-box methods

In many modern models, internal states and architecture are not available for study, but even in this case there are many methods for estimating uncertainty based only on the response model $F(x)$. Black-box methods were also used in the work[1]. The task is to study these methods and compare their effectiveness with the white-box methods.

2.3 Classification models

After calculating the metrics, classification needs to be performed based on them. The main objective is to achieve relatively good metric scores (ROC-AUC, accuracy) while spending minimal time on training. Therefore, we will focus on models capable of effectively performing classification on numerical data. Suitable candidates include: Logistic Regression, Neural Network Classifiers, Random Forest and Gradient Boosting.

3 Method

3.1 Uncertainty Calculation

Uncertainty estimation is a key method we use to analyze texts in our paper. One of the main approaches is information-based methods that work with the model's output probabilities. These methods include:

1) **Perplexity (PPL)** - measures how surprised the model is by the text:

$$PPL = \exp\left(-\frac{1}{L}\sum_{l=1}^{L}\log P(w_l|w_{<l})\right)$$

2) **Mean token entropy** - calculates average uncertainty per token:

$$H = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j} P(w_j|w_{<i})\log P(w_j|w_{<i})$$

3) **Monte Carlo Sequence Entropy** - we run the model multiple times with small random changes and average the results:

$$H_S(x;\theta) = -\frac{1}{K}\sum_{k=1}^{K}\log P(y^{(k)}|x,\theta)$$

Another approach uses **density-based methods**. We first analyze the model's hidden states from the training data - calculating their mean and variance. Then, we compare these learned patterns to the hidden states of new test texts to measure how different they are.

The main method here is **Mahalanobis Distance**:

$$MD(x) = (h(x)-\mu)^T\Sigma^{-1}(h(x)-\mu)$$

3.2 LLM

It is very important to choose the model through which the logits of the text contexts will be obtained. The choice fell on **Llama-3-8B-Instruct**. Its main advantages are its relative lightness and access to the internal representations of the context. Text will be sent to the input of the model without additional requests. The interesting output of the model is the context logits, and we will usually focus on the k most likely logits to simplify calculations.

3.3 Datasets

The basis of the research is the availability of extensive datasets. An important requirement for them will be the presence of texts from multiple domains and several generation models. These qualities help to increase the robustness and accuracy of the detectors.

M4GT[3] is designed for the task of binary classification and contains 65,177 human-written texts and 73,288 machine-generated texts. It includes several domains (Reddit, wikiHow, Wikipedia) and generative models (GPT-4, Cohere, Dolly).

RAID[4] is a massive dataset of 6 million text examples from a wide range of domains and models. During generation, various decoding strategies and repetition penalties were used, significantly increasing text diversity. Its key feature is the inclusion of adversarial attacks on texts, which aids in training robust detectors.

3.4 Binare Classification Models

For our baseline text classification, we'll use an untrained RobertaForSequenceClassification model fine-tuned for one epoch to compare training times while expecting high accuracy. For uncertainty-based classification, we implement three approaches: (1) logistic regression as the simplest method, (2) a RandomForestClassifier with 300 trees (max depth of tree = 10), and (3) a neural network classifier consisting of 4 linear layers with BatchNorm and Dropout regularization on each layer, optimized using Adam with Binary Cross-Entropy loss ($L = -\frac{1}{N}\sum[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$), trained for 300 epochs to ensure convergence. This setup allows comprehensive comparison of both traditional and neural approaches while controlling for computational efficiency.

4 Computational Experiment

The primary objective is to compare binary classification based on uncertainty estimates with alternative methods for detecting machine-generated text. To do this, we will use part of the M4GT[3] dataset as data. The dataset comprises 18,000 texts, with two-thirds generated uniformly by six different models and the remaining one-third consisting of human-written texts. All texts were sourced from arXiv.

Metric	Label	Mean	Median	Std	Min	Max
mean entropy	Human	0.0038	0.0038	0.0006	0.0010	0.0066
mean entropy	Machine	0.0034	0.0033	0.0007	0.0007	0.0061
perplexity	Human	2.2668	2.2414	0.2887	1.2250	4.1787
perplexity	Machine	2.1129	2.0516	0.3197	1.1083	3.8121
mc entropy	Human	232.603	224.375	54.179	63.7188	530.000
mc entropy	Machine	175.064	165.625	66.8385	19.2969	550.000
mahalanobis	Human	125.960	124.123	23.165	72.2812	494.407
mahalanobis	Machine	149.813	143.992	29.6482	92.4591	653.824

Table 1: Metrics for human and machine-generated texts

We processed texts from both labels using the LLM (Llama-3.1-8B-Instruct) and utilized contextual logits and hidden states to compute four metrics: perplexity, mean token entropy, monte carlo sequence entropy and mahalanobis distance. The results can be seen in Table 1. Some metrics show pronounced variation between human- and machine-generated texts. This finding is promising, as it suggests that the distinction between MG and HW text distributions can potentially be detected using uncertainty estimation (UE). In the following sections, we explore this idea further to develop a more robust detection framework.

The next step involves training and validating the proposed classification models using our UE.

Model	**Accuracy**	**ROC-AUC**	**Train Time (s)**
BERT Classifier	0.9942	0.9954	1489.0528
Neural Classifier with uncertainty	0.8183	0.7942	208.9576
Random forest with uncertainty	0.8103	0.7831	6.7727
Logistic Regression with uncertainty	0.7744	0.7317	0.0134

Table 2: Performance comparison of classification approaches with uncertainty estimation on data from arXiv

Table 2 presents the results. The results demonstrate that we have developed an effective approach for text classification with minimal training time requirements. As evidenced by our experiments, the Random Forest classifier achieves competitive ROC-AUC performance (0.7831) while requiring less than 10 seconds for training. In stark contrast, fine-tuning the LLM-based model demands substantially more computational resources, with a training time exceeding 1400 seconds - approximately 200 times slower than our uncertainty-based Random Forest implementation.. Therefore, by selecting a different classifier model, performance could be improved. However, it is noteworthy that the training time of our classifier is significantly shorter.

5 Conclusion

Our results demonstrate that uncertainty estimation methods can effectively identify machine-generated texts, with our proposed model achieving good ROC-AUC performance (0.783) while requiring minimal training time (<10 seconds). The research potential in this field is substantial—with greater computational resources, we could expand uncertainty estimation through ensemble methods and analyze more diverse text sources, paving the way for next-generation detectors that combine high accuracy, low computational costs, and resilience against evolving generation techniques.

References

[1] A. Tsvigun A. Vazhentsev S. Petrakov E. Fadeeva, R. Vashurin. Lm-polygraph: Uncertainty estimation for language models, 2023.

[2] T. Liu Y. Cheung Bo Han X. Tian J. Nie, Y. Zhang. Detecting discrepancies between ai-generated and natural images using uncertainty, 2024.

[3] Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohanned Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, et al. M4gt-bench: Evaluation benchmark for black-box machine-generated text detection. *to appear in ACL 2024*, 2024.

[4] F. Trhlik J. M. Ludan A. Zhu H. Xu D. Ippolito C. Callison-Burch L. Dugan, A. Hwang. Raid: A shared benchmark for robust evaluation of machine-generated text detectors., 2024.