Detecting Subtle Differences between Human and Model Languages Using Spectrum of Relative Likelihood

Yang Xu[1,*], Yu Wang[2], Hao An[1], Zhichen Liu[1], Yongyuan Li[1]
[1]Department of Computer Science and Engineering,
Southern University of Science and Technology
[2]Digital Linguistics Lab, Department of Linguistics,
Bielefeld University
***Correspondence:** xuyang@sustech.edu.cn

Abstract

Human and model-generated texts can be distinguished by examining the magnitude of likelihood in language. However, it is becoming increasingly difficult as language model's capabilities of generating human-like texts keep evolving. This study provides a new perspective by using the relative likelihood values instead of absolute ones, and extracting useful features from the spectrum-view of likelihood for the human-model text detection task. We propose a detection procedure with two classification methods, supervised and heuristic-based, respectively, which results in competitive performances with previous zero-shot detection methods and a new state-of-the-art on short-text detection. Our method can also reveal subtle differences between human and model languages, which find theoretical roots in psycholinguistics studies. Our code is available at `https://github.com/CLCS-SUSTech/FourierGPT`.

1 Introduction

A recent focus in natural language generation is to develop effective methods of detecting model-generated texts from real human texts. This endeavor gives human users an opportunity to know whether a given text, e.g., an explanation, is either generated by language model or written by human, allowing them to decide how much trust to place in the explanation. The current most effective methods for this task utilize the likelihood information in text data (e.g.,see Verma et al., 2024; Venkatraman et al., 2024). More specifically, most existing methods rely on the observation that the *absolute* likelihood values of texts naturally distribute differently, depending on their sources. Examples include the early work like GLTR (Gehrmann et al., 2019) and more recent ones like DetectGPT (Mitchell et al., 2023) and Fast-DetectGPT (Bao et al., 2024).

However, existing methods have the following limitations: First, most work estimates likelihood as a **static** property of language, but overlooks the fact that human language processing is a **dynamic** process during which the likelihood of information under processing is bound to certain linguistic and cognitive constraints. For example, the trade-off between processing effort and likelihood of words (Levy, 2008; Smith and Levy, 2013), limited and selective context access (Gibson, 1998, 2000), activation decay (Lewis et al., 2006) and so on. Second, merely using the *absolute* likelihood values to distinguish generated and human texts is a tricky "cat and mouse game" – as models' capability of mimicking human language keeps growing, their productions would eventually become hardly distinguishable (Brown et al., 2020). Third, current methods are not computationally economical, because most of them need to run at least one time of inference on text with a fairly large language model.

We propose a human-model text detection approach that addresses the aforementioned issues and achieves better or comparable performance with existing methods. Our approach extracts features from the spectrum view of relative likelihood scores of texts, to capture the dynamic changes of likelihood in language. These features are used to design two types of classifiers, a supervised learning-based one and a heuristic-based zero-shot one, both of which reach impressive performances. The core idea is to obtain the spectrum of likelihood using the Fourier transform, which summarizes the complex patterns of likelihood change in time domain into a much more compact view that magnifies the subtle differences between different texts. It has a theoretical basis in psycholinguistic studies on surprisal (likelihood) distribution in natural language. Further, our method is still effective when likelihood scores are estimated by naïve n-gram models, which places much less computa-

tional cost. We name our approach FourierGPT, inspired by existing methods like DetectGPT and Fast-DetectGPT.

2 Related Work

2.1 Likelihood-based zero-shot text detection

"Zero-shot" means the text detection is cast not as a supervised classification task, but rather a statistics-based detection task. Early works directly use the magnitude of token-level likelihood scores. For example, Gehrmann et al. (2019) renders the likelihood value of tokens to human-readable color schemes, which creates good visual distinction between GPT-2 generated text and real human texts. Token ranks based on their log-likelihood scores (LogRank) are used for the same task (Solaiman et al., 2019).

Recent works develop more advanced statistics based on deeper insights into the distributional difference between human-created and model-generated languages in log-likelihood space. For example, Mitchell et al. (2023) finds that the probability distribution of model-generated text tends to lie under the areas of negative curvature of the log-likelihood function, and in contrast, human text tends not. Based on this finding, they propose DetectGPT, a zero-shot detection method that measures *perturbation discrepency*, the gap between an original text and its rewritten variant that maintains the same meaning. The assumption is that human text presents smaller gaps than model text. Bao et al. (2024) make substantial methodological improvements to DetectGPT and propose Fast-DetectGPT by replacing the probability curvature with conditional probability curvature, which broadly improves the detection accuracy and greatly shorten the computational time. Therefore, Fast-DetectGPT is the main state-of-the-art method compared with in this study. In nature, both Detect-GPT and Fast-DetectGPT find an empirical threshold for the variance of *absolute* likelihood values, which depend on the choice of the inference model.

Two other likelihood-based methods are also compared within this study: normalized log-rank perturbation (NPR) (Su et al., 2023) and divergence between multiple completions of a truncated passage (DNA-GPT) (Yang et al., 2024). Both rely on estimation of absolute likelihood to some extent.

2.2 Surprisal and likelihood of language

The way likelihood scores are defined in the previous section is equivalent to the concepts of "surprisal" and "information density", which are commonly used interchangeably in the psycholinguistics literature. Surprisal is known to reflect the cognitive load of processing a word, phrase, or sentence – it takes more effort and time to produce and comprehend units of higher surprisal, such as rare words (Hale, 2001). There is a preference in human language to keep the surprisal intensity evenly distributed in time, known as uniform information density (UID) (Jaeger, 2010), or entropy rate constancy (ERC) (Genzel and Charniak, 2002, 2003). This preference is an outcome of the speaker/writer's intention to make the listener's comprehension easier, which, therefore, draws a potential connection to the topic of this study – is this preference learned by language models? Another relevant work is Xu and Reitter (2017)'s finding that periodicity of surprisal exists in natural language, which can be captured by spectrum analysis methods and be used to predict the interaction outcome of dialogue partners.

Understanding the human mind's preference and tendency in handling surprisal/likelihood leads to new ideas for natural language generation techniques. For example, some recent endeavors build on top of the assumption that model-generated language appears more natural and human-like if it is generated through a decoding algorithm that follows the UID theory, such as the beam search algorithm as evidenced in Meister et al. (2020); or it falls under the so-called stable entropy zone (Arora et al., 2023); Meister et al. (2023) proposes locally typical sampling, which enforces the uniform distribution of likelihood during the generation process, and results in generated texts that are more aligned with human texts.

2.3 Evaluation of natural language generation with likelihood

The task of evaluating natural language generation (NLG) is essentially related to the text detection task. Therefore, likelihood (and its variants) is a natural option here. Early works in NLG often frame the evaluation equivalent to a detection task, which treats human text as gold-standard, and uses the "distance" from human text to measure the quality of generated text. For example, Ippolito et al. (2020) uses total probability as a measurement and

Holtzman et al. (2020) compares the generation perplexity and Zipf coefficient (Zipf, 1949) (closely related to LogRank) of texts from different sampling methods. These works are very similar in methodology to those reviewed in Section 2.1, only except that they did not emphasize detection accuracy, but focused on "quality control" of generation.

Some recent evaluation metrics compare model text with human text in high dimensional space, such as MAUVE (Pillutla et al., 2021). While this type of method does not directly use likelihood information, interesting correlations have been found between likelihood-based metrics. For example, Yang et al. (2023) proposes a novel evaluation metric called Fourier analysis of cross-entropy (FACE), which converts the cross-entropy scores (i.e., likelihood) to spectrum representations and then measures the distances in frequency-domain. The resulted distance scores can reflect generation qualities that are co-examined with other metrics, such as MAUVE, and align well with human judgements. This work indicates that with proper transformation on simple likelihood scores, rich insights about language use are viable. In fact, the text detection method proposed in this study is directly inspired by Yang et al.'s work (2023).

3 Method

The procedure of FourierGPT consists of three steps: 1) Estimate and normalize likelihood scores; 2) Carry out Fourier transform to get the spectrum view; 3) Conduct classification on the spectrum. The procedure is illustrated in Figure 1 with an example. Details of each step are described below.

3.1 Estimation and normalization of likelihood scores

We estimate the likelihood scores of text data with pretrained language models of various scales: Mistral-7B (Jiang et al., 2023), GPT-2 families (Radford et al., 2019), and a bigram language model trained from scratch. We use the implementation of bigram model from Heafield (2011) and train it on a subset of C4 dataset (Raffel et al., 2020). Raw likelihood scores are estimated by running a forward pass on the input text. Taking an input of N tokens, $t_1, \ldots, t_N$, the likelihood score of ith token s_i is computed by $s_i = -\log P(t_i|t_1, \ldots, t_{i-1})$, i.e., the negative log probability returned by the language model, which we call the *estimator* model.

Then we normalize the raw likelihood scores $s_1, \ldots, s_N$ within each sequence, obtaining the z-scored likelihood $\tilde{s}_1, \ldots, \tilde{s}_N$, in which $\tilde{s}_i = \frac{s_i - \mu}{\sigma}$, $\mu = \sum s_i / N$ and $\sigma = \sum (s_i - \mu)^2 / (N-1)$. Here we would like to stress that this seemingly trivial normalization step is actually *critical* for verifying the hypothesis of this study: the raw likelihood s_i's value depends on the choice of estimator – larger models usually result in smaller values (similar to perplexity); but the z-scored $\tilde{s}_i$ characterizes the relative level of likelihood within the range $(-\infty, \infty)$, which is less dependent on the estimator. As expected, the distribution of raw likelihood is highly skewed, while the z-scored likelihood is closer to normal distribution and gives better classification results in the following steps.

We also find that z-score normalization before the Fourier transform is supported by practices in signal processing. Reno et al. (2018) points out the secondary motion imaging artifact: when a time series consists of high and low-frequency components of different intensities, the vanilla Fourier transformed spectrum will be dominated by the low-frequency component, and normalization can eliminate this artifact.

3.2 Fourier transform

The next step is to obtain the spectrum view of the z-scored likelihood sequence $\tilde{s}_0, \ldots, \tilde{s}_{N-1}$ as input. We apply discrete Fourier transform (DFT) according to the following:

$$X(\omega_k) \triangleq \sum_{n=0}^{N-1} \tilde{s}_n e^{-j\omega_k n} \quad (1)$$

The result is a set of complex numbers $\mathcal{F} = \{X(\omega_k)\}|_{k=0,\ldots,N-1}$ as the frequency-domain representation of the input time-domain signal (likelihood scores), in which ω_k is the k-th frequency component. We change the starting index of $\tilde{s}_i$ to 0 because DFT requires $k = 0$ as the lowest frequency component. $X(\omega_k)$ is a complex number made up of real and imaginary parts, $X(\omega_k) = \text{Re}(X(\omega_k)) + \text{Im}(X(\omega_k))j$. The norm $\|X(\omega_k)\| = \sqrt{\text{Re}(X(\omega_k))^2 + \text{Im}(X(\omega_k))^2}$ represents the intensity of the kth component ω_k. Finally, we use the sequence $\{\|X(\omega_k)\|\}|_{k=0,\ldots,N-1}$ as the **spectrum-view of likelihood**, which provides features for the next classification step.

The range of ω_k is $[0, \pi]$, and its interpretation is not trivial. Based on an intuitive interpretation provided by Yang et al. (2023), we can roughly tell
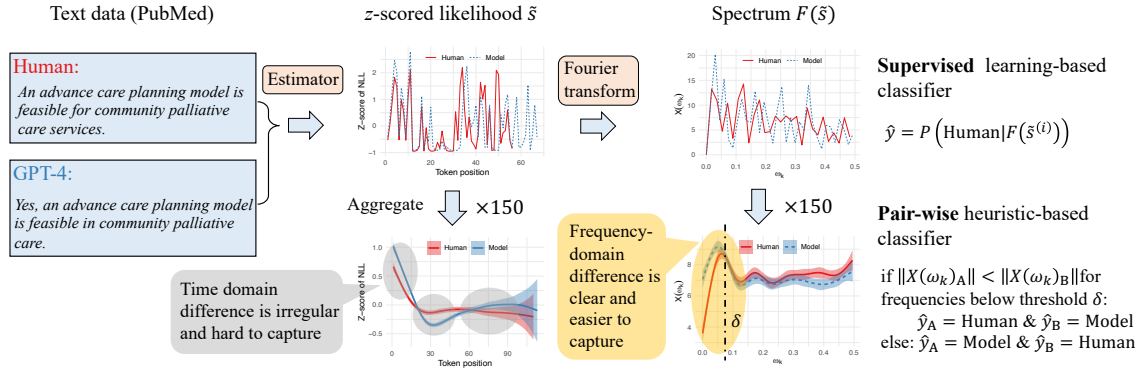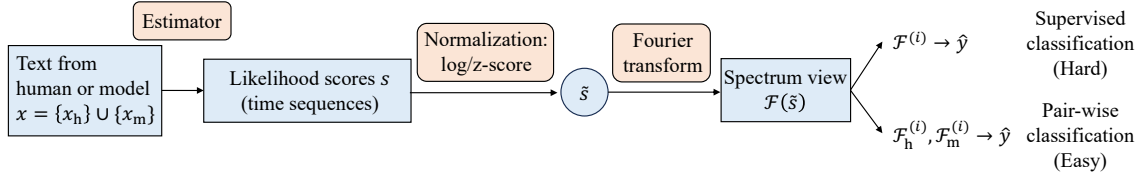
Figure 1: The procedure (above) and example (below) of FourierGPT.

that the likelihood score $\tilde{s}_i$ at the level of $\|X(\omega_k)\|$ tends to occur every $1/\omega_k$ tokens in the text data. Interestingly, we find the way $\|X(\omega_k)\|$ distributes along ω_k provides unique information to distinguish human from model. To develop solid explanations of what the spectrum of likelihood means is important, yet a different topic. We primarily focus on *harnessing* the spectrum information for the detection task, and try to do gain some interpretive insights at our best in Section 5.

3.3 Classification methods

We use two classification methods for the text detection task: A **supervised** learning-based classifier trained from the entire labeled spectrum data, which makes a binary prediction (human or model) on any given input spectrum representation; and a **pair-wise** heuristic-based classifier that tells which one is from human (hence the other one is from a model) in any given *pair* of input spectrum representations. For the pair-wise classifier, we require that the input pair must come from the same text *prompt*, which guarantees that one of them is from human and the other one is from model. It is obvious to see that the supervised classifier is more difficult to train as no prior information is given.

3.3.1 Supervised learning-based classifier

We train the supervised classifier using an augmented spectrum as input feature, which is obtained with multiple rounds of *circularization* operation on the likelihood scores: given an

original time series of likelihood scores $\mathcal{C}_0 = s_1, s_2, \ldots, s_n$, circularization at step T is to chop off the segment of length T at the head and then append it to the end, resulting in a new series $\mathcal{C}_T = s_{T+1}, \ldots, s_n, s_1, \ldots, s_T$. See the following complete procedure:

$$
\begin{array}{ll}
\text{Original scores } \mathcal{C}_0 & \rightarrow s_1, s_2, \ldots, s_n \\
\text{Circularized scores } \mathcal{C}_1 & \rightarrow s_2, \ldots, s_n, s_1 \\
\text{Circularized scores } \mathcal{C}_2 & \rightarrow s_3, \ldots, s_1, s_2 \\
\quad\quad\quad\vdots & \\
\text{Circularized scores } \mathcal{C}_{n-1} & \rightarrow s_n, s_1, \ldots, s_{n-1}
\end{array}
$$

Next, we apply Fourier transform to each circularized likelihood sequence, which produces n spectra in total, $\mathcal{F}(\mathcal{C}_t)$, $t = 0, \ldots, n-1$. The average spectrum $\overline{\mathcal{F}} = \frac{1}{n} \sum \mathcal{F}(\mathcal{C}_t)$ is used as the input feature for training the classifier. Lastly, we train several common types of classification models and evaluate their performances in Section 4.2. The circularization operation draws inspiration from the circular convolution in digital signal processing (Elliott, 2013). The intuition is: if a weak periodicity exists in the original "signal" $\mathcal{C}_0$, then obtaining multiple spectra from its multiple variants ($\mathcal{C}_1$ through $\mathcal{C}_{n-1}$) should amplify the periodicity that is undetectable otherwise. From a machine learning perspective, it is like a way of data augmentation, which picks the most salient features by aggregating multiple variants of the original data.

3.3.2 Pair-wise Heuristic-based classifier

We design a set of classifiers based on an empirical *heuristic* obtained by observing the difference between human and model's spectrum views: the likelihood spectrum presents a salient **difference at the low-frequency end**. The direction of difference slightly varies across dataset $\times$ model groups, but for most groups, the model's spectrum has a larger power amplitude than the human's, except for GPT-4 on Writing and Xsum (see Figure 2).

The heuristic is expressed as follows:

$$\left| \sum_{k=1}^{\delta_k} \|X^{\text{Human}}(\omega_k)\| - \sum_{k=1}^{\delta_k} \|X^{\text{Model}}(\omega_k)\| \right| > \varepsilon$$

in which $\delta_k \in \mathbb{Z}$ is an integer threshold defining the range of frequency components ω_k selected for comparing the spectrum power $\|X(\omega_k)\|$ s.t. $1 \leq k \leq \delta_k$, and its value is determined empirically in each dataset group. $\varepsilon \in \mathbb{R}$ is a real number threshold characterizing the observed difference in $\|X(\omega_k)\|$ between human and model, which is also determined empirically. A larger ε value means a more strict standard for distinguishing $\|X^{\text{Human}}(\omega_k)\|$ and $\|X^{\text{Model}}(\omega_k)\|$. In our experiments, we use $\varepsilon = 0$ for convenience.

4 Experiment Results

4.1 Datasets

We use the text detection datasets provided by Bao et al. (2024), which follows the experiment settings of Mitchell et al. (2023). It makes sure that all comparisons made to the previous methods are valid and consistent. The datasets text prompts are gathered from three sources: PubMedQA dataset (Jin et al., 2019) which consists of 273.5k human experts' answers to biomedical research questions; Reddit WritingPrompts dataset (Fan et al., 2018) which includes 300k human-written stories with prompts; XSum dataset (Narayan et al., 2018) which contains human-written summarization of 226.7k online articles in British Broadcasting Corporation (BBC). The datasets are compiled by Bao et al. (2024), using the OpenAI API[1]. Three APIs are used for generation: GPT-4, GPT-3.5 (Chat-GPT), and GPT-3 (Davinci). Each one of the three datasets (PubMed, Writing, and Xsum) contains 150 pairs of human and model texts. Each pair shares the first 30 tokens and differs afterward.

---

[1] https://openai.com/blog/openai-api

(a) Human and GPT-4
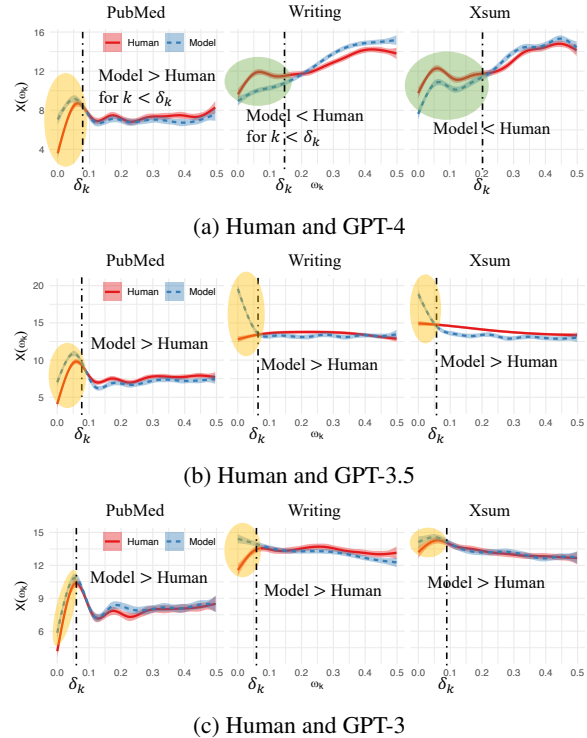
(b) Human and GPT-3.5

(c) Human and GPT-3

Figure 2: Heuristics for constructing pair-wise classifiers: Likelihood spectrum shows salient difference at low frequency components. Curves are fit using generative additive models (GAM). Shaded areas are 95% confidence intervals from bootstrap.

Therefore, our main experiments work on 3 (genres) $\times$ 3 (generation models) = 9 conditions.

4.2 Supervised learning-based classification

Six common classification models are trained and evaluated using 5-fold cross-validation, and we find that the Support Vector Machine (SVM) model achieves the best overall performance. The accuracy scores of SVM on all datasets are shown in Table 1. It can be seen that our method performs particularly well for the PubMed dataset, which achieves an above 80% accuracy score, as compared to the scores around 70% for the other two datasets. This indicates that the supervised classifier can learn features in short texts better than in longer ones.

Although our best scores on PubMed are lower than the state-of-the-art from Fast-DetectGPT (see Table 3), we think this is still an impressive result because it outperforms most of the other previous methods that use *absolute* likelihood scores for detection, and the gap from SOTA is small. We list the comparison on PubMed (GPT-4) and Writing (GPT-3) in Table 2.

Dataset	Gen. model	Best acc.	Classifier
PubMed	GPT-4	**0.8267**	SVM
	GPT-3.5	0.6000	SVM
	GPT-3	0.5800	HGBT
Writing	GPT-4	0.7167	NB
	GPT-3.5	0.7500	NB
	GPT-3	**0.8267**	SVM
Xsum	GPT-4	0.7400	SVM
	GPT-3.5	0.7500	SVM
	GPT-3	0.7400	SVM

Table 1: Accuracy scores of supervised classifiers. All spectrum features used for training are based on likelihood scores estimated by GPT2-xl model.

Method	PubMed GPT-4	Writing GPT-3	Avg.
Likelihood	0.8104	0.8496	0.8300
LogRank	0.8003	0.8320	0.8162
DNA-GPT	0.7565	0.8354	0.7960
NPR	0.6328	0.7847	0.7088
DetectGPT	0.6805	0.7818	0.7312
Fast-Detect	**0.8503**	**0.9568**	0.9036
FourierGPT	0.8267†	0.8267	0.8267

Table 2: Accuracy of our best supervised classifiers compared to other likelihood-based zero-shot methods reported in (Bao et al., 2024) on selected two task subsets. Best scores are in bold, and † indicates second best.

4.3 Pair-wise heuristic-based classification

The pair-wise heuristic-based classification results are shown in Table 3. For a comprehensive comparison, we include the second-best performing open-source method Likelihood and a commercial detection solution GPTZero (Tian and Cui, 2023) in the table. Our method performs generally better on GPT-4 and GPT-3.5 groups: it outperforms the state-of-the-art Fast-DetectGPT on PubMed data, though not as good in Writing or Xsum. Yet, the performance on the latter two datasets is quite competitive to the second-best previous method.

Similar to the supervised classifier, we also experiment with heuristic-based classifiers using likelihood scores estimated from bigram models, whose accuracy results are shown in Table 4. It has surprisingly good performance on Writing data: the accuracy on Writing+GPT-3.5 reaches 0.9067,

which is better than Fast-DetectGPT.

5 Discussion: Text Features Affects Spectrum of Likelihood

The purpose of this section is to investigate why the spectrum view of relative likelihood scores can be used to distinguish texts from humans and models. What specific features in the text are reflected in the frequency-domain? Can we know more about what language models learned (and did not learn) from humans by reading their likelihood spectrum? With these questions in mind, we present some interesting patterns discovered.

5.1 Answers starting with "yes/no"

We find that in PubMed data, model-generated answers are much more likely to start with a fixed pattern of "*Yes*"/"*No*", while humans do not answer in this style at all (at least in the current data). The ratios of answers with this pattern are listed in Table 5. Since each model group comes with a different set of 150 human question/answer texts, the total odds of the human group is as low as 0/450.

This is an interesting finding because it indicates the tendency of models to generate texts of high certainty: when the prompt is in an explicit form like "*Question: ...*", then the model tends to address it first by giving a certain answer like "*Yes*" or "*No*". On the other side, human answers sound less confident and tend to avoid certainty. We conjecture that this finding could be due to the general tendency of human language to use more *hedging* and avoid over-confidence, particularly in face of difficult questions such as the highly professional ones in PubMed.

We use a simple ablation experiment on data to examine whether this subtle difference is reflected in the spectrum of likelihood. We remove the "*Yes*"/"*No*" at the beginning of the answer, re-computing the likelihood scores, and re-do the Fourier transform. Consequently, the spectrum of the model morphs in shape towards the direction of human (Figure 3a (left)): the altered GPT-4 data's low-frequency components drop, and the high ends rise, both towards the direction of human. GPT-3.5 and GPT-3 have much fewer data points containing "*Yes*"/"*No*", but a similar trend exists in the high end (see Figure 3b). To showcase the advantage of spectrum view, we also plot the z-scored likelihood against token position (Figure 3a (right)), which shows that removing "*Yes*"/"*No*" makes the likeli-

Dataset	Gen. model	FourierGPT	δ_k	Est. model	Fast-Detect	Likelihood	GPTZero
	GPT-4	**0.9133**	3	GPT2-xl	0.8503	0.8104	0.8482
PubMed	GPT-3.5	**0.9467**	2	Mistral	0.9021	0.8775	0.8799
	GPT-3	0.6867	5	Mistral	**0.7225**	0.5668	0.4246
	GPT-4	0.8467	23	GPT2-xl	**0.9612**	0.8553	0.8262
Writing	GPT-3.5	0.9200	30	Mistral	**0.9916**	0.9740	0.9292
	GPT-3	0.7200	6	Mistral	**0.9568**	0.8496	0.6009
	GPT-4	0.8733	29	GPT2-xl	0.9067	0.7980	**0.9815**
Xsum	GPT-3.5	0.9200	24	GPT2-xl	0.9907	0.9578	**0.9952**
	GPT-3	0.6067	13	GPT2-xl	**0.9396**	0.8370	0.4860

Table 3: Accuracy of pair-wise heuristic-based classifiers. The best accuracy, corresponding heuristic δ_k, and estimator model used are reported. We report the classification accuracy scores from three previous zero-shot text detection methods, including two open-source solutions, Fast-DetecGPT and Likelihood, and one commercial detector GPTZero. We report the scores directly from (Bao et al., 2024). Best scores are in bold.

Dataset	Best group	Best acc.	Avg. acc.
Pubmed	GPT-3	0.6733	0.6511
Writing	GPT-3.5	**0.9067**	0.7867
Xsum	GPT-3.5	0.7800	0.7289

Table 4: Accuracy of FourierGPT pair-wise classifiers using likelihood spectrum from bigram language model. The bold number performs better than Fast-DetectGPT.

Group	Start w/ "*Yes*"	Start w/ "*No*"
GPT-4	78/150	10/150
GPT-3.5	35/150	2/150
Davinci	32/150	32/150
Human	0/150	0/150

Table 5: Proportions of answers that start with "*Yes*"/"*No*" pattern in PubMed data.

hood curve flatter (thus, closer to human), but this change is not as easy to describe as the spectrum. In sum, the subtle differences between human and model languages, like the "*Yes*"/"*No*" use discussed here, can be reflected in likelihood space, and the spectrum view can capture this difference conveniently.

5.2 Text lengths effect

It is pointed out in previous work that zero-shot detectors are supposed to perform worse on short text because shorter text means fewer data points to compute the likelihood-based statistics (Bao et al., 2024). We examine the effect of text length on FourierGPT's performance, by using only the first $n = 50, 100, 150$ tokens for the entire classification procedure on Writing and Xsum datasets.
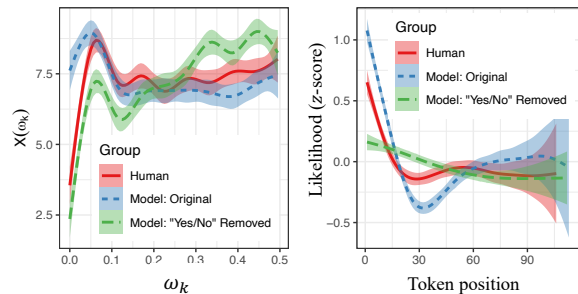
(a) Estimated by GPT-4.
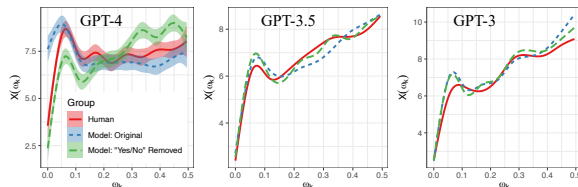
(b) Estimated by all three models.

Figure 3: (a) The changes of likelihood spectrum ((a) left) and likelihood-position plot ((a) right) after removing the "*Yes*"/"*No*" in answer from PubMed data (estimated by GPT-4 only). (b) The changes of spectrum estimated by all three models. Curves are fit with GAM. Shaded areas are 95% confidence intervals from bootstrap.

As PubMed data are already short, with the mean length of the answer part being $n = 35.2$ words, they are not included in the experiment.

From Figure 4 it can be seen that shorter texts indeed result in more indistinguishable spectrum shapes between human and model. Surprisingly, however, when we use the cut-off token count $n = 150$ on Writing data, the pair-wise classifier's accuracy increases by a significant percentage, even better than using full length. It strengthens the finding on PubMed that likelihood spectrum
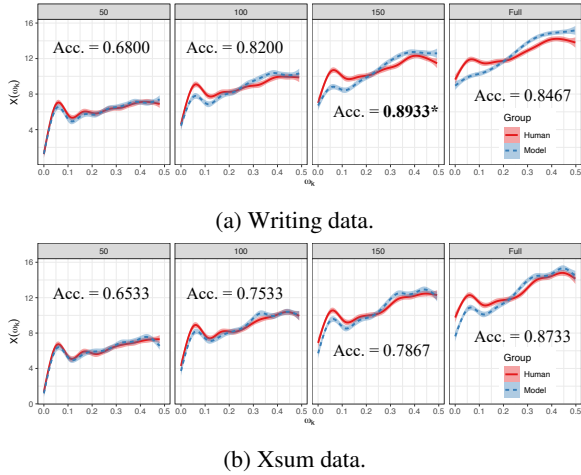
(a) Writing data.

(b) Xsum data.

Figure 4: Text lengths affect likelihood spectrum and pairwise classifier performance. Each plot corresponds to lengths of text, $n = 50, 100, 150$, compared to "Full". Curves are fit with GAM. Shaded areas are 95% confidence intervals from bootstrap.

better captures the characteristics of short texts.

5.3 Part-of-speech masking

As the last part of discussion, we investigate the role played by words of different part-of-speech (POS) tags in affecting the likelihood spectrum. Many previous studies have revealed that adding POS information in either the encoder or decoder side of (multimodal) language models can enhance language understanding and generation tasks (e.g., see Wray et al., 2019; Deshpande et al., 2019). Yang et al. (2022) further demonstrated that by informing the sampling method of different POS patterns, the generated texts show more diversity without sacrificing quality. The above findings suggest that, POS patterns as linguistic knowledge can significantly impact the NLG process.

Based on this, we use masking to break some selected POS patterns during the generation process, which will restrain the estimated likelihood from encoding complete information of POS, and then see how this will affect the likelihood spectrum. First, we mask three POS tags in text: 'NOUN', 'VERB', and 'ADJ', individually; and the union of the three, 'NOUN+VERB+ADJ' (thus, 'NVA'). Then, the masked tokens' likelihood scores are replaced with the average score, thus eliminating the contribution from that specific POS tag. Masking is done use the spaCy (Honnibal et al., 2020) POS tagger. The likelihood spectrum results after 'NVA' being masked is shown in Figure 5.

We find an interesting phenomenon: after ap-
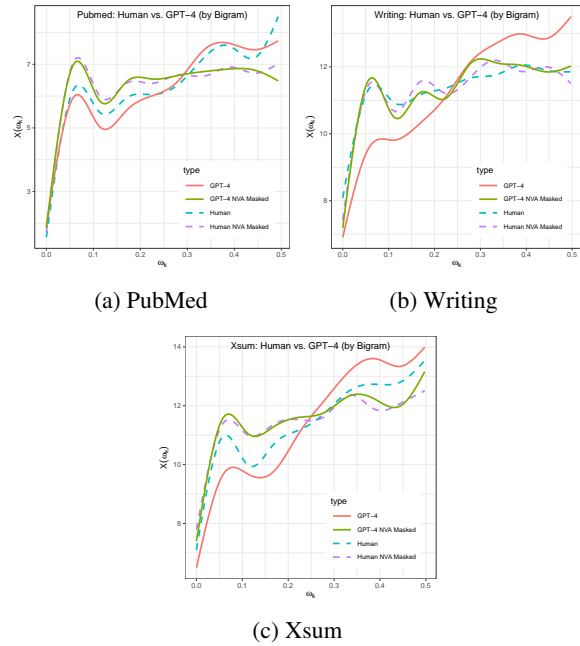

(a) PubMed

(b) Writing

(c) Xsum

Figure 5: Likelihood spectrum before and after masking out the 'NOUN+VERB+ADJ' POS tags. It can be seen that GPT-4 texts show greater variation before and after the mask, while human text show smaller change. Likelihood is estimated with a bigram model. Curves are fit with GAM. Shaded areas are 95% confidence intervals from bootstrap.

plying the POS mask, the change of likelihood spectrum for human text is relatively small, while for the model text (GPT-4), the change is much bigger (see Figure 5). Such difference is not limited to the collection of 'NVA' tags alone, but also observed for the 'NOUN', 'VERB', and 'ADJ' POS tags individually (shown from Figure 6 to Figure 8 in Appendix). This difference is most salient if bigram is used as the estimator[2] as compared to GPT-2. We back up the observation by calculating the *spectral overlap* between the original spectrum and the one after POS masking, for both human and model texts. It is a common metric for measuring spectrum similarity, and it was first used by Yang et al. (2023) as a means for NLG evaluation. It turns out that indeed the generated text has smaller before-after spectral overlap, as compared to human text, which indicates that the model spectrum is less stable against POS masking (Details

---

[2]Technically, it is impossible to apply attention mask in bigram model. Our solution is: as an example, if a verb is encountered, we replace the current likelihood value with a random value between the minimum and maximum, both of which are computed in the same sentence where the verb is located. This operation metaphorically mimics the attention masking.

reported in Table 8 in Appendix).

This finding is consistent with the main finding about probability curvature in DetectGPT (Mitchell et al., 2023): human text is less likely to reach the local maxima of likelihood than model-generated text. In our case, the results suggest that there is more randomness in real human text, which makes it more stable against perturbation in time domain, such as the POS masking.

6 Conclusions

In this study, we propose a new text detection method FourierGPT, which draws information from the spectrum view of relative likelihood scores in language, as the basis for distinguishing human and model texts. Our approach reaches better or competitive performances with state-of-the-art methods on typical zero-shot detection tasks, and particularly better on short text detection tasks.

Our method has the following strengths: First, it utilizes the **relative likelihood** (z-scores) rather than absolute values as used by most previous methods, which means it can capture likelihood patterns in language that are less dependent on the expressiveness of the generation model. We consider this as an advantage because the LM's capability of producing more "likely" texts inevitably grows, and thus, detection methods relying on the absolute "thresholds" of likelihood will also eventually fail.

Secondly, we take a novel **spectrum view** of likelihood, which goes beyond the static view that simply aggregates likelihood at multiple time steps into a single value, but instead, characterizes the *dynamic* features how likelihood changes in time. This spectrum view draws inspiration from cognitive characteristics of language production revealed in the psycholinguistics literature, such as UID, periodicity of surprisal etc. The likehood spectrum can reflect subtle differences in human and model languages that are otherwise indetectable.

Thirdly, our method places a relatively low requirement on how accurate the likelihood scores need be estimated. A GPT-2 level model or even n-gram model suffices to provide likelihood features to reach a decent detection performance. It suggests that how the likelihood of human language distributes in time is a subtle process, which may not be easily mimicked by language models trained via maximum likelihood estimation. LLMs that try hard to squeeze out the gap between every single prediction and ground-truth token may still lack the ability to produce human-like language.

For future work, we will address the limitations with focus on: building stronger supervised classifiers by better utilizing the circularized spectrum; collecting larger datasets from broader domains and multiple languages; looking for more concrete linguistic cases (such as the "*Yes*"/"*No*" example in Section 5.1) to provide richer interpretations for the spectrum-view of likelihood.

7 Limitations

The limitations of the current study are: First, the pair-wise classifier requires that the two texts being classified must be generated from the same prompt. It is yet to be verified whether the classifier's performance will retain if the source prompts are different. Second, the supervised classifier is not a strict zero-shot detector, and its performance still has space for improvement. We argue that it would be less of an issue if the classifier learns the general features instead of idiosyncrasy in certain data. This, however, requires further investigation to whether likelihood spectrum is such a general feature. Third, the datasets examined are relatively small. It is worth exploration on larger datasets (especially short text corpus, such as QA) to further confirm the effectiveness of the method.

References

Kushal Arora, Timothy J O'Donnell, Doina Precup, Jason Weston, and Jackie CK Cheung. 2023. The stable entropy hypothesis and entropy-aware decoding: An analysis and algorithm for robust natural language generation. *arXiv preprint arXiv:2302.06784*.

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*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens

Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aditya Deshpande, Jyoti Aneja, Liwei Wang, Alexander G Schwing, and David Forsyth. 2019. Fast, diverse and accurate image captioning guided by part-of-speech. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10695–10704, California, USA.

Douglas F Elliott. 2013. *Handbook of Digital Signal Processing: Engineering Applications*. Elsevier.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

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.

Dmitriy Genzel and Eugene Charniak. 2002. Entropy rate constancy in text. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 199–206, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Dmitriy Genzel and Eugene Charniak. 2003. Variation of entropy and parse trees of sentences as a function of the sentence number. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 65–72.

Edward Gibson. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Edward Gibson. 2000. The dependency locality theory: A distance-based theory of linguistic complexity. *Image, language, brain/MIT Press*, pages 95–126.

John Hale. 2001. A probabilistic earley parser as a psycholinguistic model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, NAACL '01, page 1–8, USA. Association for Computational Linguistics.

Kenneth Heafield. 2011. Kenlm: faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, page 187–197, USA. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*, Online.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

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

T Florian Jaeger. 2010. Redundancy and reduction: Speakers manage syntactic information density. *Cognitive psychology*, 61(1):23–62.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Richard L Lewis, Shravan Vasishth, and Julie A Van Dyke. 2006. Computational principles of working memory in sentence comprehension. *Trends in cognitive sciences*, 10(10):447–454.

Clara Meister, Ryan Cotterell, and Tim Vieira. 2020. If beam search is the answer, what was the question? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185, Online. Association for Computational Linguistics.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121.

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

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems*, volume 34, pages 4816–4828. Curran Associates, Inc.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Allison Reno, Andrew W Hunter, Yang Li, Tong Ye, and Ann C Foley. 2018. Quantification of cardiomyocyte beating frequency using fourier transform analysis. In *Photonics*, volume 5, page 39. MDPI.

Nathaniel J Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Jinyan Su, Terry Zhuo, Di Wang, and Preslav Nakov. 2023. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12395–12412, Singapore. Association for Computational Linguistics.

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

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. 2024. GPT-who: An information density-based machine-generated text detector. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 103–115, Mexico City, Mexico. Association for Computational Linguistics.

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.

Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. 2019. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea.

Yang Xu and David Reitter. 2017. Spectral analysis of information density in dialogue predicts collaborative task performance. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 623–633, Vancouver, Canada. Association for Computational Linguistics.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. 2024. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. In *The Twelfth International Conference on Learning Representations*.

Zhixian Yang, Pengxuan Xu, and Xiaojun Wan. 2022. Diversifying neural text generation with part-of-speech guided softmax and sampling. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6547–6563, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Zuhao Yang, Yingfang Yuan, Yang Xu, SHUO ZHAN, Huajun Bai, and Kefan Chen. 2023. Face: Evaluating natural language generation with fourier analysis of cross-entropy. In *Advances in Neural Information Processing Systems*, volume 36, pages 17038–17056. Curran Associates, Inc.

G Zipf. 1949. *Human Behavior and the Principle of Least Effort*. New York: Addison-Wesley.

A Hyperparameters for Supervised Classifiers

Detail classification results are shown on Appendix A and Appendix A. The result is the mean of 5-fold cross-validation score of each dataset.

For classification, the data will pass through a scaler, and then a k-best feature selector, at last, the classifier. We apply grid-search on different parameters and report the best outputs. The parameters of the overall workflow are shown below:

- Scaler: MinMax, ZScore, Robust

- KBestFeatures: 50, 80, 100, 120, 150, 200, 250, 300, 400, 500

- SVM (Support Vector Machine):
    - kernel: rbf, linear
    - C: 1, 2, 10
    - gamma: scale, auto

- HGBT (Histogram Gradient Boosting Trees):
    - max iter: 500
    - learning rate: 0.1, 0.05, 0.01, 0.005, 0.001
    - min samples leaf: 7, 13

- MLP (Multi-Layer Perceptrons):
    - constant learning rate: 0.001
    - SGD momentum: 0.9
    - max iter: 800
    - hidden layer: (500), (500, 50)

- LR (Logistic Regression):
    - solver: liblinear
    - penalty: l1, l2
    - C: 1, 2, 10

- KNN (K-Neighbors Classifier):
    - n: 3, 5, 7, 9

- NB (Complement Naive Bayes):
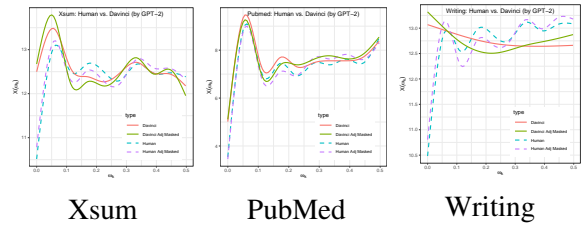   - alpha: 0.5, 1, 2

Xsum   PubMed   Writing

Figure 6: Likelihood spectrum before and after attention mask on ADJ with **GPT-2** estimator.
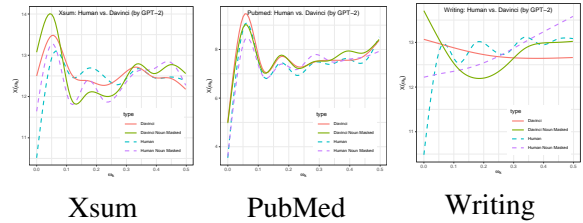

Xsum   PubMed   Writing

Figure 7: Likelihood spectrum before and after attention mask on NOUN with **GPT-2** estimator.

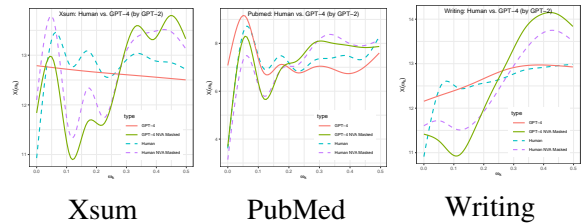Xsum   PubMed   Writing

Figure 8: Likelihood spectrum before and after attention mask on NVA with **GPT-2** estimator.
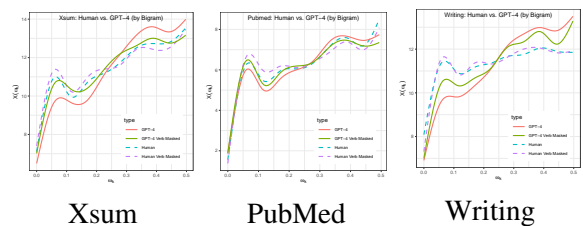

Xsum   PubMed   Writing

Figure 9: Likelihood spectrum before and after attention mask on VERB with **bigram** estimator.

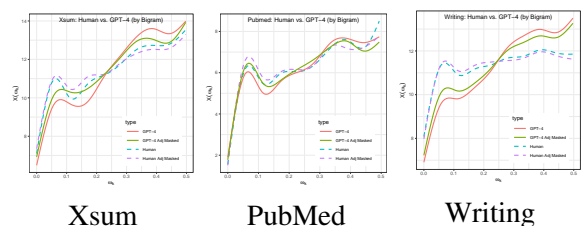

Xsum   PubMed   Writing

Figure 10: Likelihood spectrum before and after attention mask on ADJ with **bigram** estimator.

Dataset	Gen. model	HGBT	KNN	MLP	SVM	NB	LR
PubMed	GPT4	0.567	0.580	0.580	0.593	0.573	0.597
	GPT3.5	0.597	0.583	0.597	0.607	0.607	0.603
	GPT3	0.577	0.600	0.613	0.603	0.597	0.593
Writing	GPT4	0.663	0.677	0.707	0.717	0.710	0.677
	GPT3.5	0.680	0.713	0.693	0.743	0.733	0.737
	GPT3	0.553	0.543	0.537	0.530	0.530	0.527
Xsum	GPT4	0.693	0.670	0.713	0.717	0.707	0.697
	GPT3.5	0.640	0.623	0.660	0.677	0.660	0.667
	GPT3	0.557	0.560	0.550	0.557	0.550	0.563

Table 6: Accuracy of supervised classifier using likelihood spectrum estimated by bigram language model.

Dataset	Gen. model	HGBT	KNN	MLP	SVM	NB	LR
PubMed	GPT4	0.797	0.800	0.800	0.827	0.806	0.810
	GPT3.5	0.580	0.583	0.557	0.600	0.533	0.573
	GPT3	0.580	0.553	0.553	0.570	0.567	0.557
Writing	GPT4	0.713	0.690	0.693	0.707	0.717	0.663
	GPT3.5	0.687	0.683	0.713	0.737	0.750	0.723
	GPT3	0.797	0.800	0.817	0.827	0.807	0.810
Xsum	GPT4	0.717	0.690	0.737	0.740	0.733	0.723
	GPT3.5	0.730	0.710	0.727	0.750	0.737	0.730
	GPT3	0.717	0.687	0.723	0.740	0.733	0.723

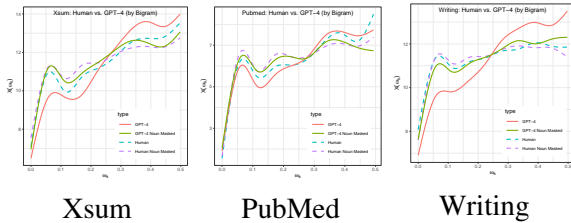Table 7: Accuracy of supervised classifier using likelihood spectrum estimated by GPT2-xl.

Xsum   PubMed   Writing

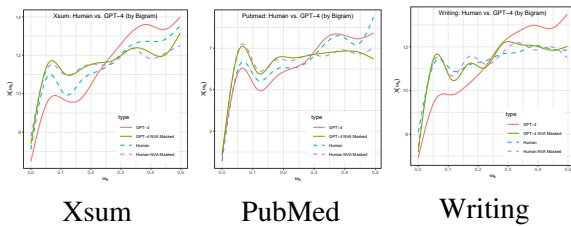Figure 11: Likelihood spectrum before and after attention mask on NOUN with **bigram** estimator.

Xsum   PubMed   Writing

Figure 12: Likelihood spectrum before and after attention mask on NVA with **bigram** estimator.

		Pubmed			Writing			Xsum		
		GPT-3	GPT-3.5	GPT-4	GPT-3	GPT-3.5	GPT-4	GPT-3	GPT-3.5	GPT-4
VERB	H	0.8559	0.8574	0.8536	0.8155	0.8185	0.8187	0.8062	0.8015	**0.8017**
	M	0.8497	0.8390	0.8311	0.8001	0.7950	0.8074	0.8036	0.7947	**0.8029**
NOUN	H	0.7564	0.7598	0.7564	0.7972	0.7982	0.7979	0.7736	0.7717	0.7714
	M	0.7528	0.7280	0.7267	0.7827	0.7624	0.7676	0.7639	0.7537	0.7600
ADJ	H	**0.8109**	0.8110	0.8077	0.8700	0.8679	0.8674	0.8575	0.8526	0.8548
	M	**0.8196**	0.7830	0.7833	0.8485	0.8349	0.8357	0.8486	0.8272	0.8302
NVA	H	0.6976	0.6961	**0.6953**	0.7397	0.7433	0.7414	0.7250	0.7238	0.7232
	M	0.6938	0.6634	**0.6697**	0.7249	0.7151	0.7207	0.7220	0.7083	0.7165

Table 8: Attention mask effect on Spectral Overlap. H denotes human. M denotes model.Bolded Number is the minority which shows that likelihood spectrum from human text changes more than the equivalent from model generated text after attention mask. All the other number shows the opposite situation.