ExaGPT: Example-Based Machine-Generated Text Detection for Human Interpretability

Ryuto Koike[1] **Masahiro Kaneko**[2,1] **Ayana Niwa**[2] **Preslav Nakov**[2] **Naoaki Okazaki**[1,3,4]
[1]Institute of Science Tokyo [2]MBZUAI [3]AIST [4]NII LLMC
{ryuto.koike@nlp., okazaki@}comp.isct.ac.jp
{masahiro.kaneko, ayana.niwa, preslav.nakov}@mbzuai.ac.ae

Abstract

Detecting texts generated by Large Language Models (LLMs) could cause grave mistakes due to incorrect decisions, such as undermining student's academic dignity. LLM text detection thus needs to ensure the interpretability of the decision, which can help users judge how reliably correct its prediction is. When humans verify whether a text is human-written or LLM-generated, they intuitively investigate which of them it shares more similar spans with. However, existing interpretable detectors are not aligned with the human decision-making process and fail to offer evidence that users easily understand. To bridge this gap, we introduce **ExaGPT**, an interpretable detection approach grounded in the human decision-making process for verifying the origin of a text. ExaGPT identifies a text by checking whether it shares more similar spans with human-written vs. with LLM-generated texts from a datastore. This approach can provide similar span examples that contribute to the decision for each span in the text as evidence. Our human evaluation demonstrates that providing similar span examples contributes more effectively to judging the correctness of the decision than existing interpretable methods. Moreover, extensive experiments in four domains and three generators show that ExaGPT massively outperforms prior interpretable detectors by up to +37.0 points of accuracy at a false positive rate of 1%.

1 Introduction

LLMs can yield human-like texts in response to various textual instructions (OpenAI, 2023b; Touvron et al., 2023). Ironically, the powerful generative capability has resulted in various misuses of LLMs, such as cheating on student homework assignments and mass-producing fake news (Tang et al., 2023; Wu et al., 2023). Such abuse of LLMs has sparked the demand for discerning LLM-generated texts from human-written ones.

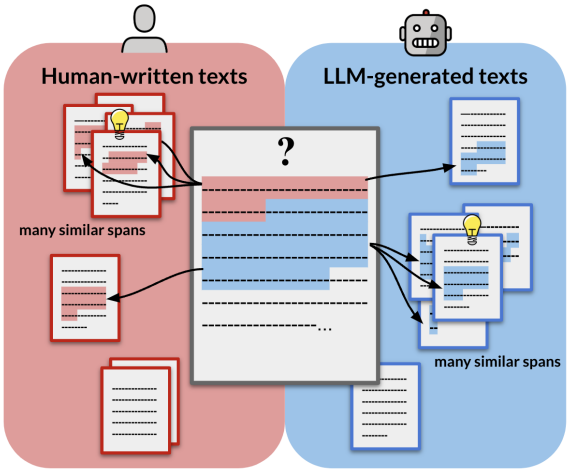

Figure 1: Identifying the author of a text (human vs. LLM) by examining if it shares more similar spans, including verbatim overlaps and semantically similar spans, with human-written vs. LLM-generated texts.

While recent powerful detectors (Mitchell et al., 2023; Su et al., 2023; Koike et al., 2024; Hans et al., 2024; Verma et al., 2024) can help prevent potential misuse of LLMs, misclassifications could lead to severe consequences. For instance, web content writers have faced the career risk due to false-positive classification (Gizmodo, 2024). In school education, incorrect detection might ruin students' academic dignity (OpenAI, 2023a; Bloomberg, 2024). It is extremely difficult to develop a perfect detector with 100% accuracy in such real-world scenarios. There remain edge cases where human-written texts can be misidentified as LLM-generated and vice versa. Thus, it is crucial to develop detectors that provide interpretable evidence, enabling users to assess how reliably correct the predictions are and identify potential misclassifications (Tang et al., 2023; Ji et al., 2024).

Most detectors lack interpretability of their decisions, outputting only binary prediction labels. There are few studies on the interpretability of de-

Findings of the Association for Computational Linguistics: ACL 2026, pages 7702–7715

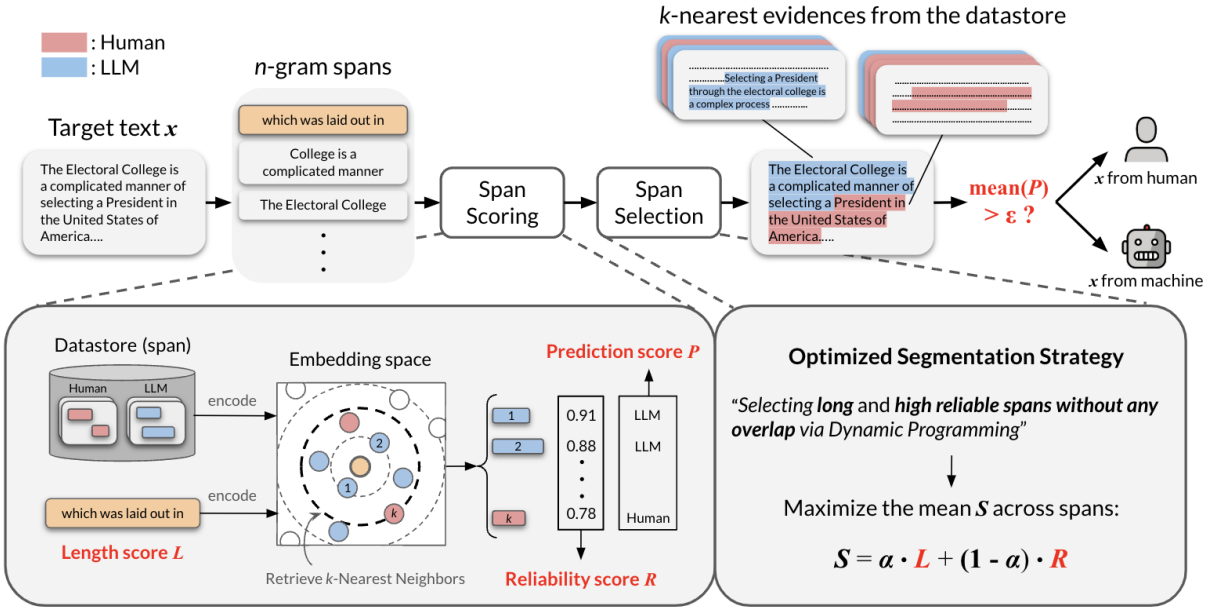

Figure 2: Overview of ExaGPT. It detects the author of a text by examining whether the text shares more similar spans with human-written texts vs. with LLM-generated texts from a datastore.

tection. Gehrmann et al. (2019) color-highlighted the tokens with high probability under the predicted distribution of LMs. Mitrović et al. (2023); Wang et al. (2024) showed which part of a text contributed to a decision based on prediction shifts via perturbations to the text. Yang et al. (2024) provided the n-gram overlaps between the original text and re-prompted ones by LLMs. Here, humans intuitively judge whether a text is human-written or LLM-generated by assessing with which source it shares more *similar spans*, including verbatim overlaps and semantically similar spans (Maurer et al., 2006; Barrón-Cedeño et al., 2013). However, current detectors are not aligned with the human decision-making process (Figure 1) and fail to yield sufficiently interpretable evidence for users.

Motivated by this gap, we present **ExaGPT**, an interpretable detection method based on the human decision-making process of verifying the origin of a text. In particular, ExaGPT makes a prediction by examining whether the text shares more similar spans with human-written vs. with LLM-generated texts from a datastore. This approach can provide similar span examples that contribute to the decision for each span in the text as interpretable evidence. To present interpretable span-segmented text as a final result, we apply a dynamic programming algorithm and determine the optimal span break. It balances the long span length and its high frequency with the datastore (i.e., many similar phrases to the span exist in the datastore). The similarity of the retrieved spans to each span in the target text can help users judge the reliability of the detection result.

To evaluate the interpretability of detection, we conducted a human evaluation of how well people can infer the correctness of the detection from the detector's evidence. We found that providing similar span examples contributes more effectively to judging the correctness of the detection than existing interpretable methods. Moreover, extensive experiments in four domains and three generators showed that ExaGPT massively outperforms prior interpretable and powerful detectors by up to +37.0 points accuracy, even at a constant false positive rate of 1%. From these results, we observe that ExaGPT achieves high interpretability in its detection result and also high detection performance.[1]

2 Methodology

ExaGPT classifies a text based on whether it shares more similar spans with human-written or with LLM-generated texts from a datastore. As a final result, ExaGPT offers the span-segmented text where each span is accompanied by similar span examples that contribute to the decision. Figure 2 illustrates the workflow of ExaGPT, which has two phases: **Span Scoring** and **Span Selection**. In

[1]Codes and datasets are available at https://github.com/ryuryukke/ExaGPT.

the first phase, we mainly investigate whether each span in the target text shares more similar spans with human-written or LLM-generated texts from a datastore. Meanwhile, we calculate scores for each span, which we use in the second phase (§2.1). In the second phase, we primarily decide the optimal span segmentation to aid users' understanding of the final result. Specifically, we apply a dynamic programming (DP) algorithm with the scores from the first phase to find the span boundaries, balancing span length and its frequency within the datastore (§2.2). Finally, we detect the target text based on the selected spans and we provide similar span examples for each target span as evidence (§2.3). We will go into further details below.

2.1 Span Scoring via *k*-NN Search

Given a target text x to be classified, we define an n-gram span in the text x as $x_{i:i+n}$, which is any continuous sequence of n tokens starting in the i-th token. For each n-gram target span $x_{i:i+n}$, we retrieve the top-k most similar[2] n-gram spans s_j ($j \in \{1, \ldots, k\}$) from the datastore, with each original label and similarity $\{(s_j, l_j, c_j)\}_{j=1}^{k}$. Here, l_j is `Human` when the span s_j is part of a human-written text, or `LLM` when the span s_j is a part of a LLM-generated text. c_j is the similarity between the target span $x_{i:i+n}$ and each retrieved span s_j.

Consequently, we calculate the following metrics for each target span $x_{i:i+n}$: *length score* L, *reliability score* R, and *prediction score* P. The length score L is the number of tokens in the target span:

$$L(x_{i:i+n}) = n \tag{1}$$

The reliability score R is the mean similarity c_j between the target span and each retrieved span:

$$R(x_{i:i+n}) = \frac{\sum_{j=1}^{k} c_j}{k} \tag{2}$$

The reliability score R indicates how many similar spans exist in the datastore for the target span. The prediction score P is a ratio of `LLM` label in the original labels l_j of the retrieved spans:

$$P(x_{i:i+n}) = \frac{\sum_{j=1}^{k} \mathbb{1}(l_j = \texttt{LLM})}{k}. \tag{3}$$

The prediction score P indicates whether the target span shares more similar spans with human-written vs. with LLM-generated texts in the datastore.

[2]We encode the target span, and all spans in the datastore into the same embedding space. We then perform k-nearest neighbor (k-NN) search based on the cosine similarity of each two span embeddings. See more details in §3.1.

Algorithm 1 Span Segmentation Optimization

Input: Target text x; Length of target text m; Length score L; Reliability score R; Maximum length of n-gram span N; Hyper-parameter α
Output: List of selected n-grams T

```
dp[0, ..., m − 1] ← [([0], None)] ∗ m
for i = 1 to m do
  for j = min(i − N, 0) to i do
    l, r ← L^std(x_{j:i}), R^std(x_{j:i})
    scores ← dp[j][0] + [αl + (1 − α)r]
    s_cand ← average(scores)
    if average(dp[i][0]) < s_cand then
      dp[i] ← (scores, j)
    end if
  end for
end for
Traverse dp backward and collect span breaks
return List of selected n-grams T
```

2.2 Span Selection via Dynamic Programming

In this phase, we select spans $T = [t_1, \ldots, t_H]$ in the target text x, so that the text is segmented without overlaps as a final result:

$$\begin{aligned} &x = t_1 \oplus t_2 \oplus \cdots \oplus t_H, \\ &t_i \cap t_j = \varnothing \quad (i, j \in \{1, \ldots, H\}, i \neq j) \end{aligned} \tag{4}$$

To facilitate users' understanding of the final result, we optimize the span segmentation that includes longer and more similar spans with ones from the datastore. Algorithm 1 describes our dynamic programming strategy to find the best span break. Formally, we select spans T to maximize the score S across the spans in the target text:

$$S(T) = \frac{\sum_{h=1}^{H} \{\alpha L^{\text{std}}(t_h) + (1-\alpha) R^{\text{std}}(t_h)\}}{H}. \tag{5}$$

Here, $L^{\text{std}}(t_h)$ and $R^{\text{std}}(t_h)$ are the normalized[3] versions of the length score L and the reliability score R of the span t_h. α is an interpolation coefficient ranging from 0.0 to 1.0. α determines the relative contribution of the length score and the reliability score to the span segmentation.

2.3 Overall Detection with Evidence

Given a sequence of the selected spans T each with a prediction score for the target text x, ExaGPT

[3]To align the scales of the length score and the reliability score, each score is normalized using the mean and the variance in the validation split of our dataset.

identifies a text based on the mean prediction score:

$$P_{\text{overall}} = \frac{\sum_{h=1}^{H} P(t_h)}{H}. \quad (6)$$

ExaGPT classifies a text as `LLM` if P_{overall} exceeds a detection threshold ϵ, and otherwise as `Human`. As evidence of the decision, ExaGPT provides retrieved top-k similar spans for each span in the text:

$$E = [(t_h, [s_h^1, \ldots, s_h^k])]_{h=1}^{H}. \quad (7)$$

The similarity of the retrieved spans to each span in the target text can help users judge how reliably correct the detection result is.

3 Experiments and Results

3.1 Overall Setup

Metrics. To assess detection performance, we use the AUROC score, which is widely used in detection studies. However, it is only useful to observe the overall behavior of a detector through all possible thresholds. In practice, it is critical to minimize the false positive classification, i.e., wrongly identifying human-written texts as LLM-generated. We thus report detection accuracy with a threshold by fixing the false-positive rate (FPR) at 1%, a common evaluation metric in recent robustness studies (Krishna et al., 2023; Hans et al., 2024; Dugan et al., 2024).

Datasets. We use the M4 dataset (Wang et al., 2024), a large-scale detection benchmark comprising pairs of human-written and LLM-generated texts across multiple languages, domains, and generators. Our experiments use the English subset, including 3,000 pairs of human-written and LLM-generated texts from each combination of four domains: Wikipedia, Reddit, WikiHow, and arXiv, and three generators: ChatGPT, GPT-4 as closed-source LLMs, and Dolly-v2 (Conover et al., 2023) as open-source LLMs. For each combination, we split the dataset into three parts: train/valid/test with 2,000/500/500 pairs, respectively.

Baselines. We compare ExaGPT to three strong and interpretable detectors (as detailed in §5): RoBERTa with SHAP (Mitrović et al., 2023), LR-GLTR (Wang et al., 2024), and DNA-GPT (Yang et al., 2024). The first one is a supervised classifier based on RoBERTa (Liu et al., 2019), which we fine-tune for detection on our train split. Similarly, we train the LR-GLTR detector on our train split with selected and hand-crafted GLTR features (Gehrmann et al., 2019), following Wang et al. (2024). The hyperparameter settings for training both RoBERTa and LR-GLTR are aligned with those of Wang et al. (2024). Further configurations of the baselines are in Appendix A.

ExaGPT. In the span scoring phase, ExaGPT uses our train split as the datastore for each combination of domains and generators. We consider the n-gram size to be from 1 to 20 across the entire dataset. We embed the target span and all spans in the datastore into the same vector space using BERT[4], a standard embedding model. For a span embedding, we feed a text into BERT and take the mean hidden states[5] of the tokens within the span. We retrieve the top-k (=10)[6] most similar spans from the datastore for each target span via k-NN search using the FAISS (Johnson et al., 2017).

In the span selection phase, we select the optimal α from values between 0.0 and 1.0 at 0.125 intervals, where ExaGPT exhibits the best detection performance in our validation split. The α is constant through our evaluation of the interpretability and the detection performance of ExaGPT.

Human Evaluation on Interpretability. We assess the interpretability of detectors through human evaluation, as it is crucial that detectors provide evidence enabling users to judge the reliability of detection results. Thus, we first design an evaluation to test whether the provided evidence *actually helps* users determine whether a detection is correct, a practical aspect overlooked in prior work. Participants are shown the detection evidence and asked to judge correctness. Accordingly, our interpretability metric is defined as the accuracy of these human judgments.

For each detector, we evaluate 96 samples[7] from our test split across all combinations of domains and generators, with an equal ratio of correct and

[4] https://huggingface.co/google-bert/bert-large-uncased

[5] We select the second layer where the k-NN spans are similar to the target span well-balanced lexically and semantically, enhancing its interpretability in our pilot study.

[6] We choose the value of k so that ExaGPT shows favorable detection performance over smaller values in our pilot study and does not reduce the interpretability. Since ExaGPT presents retrieved spans as evidence, keeping k small helps users assess detection correctness based on a manageable amount of information.

[7] The 96 samples for each detector consist of two samples (one correct and one incorrect) across four domains and three generators, distributed among four participants.

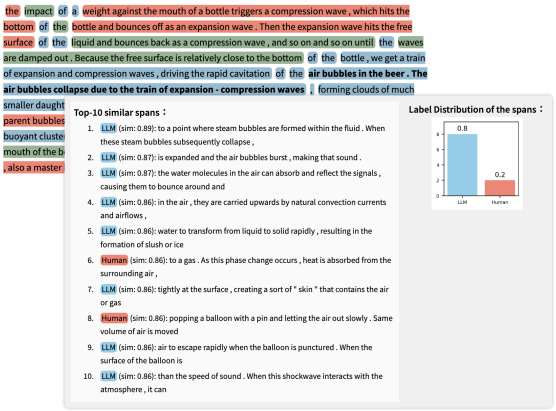

Figure 3: User interface of ExaGPT. Hovering over a text span displays the tooltip about the retrieved similar spans each with the similarity to the span and the original label distribution.

incorrect detections. Four annotators (one MSc student, one PhD student, and two NLP researchers)[8] were provided with different samples. Figure 3 shows the user interface of ExaGPT in the evaluation. Spans are highlighted[9] in red, green, and blue for which prediction score P is lower than 0.5 (human-written), equal to 0.5 (neither), and higher than 0.5 (LLM-generated). Participants judge the correctness of the detection by mainly examining similar span examples. Details of baseline evidence are provided in Appendix B.

3.2 Results

Detection Interpretability. Table 1 presents the differences in the accuracy of human judgments on detection correctness based on the provided evidence across baseline detectors and ExaGPT. The accuracy of human judgments on ExaGPT is relatively higher compared to baseline detectors by up to +13.6 points. This indicates that ExaGPT provides more interpretable evidence than other baselines, helping humans judge the correctness of detections more effectively.

Specifically, DNA-GPT also provides n-gram span overlaps between the target text and regenerated LLM texts from the truncated part as evidence. The comparison of the human evaluation score between DNA-GPT and ExaGPT suggests

[8]They are not authors of this paper. Although all annotators have NLP backgrounds, they vary in research experience and language proficiency (including both native and non-native English speakers), providing diverse perspectives.

[9]ExaGPT performs the overall detection rather than detecting each span individually. However, for better readability, each span is color-highlighted on its prediction score.

Detector	ACC. of Human Judgements (%) ↑
RoBERTa	47.9
LR-GLTR	57.3
DNA-GPT	53.1
ExaGPT	**61.5**

Table 1: Comparison of the accuracy (ACC.) of human judgments on the correctness of detections based on evidence across baseline detectors and ExaGPT. Higher accuracy implies that the detector provides more interpretable evidence to users.

that providing not only simple overlaps but also semantically similar spans contributes to better interpretability. We further investigate how the similarity between the target span and retrieved spans correlates with the correctness of the detection of ExaGPT in §4.

Detection Performance. Table 2 presents the differences in detection performance between baseline detectors and ExaGPT across domains and generators. The detection performance includes AUROC and the accuracy at 1% FPR. Overall, ExaGPT consistently demonstrates detection performance on par with or better than baselines, including supervised classifiers. Specifically, on accuracy at 1% FPR, ExaGPT achieves the best average detection performance on all three generators, outperforming baselines by a large margin of up to +37.0 points. This suggests that ExaGPT is the most effective detector in practical scenarios, where we need to minimize the false positives.

Summary. Experiments demonstrate that ExaGPT achieves both superior interpretability and stronger detection performance compared to prior interpretable detectors.

4 Analysis

What Makes ExaGPT Interpretable. Our human evaluations demonstrate that ExaGPT provides highly interpretable evidence compared to prior detectors. To explore the reason for this, we investigate the difference in the characteristics of the selected spans as a final output between correct and incorrect predictions by ExaGPT. We focus on span length and mean similarity between each target span and the retrieved spans (reliability score R), which are prioritized in the span selection. We randomly select 1,000 correct and 1,000 incorrect ExaGPT predictions on our test splits across all combinations of domains and generators.

Generator	Detector	Wikipedia		Reddit		WikiHow		arXiv		Average	
		AUROC	ACC.	AUROC	ACC.	AUROC	ACC.	AUROC	ACC.	AUROC	ACC.
ChatGPT	RoBERTa	**100.0**	77.1	**99.8**	61.0	**100.0**	50.0	**100.0**	87.3	**100.0**	68.9
	LR-GLTR	95.0	60.0	99.4	**94.0**	97.5	85.8	99.8	**97.7**	97.9	84.4
	DNA-GPT	84.8	49.4	92.3	62.9	99.4	93.5	89.0	59.9	91.4	66.4
	ExaGPT	98.6	**92.3**	98.9	86.6	99.5	**96.0**	99.6	95.8	99.2	**92.7**
GPT-4	RoBERTa	**100.0**	**87.8**	**100.0**	66.4	**100.0**	77.4	**100.0**	68.6	**100.0**	75.1
	LR-GLTR	97.8	85.7	99.6	**97.2**	94.8	77.8	**100.0**	98.5	98.1	89.8
	DNA-GPT	40.3	48.1	71.9	68.6	44.6	49.9	72.2	54.4	57.3	55.3
	ExaGPT	98.3	87.3	99.3	91.1	98.8	**92.2**	99.7	**98.7**	99.0	**92.3**
Dolly-v2	RoBERTa	**100.0**	61.8	**100.0**	50.0	**100.0**	70.8	**100.0**	**82.8**	**100.0**	66.4
	LR-GLTR	79.7	57.7	95.3	**79.0**	72.4	55.0	93.7	78.2	85.3	67.5
	DNA-GPT	68.0	61.5	67.5	66.1	87.7	**82.3**	64.9	57.7	72.0	66.9
	ExaGPT	85.8	**63.8**	96.2	76.6	94.3	75.6	85.2	67.3	90.4	**70.8**

Table 2: Comparison of detection performances of ExaGPT and baseline detectors on texts from various domains and generators. *ACC.* indicates the detection accuracy at 1% FPR. **Bold** and Underline indicate the best and runner-up performance for each combination of domains and generators.

Target Span	LLM	*published in 1993. The novel tells the story of a young Jewish slave, Hadassah,*
***k*-NN Spans**	LLM (0.92)	*and was first published in 1936. The book tells the story of three orphaned sisters,*
	LLM (0.92)	*published in 2012. The novel revolves around the story of a young woman*
	LLM (0.90)	*and published in 2010. The novel tells the story of Michael Beard, a*
	LLM (0.90)	*ling of the biblical book, Song of Solomon, and is considered one of the*
	LLM (0.90)	*man and published in 1963. The book was later adapted into a Disney film of the*
	LLM (0.90)	*. The film tells the story of a young*
	Human (0.89)	*the Xanth series. It is the second book of a trilogy beginning with Vale of the*
	LLM (0.89)	*published in 1959. The novel is set in the Arctic region and follows the story of Dr.*
	Human (0.89)	*. It is the third novel in the Dahak trilogy, after the de*
	LLM (0.89)	*for his semi-autobiographical novel, "The Watch that Ends the Night". Born in*

Table 3: Examples of k-NN spans for a target span retrieved by ExaGPT. The colored part represents the original label for each span (LLM in blue and Human in red, respectively). In the part of k-NN spans, the similarity between the target span and each k-NN span is added.

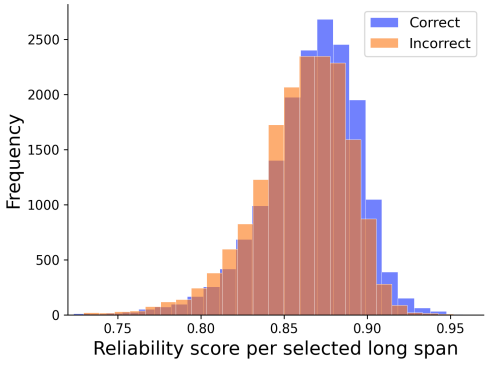

Figure 4: Reliability score distributions of long spans ($n \geq 10$) in correct and incorrect samples of ExaGPT.

Figure 4 presents the reliability score distributions for long spans ($n \geq 10$) in correct and incorrect samples. A rightward shift indicates that correct samples of ExaGPT include more long spans with higher reliability scores than incorrect ones. This suggests that offering long spans with high reliability scores helps users judge the correctness of the detections. Table 3 presents examples of long spans ($n = 19$) with high reliability scores for a target span retrieved by ExaGPT. We can see that the retrieved spans are well-balanced between lexical and semantic similarity to the target span.

Impact of α. In our experiments, we determine the optimal coefficient α for ExaGPT (as used in Eq. 5) based on the best detection performance on the validation split. To examine the robustness of ExaGPT to the choice of α, we analyze how detection performance varies as α changes.

Figure 5 depicts the relationship between α and the detection performance of ExaGPT, evaluated on ChatGPT-generated text across four domains. α ranges from 0.0 to 1.0 in increments of 0.125. We observe that larger values of α generally lead to lower detection performance. This suggests that placing greater weight on the reliability score (i.e., selecting target spans that are more similar to spans

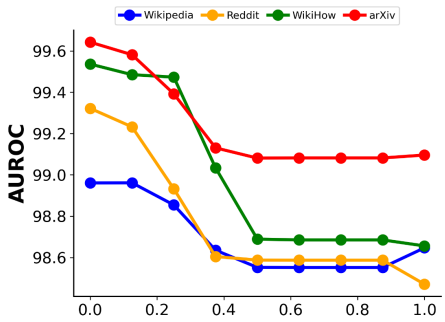

Figure 5: Impact of α on the detection performance of ExaGPT, using ChatGPT as a generator.

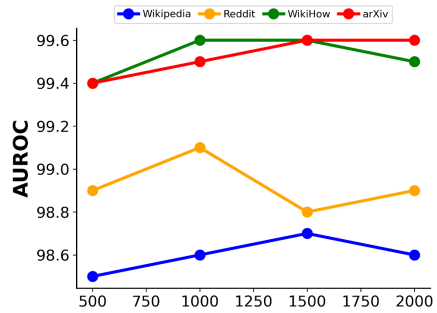

Figure 6: Impact of the datastore size on the detection performance of ExaGPT, using ChatGPT as a generator.

		Test				
		Wikipedia	**Reddit**	**WikiHow**	**arXiv**	**Average**
Train	Wikipedia	**98.3 / 87.3**	91.7 / 68.2	54.1 / 53.3	89.3 / 60.5	83.4 / 67.3
	Reddit	90.1 / 60.7	**99.3 / 91.1**	74.6 / 50.6	93.0 / 63.9	89.3 / 66.6
	WikiHow	66.2 / 50.6	76.9 / 60.4	**98.8 / 92.2**	64.7 / 51.6	76.7 / 63.7
	arXiv	73.7 / 50.4	86.3 / 56.2	57.0 / 51.5	**99.7 / 98.7**	79.2 / 64.2
	ALL	94.3 / 80.7	96.7 / 83.5	92.9 / 73.4	99.5 / 96.7	**95.9 / 83.6**

Table 4: Cross-domain detection with GPT-4 as the generator. The scores are AUROC / Acc@FPR=1%.

		Test			
		ChatGPT	**GPT-4**	**Dolly**	**Average**
Train	ChatGPT	**99.6 / 95.8**	98.2 / 84.5	63.2 / 50.3	87.0 / 76.9
	GPT-4	94.6 / 66.6	**99.7 / 98.7**	61.8 / 51.5	85.4 / 72.3
	Dolly	93.0 / 69.9	89.9 / 65.5	**85.2 / 67.3**	89.4 / 67.6
	ALL	98.9 / 91.4	99.3 / 95.6	76.4 / 52.9	**91.5 / 80.0**

Table 5: Cross-generator detection with arXiv as the domain. The scores are AUROC / Acc@FPR=1%.

Detector	**Wikipedia**	**Reddit**	**WikiHow**	**arXiv**	**Average**
LR-GLTR	89.4 / 60.2	97.0 / **76.8**	89.5 / 58.0	**99.6 / 96.7**	93.9 / 72.9
ExaGPT	**98.0 / 86.5**	**97.2** / 69.4	**91.1 / 73.8**	97.7 / 76.4	**96.0 / 76.5**

Table 6: Performance on paraphrased text. The scores are AUROC / Acc@FPR=1%.

in the datastore) improves detection performance. Notably, across all four domains, the lowest AUROC is 98.5%, suggesting that changes in α do not cause a substantial performance drop that would change the ranking of detectors. See Appendix C for consistent trends in all generators.

Impact of Datastore Size. In our evaluation, ExaGPT uses the train split as the datastore from which it retrieves the top-k most similar spans for each span in a target text. To study the robustness of ExaGPT to datastore size, we analyze how detection performance varies as the datastore size changes. The train split contains 2,000 pairs of human-written and LLM-generated texts. We randomly sample {500, 1,000, 1,500, 2,000} pairs from the train split as datastores of different sizes.

Figure 6 shows the relationship between datastore size and the detection performance of ExaGPT across four domains using ChatGPT as the generator. Overall, ExaGPT remains robust to datastore size, exhibiting only minor performance degradation. Interestingly, ExaGPT with a datastore of 500 pairs performs comparably to using the full 2,000 pairs in terms of AUROC. See Appendix C for consistent trends in all generators.

Unknown Domain or Generator. While our primary goal is to improve interpretability, we also perform cross-domain and cross-generator experiments to examine how ExaGPT can be leveraged in more realistic settings. Tables 4 and 5 report the results, where "ALL" denotes a datastore constructed from all domains or generators, with samples drawn uniformly to match the size of the single-source setting.

We observe that when the domain or generator in the datastore is different from the target model or generator, detection performance is reduced. However, the "ALL" setting, which is more realistic in practice, maintains consistently strong performance across domains and generators. The results further suggest which domains or generators are effective for generalization. For instance, including Reddit mitigates AUROC drops across domains, whereas including ChatGPT helps maintain high detection performance against GPT-4.

Paraphrased Text. We also investigate the robustness of ExaGPT against paraphrased text. Following Krishna et al. (2023), we utilize DIPPER, an 11B document-level paraphraser, to rewrite machine-generated text. Table 6 reports results

	#Instance	GPU memory (GB)	Latency (sec.)	AUROC
2000 pair	36M	162.2	14.6	99.5
500 pair	9.1M	54.7 (66%↓)	5.81 (60%↓)	99.4
500 pair + IVFPQ	9.1M	20.2 (87%↓)	1.22 (90%↓)	97.8

Table 7: Inference cost analysis of ExaGPT.

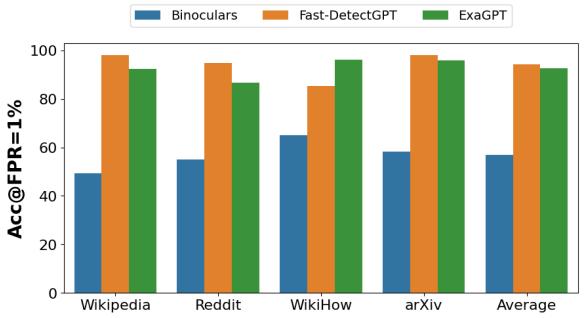

Figure 7: Performance comparison of ExaGPT with state-of-the-art non-interpretable detectors on Acc@FPR=1%, using ChatGPT as a generator.

on all domains using ChatGPT as the generator. Among strong interpretable detectors, LR-GLTR was the runner-up in our original in-domain evaluation (§3.2). Even under paraphrasing, ExaGPT maintains moderately high detection performance and consistently outperforms LR-GLTR.

Inference Cost. We evaluate the inference efficiency of ExaGPT, focusing on the k-NN search, which is the primary bottleneck. We measure latency and GPU memory usage of the k-NN search, and further evaluate FAISS-based approximate search with IVFPQ indexes to reduce resource usage and improve inference speed.

Table 7 shows results on WikiHow with ChatGPT, where #Instance refers to the number of n-gram spans in the datastore. Reducing the datastore from 2,000 to 500 pairs lowers memory usage by 66% and latency by 60% with almost no performance loss. Using FAISS-based approximation further reduces memory by 87% and latency by over 90%, while the performance drop is still moderate. These results demonstrate that ExaGPT can be deployed efficiently under practical computational budgets.

Comparison with state-of-the-art Detectors. To understand the trade-off between interpretability and detection performance, we also compare ExaGPT with state-of-the-art non-interpretable detectors. Figure 7 reports results for Binoculars (Hans et al., 2024) and Fast-DetectGPT (Bao et al., 2024), which are strong metrics-based detectors. The evaluation is conducted on ChatGPT-generated text across all domains. Notably, despite being interpretable, ExaGPT achieves detection performance on par with or even better than these state-of-the-art non-interpretable detectors.

5 Related Work

Machine-Generated Text Detection. Prior studies have presented various types of detection algorithms for LLM-generated text, which can be broadly grouped into three categories: *text watermarking*, *metrics-based*, and *supervised classifiers*. Text watermarking modifies the decoding process so that secretly selected tokens appear more frequently, enabling detection by checking their ratio in the output (Kirchenbauer et al., 2023). Metrics-based methods measure the probabilistic discrepancy of a text with the model's predicted distribution, using signals such as token log probabilities (Gehrmann et al., 2019), token ranks (Solaiman et al., 2019; Su et al., 2023), entropy (Lavergne et al., 2008), perplexity (Beresneva, 2016; Hans et al., 2024), and negative probability curvature (Mitchell et al., 2023; Bao et al., 2024). Supervised classifiers are models specifically fine-tuned to discern human-written and LLM-generated texts with labels. They vary from probabilistic (Ippolito et al., 2020; Crothers et al., 2022) to neural methods (Uchendu et al., 2020; Guo et al., 2023; Emi and Spero, 2024).

Interpretability of Detection Results. To minimize the undesired consequences of detection, there is a need to develop a detector that provides interpretable evidence for its decisions. However, most detectors output only a binary label, and only a few studies aim to provide interpretable evidence. GLTR (Gehrmann et al., 2019) highlights tokens with high model likelihood. Other studies apply explainable ML techniques such as LIME and SHAP to supervised classifiers (Mitrović et al., 2023; Wang et al., 2024; Ribeiro et al., 2016; Lund-

berg and Lee, 2017). DNA-GPT (Yang et al., 2024) compares n-gram overlaps between the target text and LLM-generated continuations, providing actual LLM texts with overlaps as evidence.

Unlike prior interpretable detectors, ExaGPT is grounded in the human decision-making process for verifying the origin of a text (Maurer et al., 2006; Barrón-Cedeño et al., 2013) and can provide more interpretable evidence, as explained in the previous sections.

Example Retrieval for Interpretability. Beyond LLM text detection, presenting retrieved examples has been widely used to improve interpretability across NLP tasks, from text generation (Khandelwal et al., 2021) to sequential text classification (Wiseman and Stratos, 2019; Kaneko et al., 2022). Predictions are typically obtained by interpolating the base model's output distribution with a distribution derived from retrieved nearest-neighbor examples.

Our work has a similar direction of using retrieved similar examples for better interpretability with prior studies in other NLP tasks. In LLM text detection, it is critical to segment the target text into n-gram spans with individually assigned labels (Cheng et al., 2025). ExaGPT therefore retrieves similar examples for each span and optimizes the final segmentation using dynamic programming.

6 Conclusion

We introduced ExaGPT, an interpretable human vs. machine detection approach grounded in how humans verify the origin of a text. ExaGPT classifies a text by examining whether it shares more verbatim and semantically similar spans with human-written vs. with LLM-generated texts from a datastore. As evidence of the detection, ExaGPT provides similar span examples for each span in the text. Human evaluation and further analysis show that providing similar span examples allows users to judge detection correctness more effectively than prior interpretable detectors. Moreover, extensive experiments demonstrate that ExaGPT achieves superior detection performance compared to previous strong detectors, even at a false positive rate of 1%. Overall, ExaGPT achieves both high interpretability and strong detection performance. For future work, we plan to extend the human evaluation to broader user populations and investigate more optimal datastore configurations for better generalization.

7 Limitations

Scope of Human Evaluation. Our human evaluation relied on four annotators with NLP backgrounds, which may limit the generalizability of the results to typical users. The evaluation was designed as a controlled comparison of interpretability between ExaGPT and existing detectors, rather than a general usability study. Since baseline methods require interpreting technical outputs (e.g., probability distributions or contribution scores), we used annotators with comparable expertise to ensure fair evaluation across methods. Nevertheless, whether the interpretability benefits of ExaGPT extend to broader user populations remains unclear, and we leave this to future work.

Datastore Dependence. ExaGPT depends on the datastore for its detection by design. This raises concerns about its generalizability to text domains or generators not covered by the datastore. Experiments show that the performance of ExaGPT is reduced in cross-domain and cross-generators settings. However, in practice, using multiple sources in the datastore, rather than relying on a single source, is considered more realistic. Our results confirm that, in such a setting, ExaGPT consistently maintained a reasonably high performance across multiple domains and generators. We look forward to future studies into optimal datastore configurations for more effective real-world deployment.

8 Ethics and Broader Impact

Human Subject Considerations. In our study, human subjects are engaged in identifying the correctness of the detection based on evidence. All annotators provided informed consent, were fully aware of the study's objectives, and had the right to withdraw at any time.

Responsible Use of Detectors. It is extremely difficult to achieve perfect detection in the real-world. Given the severe consequences of misclassifications, detectors should provide evidence enabling users to assess the reliability of predictions and identify potential errors. ExaGPT improves interpretability by presenting similar spans as evidence. While such evidence can support human judgment, it does not eliminate the inherent uncertainty in detection. Therefore, more generally, detector outputs should not be treated as authoritative proof for sanctions, but rather as advisory signals considered alongside other contextual information.

9 Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 25H01137. This work was supported by JST K Program Japan Grant Number JPMJKP24C3. This work was supported by JST SPRING, Japan Grant Number JPMJSP2106. These research results were obtained from the commissioned research (No.22501) by National Institute of Information and Communications Technology (NICT), Japan. We used ABCI 3.0 provided by AIST and AIST Solutions with support from "ABCI 3.0 Development Acceleration Use".

References

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 *Proceedings of the 12th International Conference on Learning Representations*, ICLR '24, Vienna, Austria. OpenReview.net.

Alberto Barrón-Cedeño, Marta Vila, M. Antònia Martí, and Paolo Rosso. 2013. Plagiarism meets paraphrasing: Insights for the next generation in automatic plagiarism detection. *Computational Linguistics*, 39(4):917–947.

Daria Beresneva. 2016. Computer-generated text detection using machine learning: A systematic review. In *Proceedings of the 21st International Conference on Applications of Natural Language to Data Bases*, NLDB '16, Salford, UK. Springer.

Bloomberg. 2024. AI detectors falsely accuse students of cheating—with big consequences. Accessed on 2024-10-20.

Zihao Cheng, Li Zhou, Feng Jiang, Benyou Wang, and Haizhou Li. 2025. Beyond binary: Towards fine-grained llm-generated text detection via role recognition and involvement measurement. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 2677–2688, Sydney NSW, Australia. ACM.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned LLM. Accessed: 2024-7-12.

Evan Crothers, Nathalie Japkowicz, and Herna Viktor. 2022. Machine generated text: A comprehensive survey of threat models and detection methods. *ArXiv preprint*, abs:2210.07321.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '24, pages 12463–12492, Bangkok, Thailand. Association for Computational Linguistics.

Bradley Emi and Max Spero. 2024. Technical report on the Pangram AI-generated text classifier. *ArXiv preprint*, abs:2402.14873.

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*, ACL '19, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Gizmodo. 2024. AI detectors get it wrong. writers are being fired anyway. Accessed on 2024-07-12.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is ChatGPT to human experts? Comparison corpus, evaluation, and detection. *ArXiv preprint*, arxiv:2301.07597.

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

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*, ACL '20, pages 1808–1822, Online. Association for Computational Linguistics.

Jiazhou Ji, Ruizhe Li, Shujun Li, Jie Guo, Weidong Qiu, Zheng Huang, Chiyu Chen, Xiaoyu Jiang, and Xinru Lu. 2024. Detecting machine-generated texts: Not just "AI vs humans" and explainability is complicated. *ArXiv preprint*, abs:2406.18259.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *ArXiv preprint*, abs:1702.08734.

Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. Interpretability for language learners using example-based grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '22, pages 7176–7187, Dublin, Ireland. Association for Computational Linguistics.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *Proceedings of the 9th International Conference on Learning Representations*, ICLR '21, Virtual Event, Austria. OpenReview.net.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. OUTFOX: LLM-generated essay detection through in-context learning with adversarially generated examples. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, AAAI '24, pages 21258–21266, Vancouver, Canada. AAAI Press.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems 36*, NeurIPS '23, New Orleans, LA, USA.

Thomas Lavergne, Tanguy Urvoy, and François Yvon. 2008. Detecting fake content with relative entropy scoring. In *Proceedings of the ECAI'08 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, CEUR Workshop Proceedings.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv preprint*, arxiv:1907.11692.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, NeurIPS '17, pages 4765–4774, Long Beach, CA, USA.

Hermann Maurer, Frank Kappe, and Bilal Zaka. 2006. Plagiarism – A survey. *Journal of Universal Computer Science*, 12(8):1050–1084.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962. PMLR.

Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. 2023. ChatGPT or human? detect and explain. explaining decisions of machine learning model for detecting short ChatGPT-generated text. *ArXiv preprint*, arxiv:2301.13852.

OpenAI. 2023a. How can educators respond to students presenting AI-generated content as their own? Accessed: 2024-6-10.

OpenAI. 2023b. Introducing ChatGPT. Accessed on 2024-03-10.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, San Francisco, CA, USA. ACM.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *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*, EMNLP '23, pages 12395–12412, Singapore. Association for Computational Linguistics.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting llm-generated texts. *ArXiv preprint*, arxiv:2303.07205.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs:2307.09288.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, EMNLP '20, pages 8384–8395, Online. 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)*, NAACL '24, pages 1702–1717, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4: multi-generator, multi-domain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, EACL '24, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

Sam Wiseman and Karl Stratos. 2019. Label-agnostic sequence labeling by copying nearest neighbors. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL '19, pages 5363–5369, Florence, Italy. Association for Computational Linguistics.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F. Wong, and Lidia S. Chao. 2023. A survey on LLM-generated text detection: Necessity, methods, and future directions. *ArXiv preprint*, arxiv:2310.14724.

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 *Proceedings of the 12th International Conference on Learning Representations*, ICLR '24, Vienna, Austria. OpenReview.net.

A Detailed Configurations of Baselines

LR-GLTR. Following Wang et al. (2024), we leverage the two categories of GLTR features: (1) the number of tokens in the top-{10, 100, 1,000, 1,000+} ranks in the predicted probability distribution of LLMs (four features), and (2) the probability distribution of the word divided by the maximum probability of any word at the same position over 10 bins between 0.0 and 1.0 (ten features).

DNA-GPT. For DNA-GPT, we set the truncation ratio γ to 0.7 and 0.5, and the number of re-generations K to 10 and 5 for closed-source and open-source LLMs. We ensured that the `temperature` is the same as the one used to generate a target text and that the generation prompt is known. These configurations were found to ensure the favorable performance of DNA-GPT in (Yang et al., 2024). We set all other hyperparameters to their default values.

B Detection Evidence of Baselines

RoBERTa with SHAP. Figure 8 depicts an example of evidence by RoBERTa with SHAP. We visualize the evidence using the SHAP library[10]. The red parts are spans that contribute to predicting LLM-generated. The blue parts are spans that contribute to predicting human-written. In the evidence, if the prediction value, $f(\text{inputs})$ moves further to the right compared to the base value (the expected value across all data samples), it is more likely to be LLM-generated. When we hover over a colored part, we can see a score of how much the part contributes to the result. The more a span contributes to the decision, the darker its color.

LR-GLTR. Figure 9 displays an example of evidence by LR-GLTR. We leverage a demo app[11] of GLTR by Gehrmann et al. (2019). It highlights tokens in different colors based on their rank of top-{10, 100, 1,000, 1,000+} in the predicted token distribution from an LLM. The higher the rank of the token, the more likely an LLM is to generate the token. The green parts are spans that an most likely LLM-generated. The degree decreases in the order of green, yellow, red, and purple. When we hover a cursor on a colored part, we can also see the predicted token distribution of an LLM.

DNA-GPT. Figure 10 shows an example of evidence by DNA-GPT. We implemented a demo app of DNA-GPT with the streamlit framework[12]. It shows overlapped n-gram spans between a truncated target text and multiple LLM-generated continuations. The more blue spans, the more likely the text is LLM-generated. For span matching, we follow the original implementation of DNA-GPT[13] where it was achieved by token-level matching based on preprocessing of the lower casing and stemming. We also set n to 8 in order to show a large number of overlapped spans enough to interpret as evidence.

C Analysis Details

Impact of α. Figure 11 showcases the impact of α on the detection performance of ExaGPT across four domains and three generators. We found similar overall trends of the impact of α in other LLMs, including GPT-4 and Dolly-v2, with the impact in ChatGPT, as explained in §4.

Impact of the Datastore Size. Figure 12 showcases the impact of the datastore size on the detection performance of ExaGPT across four domains and three generators. We can observe similar overall trends of the impact of datastore size in other LLMs, including GPT-4 and Dolly-v2, with the impact in ChatGPT as explained in §4.

D Computational Budget

We run all the experiments with two AMD EPYC 7453 CPUs and four NVIDIA A6000 GPUs. The total processing time is approximately 25 hours.

[10] https://shap.readthedocs.io/

[11] http://demo.gltr.io/client/index.html

[12] https://github.com/streamlit/streamlit

[13] https://github.com/Xianjun-Yang/DNA-GPT

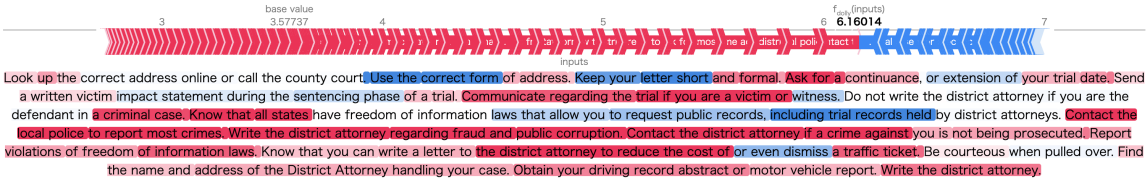

Figure 8: Example of evidence by RoBERTa with SHAP.

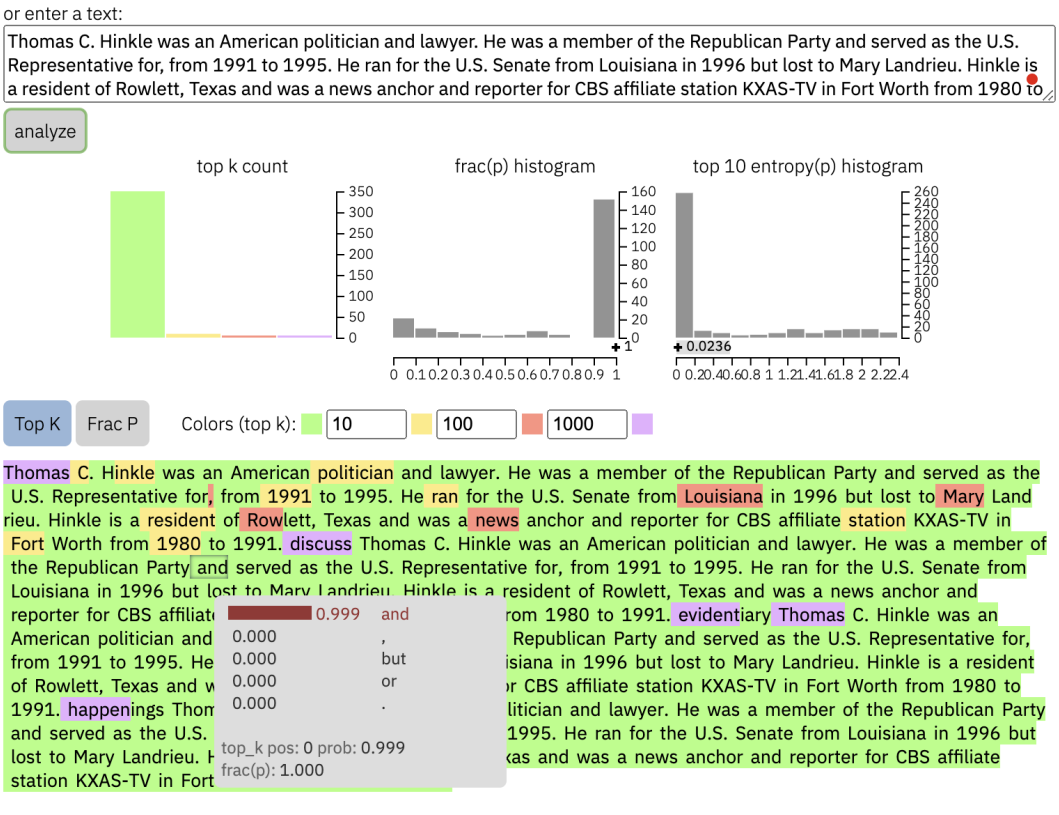

Figure 9: Example of evidence by LR-GLTR.

templat the possibl option of an apprenticeship or on the job train program an apprenticeship or on the job train program is an excel way to gain hand on experi in the automot technician trade while you earn money consid thi option if you prefer a more hand on learn experi 10 contact your guidanc counselor if you are a high school student consid enter the automot technician trade if you are a high school student talk to your guidanc counselor about the vocat train program avail in your area they can provid guidanc on what program are best suit for your goal and interest 11 visit sever websit that offer educ train in the automot technician trade research variou vocat train program onlin to compar offer requir and cost make sure you read review from previou student to see what they have to say about the program 12 select the best program for your need interest and career goal after review your option choos the program that best fit your need interest and career goal 13 chart out your final plan and proceed with confid onc you have select your program and commit to your career plan out your step to achiev your goal stay motiv and confid in your abil to succeed as an automot technician

0.

templat the possibl option of an apprenticeship or on the job train program depend on the program apprenticeship and on the job train can be a great way to gain practic experi and learn from experienc profession while earn an incom consid if thi is the right path for you 10 contact your guidanc counselor if you are a high school student consid enter the automot technician trade high school often offer vocat train program in automot technolog speak with your guidanc counselor to learn more about these program and if they are avail to you 11 visit sever websit that offer educ train in the automot technician trade research onlin for differ vocat train program and read through their program curriculum determin which program will best suit your desir career path 12 select the best program for your need interest and career goal choos the program that align with your career aspir and offer the level of train and certif you desir 13 chart out your final plan and proceed with confid now that you ve research and weigh your option it s time to start your train program stick to your plan and be confid in your decis to becom an automot technician with hard work and dedic you can have a success career in the field of automot technolog

1.

templat the possibl option of an apprenticeship or on the job train program apprenticeship and on the job train program provid a more hand on approach to learn the automot technician trade consid thi option if you prefer a more practic and immers learn environ 10 contact your guidanc counselor if you are a high school student consid enter the automot technician trade your guidanc counselor can provid you with valuabl resourc and inform about vocat train program and opportun 11 visit sever websit that offer educ train in the automot technician trade research differ vocat train program and school that offer automot technolog cours look for program that have a good reput and offer hand on train opportun 12 select the best program for your need interest and career goal choos a program that fit with your learn style schedul and goal look for program that provid a well round educ in differ area of automot technolog 13 chart out your final plan and proceed with confid onc you have select a program make a plan to achiev your goal and proceed with confid stay commit to your train and take advantag of any opportun to gain hand on experi and network with other profession in the field conclus becom an automot technician requir dedic skill and a willing to learn research the field evalu your strength and weak and determin the best type of vocat train program for your need consid your goal and determin if you want to special in a particular area of the trade or pursu a

Figure 10: Example of evidence by DNA-GPT.

Figure 11: Impact of α on the detection performance of ExaGPT, including the AUROC and the accuracy at 1% FPR, across four domains and three generators.

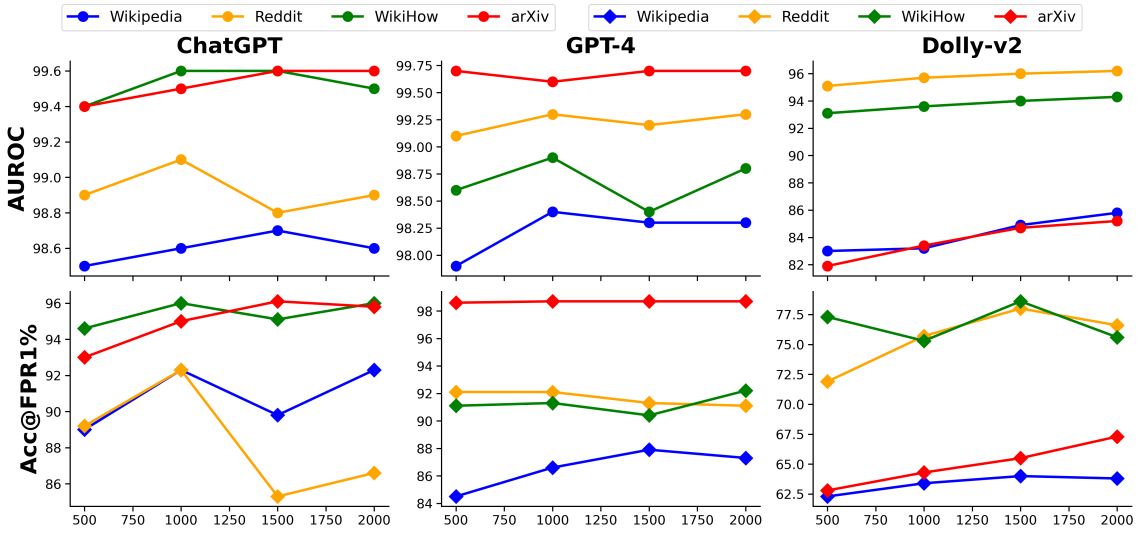

Figure 12: Impact of the datastore size on the detection performance of ExaGPT, including the AUROC and the accuracy at 1% FPR, across four domains and three generators.