DART🎯: An AIGT Detector using AMR of Rephrased Text

Hyeonchu Park[†], **Byungjun Kim**[†], **and Bugeun Kim**
Department of Artificial Intelligence, Chung-Ang University, Republic of Korea
{phchu0429, k36769, bgnkim}@cau.ac.kr

Abstract

As large language models (LLMs) generate more human-like texts, concerns about the side effects of AI-generated texts (AIGT) have grown. So, researchers have developed methods for detecting AIGT. However, two challenges remain. First, the performance of detecting black-box LLMs is low because existing models focus on probabilistic features. Second, most AIGT detectors have been tested on a single-candidate setting, which assumes that we know the origin of an AIGT and which may deviate from the real-world scenario. To resolve these challenges, we propose DART🎯, which consists of four steps: rephrasing, semantic parsing, scoring, and multiclass classification. We conducted three experiments to test the performance of DART. The experimental result shows that DART can discriminate multiple black-box LLMs without probabilistic features and the origin of AIGT.

1 Introduction

As large language models (LLMs) continue to advance, it becomes increasingly difficult for humans to discern AI-generated text (AIGT). This poses issues in society and research, such as spreading fake news and tainting AI training data. Researchers have developed AIGT detectors to address these issues. Despite their success, two challenges related to real-world applicability persist.

One challenge with applying AIGT detectors is low performance in detecting recent black-box LLMs. Traditionally, AIGT detectors rely on probabilistic features such as logits. However, in commercial black-box models, including GPT (OpenAI, 2024a,b) or Gemini (Team et al., 2024), we cannot access their computation procedure which provides logits. That is, traditional approaches cannot detect such black-box models. So, researchers have also designed detectors using syntactic features that do not require accessing computational procedures. Yet, these detectors struggle to detect black-box models because their syntactic naturalness is comparable to that of humans.

The other challenge is the vagueness of the origin of AIGTs. In the inference time of a detector, it receives a text without any information about its origin. So, similar to the inference scenario, we should verify whether a detector can successfully discriminate AIGT regardless of source models. However, existing studies mainly tested their detectors under the assumption that a candidate LLM is known in advance; they tested whether a binary detector can distinguish a 'human-written text' from an 'AIGT by the predefined candidate.' So, whether existing detectors can detect the origin without the assumption is questionable.

To address these challenges, we propose a Detector using AMR of Rephrased Text (DART🎯). DART utilizes the semantic gap between given input and rephrased text, using Abstract Meaning Representation (AMR). This rephrasing idea was first introduced by RAIDAR (Mao et al., 2024); we adopted a similar idea to reveal such a semantic gap. To examine the real-world detection performance, we assess DART in three settings: single-candidate, multi-candidate, and leave-one-out. Experimental results show that DART can successfully discriminate humans from four cutting-edge LLMs, including GPT-3.5-turbo, GPT-4o, Llama 3-70b (Dubey et al., 2024), and Gemini-1.5-Flash.

Thus, this paper has the following contributions:

- We present a semantics-based detection framework for AIGT, leveraging semantic gaps between given input text and rephrased texts.
- DART can discriminate different LLMs and outperform other models. On average, DART beat others by more than 19% in F1 score.

[†]Equal contribution.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), pages 710–721

- Also, DART can generalize its knowledge on detecting unseen source models. Specifically, DART achieved a 85.6% F1 score on leave-one-out experiment.

2 Background

In this section, we categorize existing studies regarding numbers (*single* or *multi*) and transparency (*white box* or *black box*) of candidate LLMs.

Single white-box candidates AIGT detectors first attempted to extract candidate-specific features. As the candidate is a known white-box model, some researchers designed algorithms adopting probabilistic features from the model (Gehrmann et al., 2019; Mitchell et al., 2023). For example, DetectGPT (Mitchell et al., 2023) used log probabilities of tokens as features. Other researchers used neural models that can learn features from the given texts (Solaiman et al., 2019; Hu et al., 2023). However, as many black-box LLMs recently emerged, the performance of existing detectors should be revalidated on those LLMs.

Single black-box candidates Some AIGT detectors then attempted to extract features regardless of the candidate (Bao et al., 2024; Yu et al., 2024; Yang et al., 2023; Kim et al., 2024), as black-box candidates may not provide probabilistic features. Fast-DetectGPT (Bao et al., 2024) extended DetectGPT by extracting probabilistic features from a proxy white-box model (e.g., GPT-J). Since such a proxy can provide less accurate results, other studies used syntactic or surface-level features without using a proxy (Yang et al., 2023; Kim et al., 2024). For example, DNA-GPT (Yang et al., 2023) used n-grams from multiple paraphrased texts generated by the candidate. However, such syntactic features are insufficient to detect recent LLMs because recent models generate text with human-level syntax.

Multiple candidates As a single-candidate performance is far from real-world scenarios, recent AIGT detectors were designed to detect multiple candidates (Li et al., 2023; Abburi et al., 2023; Wang et al., 2023; Shi et al., 2024; Antoun et al., 2024). For example, POGER (Shi et al., 2024) extends resampling methods to estimate probability using about 100 paraphrases. Because of such an excessive regeneration, POGER incurs high computational costs. Besides, SeqXGPT (Wang et al., 2023) used a Transformer-based detector with a proxy model estimating probabilistic features. However, these studies mainly focused on surface-level features and the limited range of LLMs (e.g., GPT family), raising questions about detecting other cutting-edge LLMs.

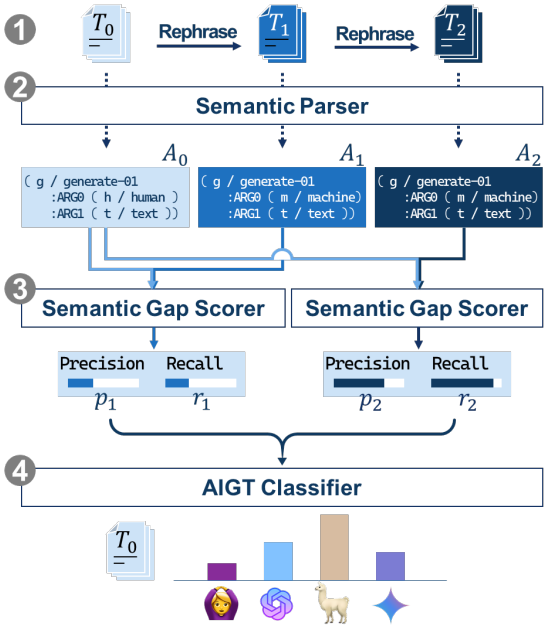


Figure 1: The DART framework

3 The DART Framework

As shown in Figure 1, DART utilizes semantic gaps between a given text and rephrased texts. To train a detector capturing such gaps, DART uses a four-step procedure: *Rephrasing*, *Semantic parsing*, *Semantic gap scoring*, and *Classification*.

Step 1, Rephrasing: We hypothesized that rephrasing texts could reveal the difference between humans and AI in the way they express semantics. To obtain the rephrased texts, DART uses a *rephraser* module that generates semantically closer text T_1 from a given text T_0. Further, we let the rephraser generate another rephrased text T_2 by giving T_1 to attain additional features. To avoid generating rephrased texts irrelevant to the given input, we need a reliable rephraser that can preserve semantics. So, we adopted `GPT-4o-20240513` as our rephraser because the model showed the highest performance in semantics-related tasks (OpenAI, 2024a). Appendix A.2 details the prompts used in the rephrasing step.

Step 2, Semantic parsing: DART adopts a semantic parser to transform texts into semantic representations. We especially adopted AMR as a

semantic representation because AMR has widely been adopted to abstract the given text into semantics (Banarescu et al., 2013). For the parser, we adopted Naseem et al. (2022). As a result, the parser constructs an AMR graph A_i from each T_i.

Step 3, Semantic gap scoring: DART uses metrics for semantic parsers to measure semantic gaps between texts. As we adopted AMR as a semantic representation in the previous step, we utilize a fast and efficient algorithm for scoring AMR similarity called SEMA (Anchiêta et al., 2019; Ki et al., 2024). To obtain semantic gaps between A_0 and A_i ($i > 0$), DART computes precision p_i and recall r_i scores generated by SEMA, resulting a feature vector $v = [p_1, p_2, r_1, r_2]^\top$ for the next step.

Step 4, Classification: DART has a classifier that predicts one possible origin of T_0. DART uses interpretable classifiers, including support vector machine (SVM) or decision tree (DT), though any classifier that maps v to origins can be used.

4 Experiments

To evaluate the performance of DART, we conducted three experiments: (1) single-candidate, (2) multi-candidate, and (3) leave-one-out settings. First, in the single-candidate setting, we formulate AIGT detection as a binary classification task. Assuming that AIGTs are exclusively produced by a specific LLM, a detector should predict whether the given text is produced by the LLM. Second, in the multi-candidate setting, we formulate the task as a multi-label classification. After training on AIGTs from multiple candidate sources, a detector should decide the source of the given input text among the candidates. Third, in the leave-one-out setting, we test the generalizability of detectors. We examined whether a detector can successfully classify AIGTs from models that were unseen during the training.

We ran each experiment 10 times for each experiment to achieve reproducibility. Further, we analyzed DART's training efficiency by examining the decreasing rate of detecting performance as the size of the training dataset.

4.1 Datasets

To train DART, we need human-written texts and AIGTs. First, we used four English datasets as human-written text datasets: XSum (Narayan et al., 2018), SQuAD 1.1 (Rajpurkar et al., 2018), Reddit (Fan et al., 2018), and PubMedQA (Jin et al., 2019). Following the practice of previous research (Mitchell et al., 2023; Wang et al., 2023), we randomly sampled texts from these datasets. We split training and validation sets with an 8:2 ratio.

Second, we generated AIGT datasets based on the human dataset. Following Mitchell et al. (2023), we collected English AIGT from each human-written text. Four cutting-edge LLMs are used to generate AIGTs: GPT-4o, GPT-3.5-turbo, Llama 3-70B, and Gemini-1.5 Flash. We obtained AIGTs by providing the first 30 tokens of each human-written text to an LLM. Because PubMedQA contains many texts shorter than 30 tokens, we provided corresponding questions instead of the first 30 tokens. Appendix A.1 illustrates the detailed prompts used for generating AIGTs. As a result, we obtained about 3,989 human-written texts and 15,956 AIGTs (= 3,989 texts $\times$ 4 LLMs). See Appendix B.2 for the statistics of the collected dataset.

4.2 Baselines

As baselines, we used five open-source state-of-the-art detectors: DetectGPT (Mitchell et al., 2023), Fast-DetectGPT (Bao et al., 2024), DNA-GPT (Yang et al., 2023), Roberta-base (Solaiman et al., 2019), and SeqXGPT (Wang et al., 2023). Among these models, DetectGPT, Fast-DetectGPT, and SeqXGPT used probabilistic features generated by third-party LLMs in order to detect cutting-edge LLMs. Meanwhile, DNA-GPT and Roberta-base used shallow semantic features, such as n-grams or contextual embeddings. DART stands out from these models because it uses AMR-based semantics rather than probabilistic features.

We used a different set of detectors for the three experiments, considering experiments reported with five baselines. For the single-candidate experiment, we compared DART with all five detectors. For the multi-candidate and the leave-one-out experiments, we compared DART only with SeqXGPT, as it is the only existing detector that can trained on multiple candidates simultaneously. To ensure a fair comparison, all detectors used in the experiment are trained on our dataset from scratch[1]. To measure the performance, we used the F1 score.

[1]Note that we used GPT-2 as a proxy model for the GPT series and Gemini-1.5 when the detectors require probabilistic features because GPT and Gemini do not provide logits, following (Bao et al., 2024).

	Average	GPT-3.5-turbo	GPT-4o	Llama3-70B	Gemini-1.5
DetectGPT*	65.8	65.8±0.20	65.6±0.16	65.8±0.17	65.7±1.12
fast-DetectGPT*	60.1	58.0±1.94	66.2±0.25	62.4±0.48	53.8±0.58
DNA-GPT	54.1	56.6±1.49	57.4±0.50	54.8±2.60	47.7±2.36
Roberta-base	77.2	76.8±3.24	80.0±2.81	74.7±1.77	77.1±2.13
SeqXGPT*	54.1	86.5±0.48	45.9±0.23	41.6±0.31	42.3±0.52
DART_{SVM}	82.8	87.1±0.65	86.1±0.70	84.8±2.20	73.3±0.76
DART_{DT}	**96.5**	**100.0±0.03**	**88.1±0.98**	**100.0±0.03**	**97.9±1.65**

* Models used GPT-2 as a proxy model, except Llama 3.

Table 1: F1 scores of detectors in the **single-candidate** experiment, with standard deviations reported.

5 Result and Discussion

Single-candidate experiment: DART outperformed existing models. As shown in Table 1, our DART_{DT} and DART_{SVM} achieved 96.5% and 82.8% F1 scores on average, which are 19.3%p and 5.6%p higher than the Roberta-base model (77.2%). Also, DART_{DT} can detect all four cutting-edge models with over 85% of F1 score. Meanwhile, other existing models showed F1 scores lower than 70%, on average. Moreover, DNA-GPT and SeqXGPT sometimes showed F1 scores lower than the random binary baseline (50%).

We suspect that DART_{DT} can achieve such outstanding performance because our semantic scoring step can successfully form several clusters according to their origins. To support this argument, we further analyzed the feature vectors of DART using principal component analysis. We found that texts sharing the same source usually form several independent clusters rather than spread over the space. Detailed results are presented in Appendix C.3.

Multi-candidate experiment: DART also outperformed SeqXGPT. As shown in Table 2, our DART_{DT} and DART_{SVM} achieved 81.2% and 65.0% macro F1 scores, which are 22.0%p and 5.8%p higher than SeqXGPT (59.2%). Interestingly, SeqXGPT achieved the lowest F1 score on detecting Llama 3 (44.8%), but DART_{DT} achieved the lowest score on detecting GPT-4o (76.6%).

We suspect how the detectors extract features using an LLM affects the performance. We present a contingency table of SeqXGPT and DART_{DT} to support this claim, as shown in Figure 2. The figure shows that (i) SeqXGPT struggled in distinguishing models other than Llama 3, and (ii) DART_{DT} struggled in distinguishing the GPT family and humans. Since SeqXGPT in our experiment used

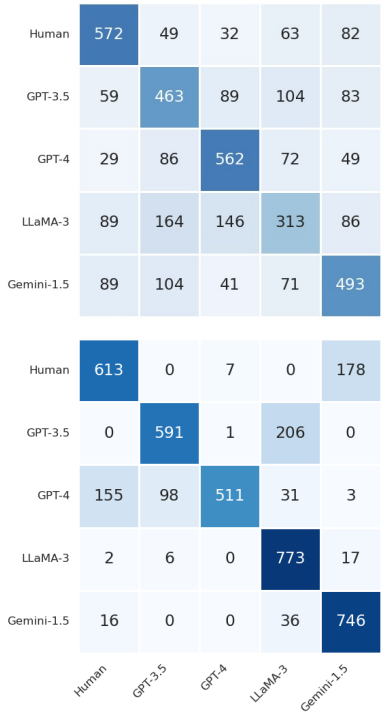


Figure 2: Contingency matrix from multi-candidate experiment. Top (a) and Bottom (b) correspond to SeqXGPT and DART_{DT}. Actual and predicted classes are depicted as horizontal and vertical axes.

GPT-2 as a proxy model, and DART_{DT} used GPT-4o as a *rephraser* module, the characteristics of the used LLMs affected the detection performances. For example, as DART_{DT} utilizes semantic gaps between the original and rephrased texts, origins should reveal distinguishable gaps to identify them successfully. So, when the gaps are too similar between origins to discriminate them, DART_{DT} faces difficulty in the classification step.

Since GPT-4o has a similar language understanding ability to humans (OpenAI, 2024a), GPT-4o and humans may be less distinguishable through

	Macro F1	GPT-3.5-turbo	GPT-4o	Llama3-70B	Gemini-1.5	Human
SeqXGPT*	59.2±0.66	54.3±1.44	66.4±1.08	44.8±0.95	61.4±1.68	69.3±0.93
DART_{SVM}	65.0±0.77	67.4±0.81	71.0±1.20	54.0±1.26	67.0±0.94	65.4±1.16
DART_{DT}	**81.2±1.71**	**80.6±4.61**	**76.6±1.16**	**85.8±5.42**	**85.5±2.36**	**77.3±0.88**

* Models used GPT-2 as a proxy model, except Llama 3.

Table 2: F1 scores of detectors in the **multi-candidate** experiment, with standard deviations reported.

	Macro F1	GPT-3.5-turbo	GPT-4o	Llama3-70B	Gemini-1.5
SeqXGPT*	78.5±1.04	79.9±1.39	80.2±0.52	78.8±0.92	75.1±1.31
DART_{SVM}	56.3±0.96	56.0±1.31	59.2±1.01	56.4±0.82	53.6±0.70
DART_{DT}	**84.2±1.39**	**99.3±0.16**	**75.8±3.82**	**99.1±0.55**	**62.5±1.03**

* Models used GPT-2 as a proxy model for black-box models, except Llama3

Table 3: F1 scores of detectors in the **leave-one-out** experiment, with standard deviations reported.

gaps. Similarly, as GPT-3.5-turbo may share some core knowledge with GPT-4o, GPT-4o can be confused with GPT-3.5-turbo in DART_{DT}.

Leave-One-Out experiment: DART_{DT} showed the best performance. As shown in Table 3, DART_{DT} achieved 85.6% average F1 score, followed by SeqXGPT (77.9%) and DART_{SVM} (56.5%). Besides, DART_{DT} scored 62.5% F1 on detecting the unseen Gemini-1.5, though DART_{DT} recorded more than 75% on detecting others.

This result indicates that DART_{DT} can generalize trained knowledge to detect unseen source models. That is, DART_{DT} can discriminate new candidate models from humans. Specifically, compared to the single-candidate result (Table 1), our model showed almost similar performance on detecting GPT-3.5-turbo and Llama 3 without training on those models. As in the single-candidate experiment, we believe that our semantic scoring step helped to detect unseen models because they form clusters independent from humans. Also, when the cluster becomes indiscernible with humans, DART_{DT} struggles to detect new models. For example, DART_{DT} showed a big performance drop when excluding Gemini-1.5 from the training set because DART_{DT} often confused Gemini-1.5 with humans (top-right corner on Figure 2b).

Training efficiency of DART: Here, we discuss the general tendency of the result. Figure 3 shows the performance changes when we decrease the size of the training set. Detailed result of training efficiency is presented in Appendix C.4. The result shows that DART_{DT} is robust even though we use a small amount of training data. Specifically, DART_{DT} maintained a similar F1 score until we used 1/32 of the training set (about 500 examples). Meanwhile, the performance of SeqXGPT and DART_{SVM} monotonically decreases as we reduce the size of the training set.

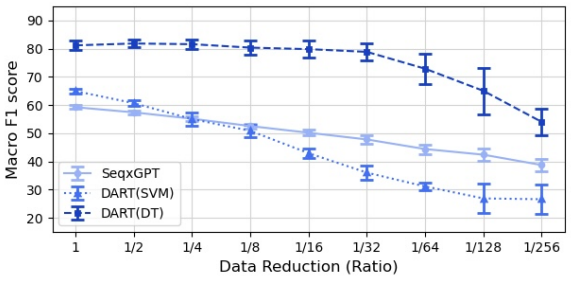


Figure 3: F1 score of detectors when we decrease the amount of training data in multi-candidate experiment.

6 Conclusion

We introduced an AIGT framework, DART🎯 to tackle challenges in applying AIGT detectors to real-world scenarios. DART employed *rephraser* and semantic gap scoring module to address the challenges of black-box models. To evaluate whether DART can address vagueness of origin, we assessed DART in three experimental settings: single-candidate, multi-candidate, and leave-one-out settings. As a result, DART achieved outstanding performance compared to existing AIGT detectors, demonstrating successful capture of differences across origins with semantic gaps.

Limitations

Despite the outstanding performance of DART, this paper has three limitations. First, we tested DART only with a single rephraser LLM, GPT-4o. Though GPT-4o provided enough semantic information to distinguish AIGTs successfully, it is questionable whether DART can be used with other rephraser LLMs, such as Llama 3, Gemini Pro, or others. Also, we recognize the cost implications of utilizing GPT-4o as a rephraser, which could restrict its applicability in resource-limited environments. Since different language models may provide different rephrased texts with lower costs, we need further study to determine how much rephraser LLM affects the performance.

Second, the performance of the adopted AMR parser may affect the detection performance of DART. Though the AMR parser rarely introduces errors in the DART framework, such errors may lead to huge changes in detection performance when they occur. Using a publicly available AMR parser (Naseem et al., 2022), DART showed the lowest bound of its performance. Thus, we need further study to improve the performance using other semantic parsers.

Third, DART tested on a narrow range of black-box models. While narrow LLMs have become publicly available through paid APIs or pretrained parameters, we tried our best to include recent LLMs, such as Gemini Pro or Claude 3. However, we finally excluded those models because their safeguards prevented from generating AIGTs based on a given human-written text when preparing the AIGT dataset. To generalize our findings to other origins, we need to conduct further studies in a broader range of models and design a new method of generating AIGTs.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-01341, Artificial Intelligence Graduate School Program, Chung-Ang University)

References

Harika Abburi, Michael Suesserman, Nirmala Pudota, Balaji Veeramani, Edward Bowen, and Sanmitra Bhattacharya. 2023. Generative ai text classification using ensemble llm approaches. In *IberLEF@SEPLN*, volume 3496 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Rafael Torres Anchiêta, Marco Antonio Sobrevilla Cabezudo, and Thiago Alexandre Salgueiro Pardo. 2019. Sema: an extended semantic evaluation for amr. In *(To appear) Proceedings of the 20th Computational Linguistics and Intelligent Text Processing*. Springer International Publishg.

Wissam Antoun, Benoît Sagot, and Djamé Seddah. 2024. From text to source: Results in detecting large language model-generated content. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7531–7543, Torino, Italia. ELRA and ICCL.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

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.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. RADAR: robust ai-text detection via adversarial learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

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.

Kyung Seo Ki, Bugeun Kim, and Gahgene Gweon. 2024. Inspecting soundness of AMR similarity metrics in terms of equivalence and inequivalence. In *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*, pages 402–409, Mexico City, Mexico. Association for Computational Linguistics.

Zae Myung Kim, Kwang Lee, Preston Zhu, Vipul Raheja, and Dongyeop Kang. 2024. Threads of subtlety: Detecting machine-generated texts through discourse motifs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5449–5474, Bangkok, Thailand. Association for Computational Linguistics.

Linyang Li, Pengyu Wang, Ke Ren, Tianxiang Sun, and Xipeng Qiu. 2023. Origin tracing and detecting of llms. *Preprint*, arXiv:2304.14072.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. Raidar: generative AI detection via rewriting.

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.

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.

Tahira Naseem, Austin Blodgett, Sadhana Kumaravel, Tim O'Gorman, Young-Suk Lee, Jeffrey Flanigan, Ramón Astudillo, Radu Florian, Salim Roukos, and Nathan Schneider. 2022. DocAMR: Multi-sentence AMR representation and evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3496–3505, Seattle, United States. Association for Computational Linguistics.

OpenAI. 2024a. Hello gpt-4o. `https://openai.com/index/hello-gpt-4o/`. Accessed: 2024-10-13.

OpenAI. 2024b. Introducing chatgpt. `https://openai.com/index/chatgpt/`. Accessed: 2024-10-13.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Yuhui Shi, Qiang Sheng, Juan Cao, Hao Mi, Beizhe Hu, and Danding Wang. 2024. Ten words only still help: Improving black-box ai-generated text detection via proxy-guided efficient re-sampling. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 494–502. International Joint Conferences on Artificial Intelligence Organization. Main Track.

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. *Preprint*, arXiv:1908.09203.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023. SeqXGPT: Sentence-level AI-generated text detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1144–1156, Singapore. Association for Computational Linguistics.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Petzold, William Yang Wang, and Haifeng Chen. 2023. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text.

Xiao Yu, Yuang Qi, Kejiang Chen, Guoqiang Chen, Xi Yang, Pengyuan Zhu, Xiuwei Shang, Weiming Zhang, and Nenghai Yu. 2024. Dpic: Decoupling prompt and intrinsic characteristics for llm generated text detection. *Preprint*, arXiv:2305.12519.

A Prompts

A.1 AIGT datasets

In general, we followed the prompts used in SeqXGPT (Wang et al., 2023) when generating the AIGT dataset. We collected AIGTs by providing LLMs with the first 30 tokens of human-written texts and letting them generate the rest of the texts, except for the PubMedQA dataset. Besides, we asked LLMs to answer the questions in the PubMedQA dataset instead of providing the 30 tokens of text, borrowing the collecting method of Mitchell et al. (2023). We used different methods for PubMedQA because most of the texts in PubMedQA were shorter than 30 tokens. In addition, to avoid collecting AIGTs with irrelevant phrases (e.g., "Here is the generation of ..."), we added a constraint clause in the prompts for Llama 3-70B and Gemini-1.5 Flash.

We understand that different datasets and different prompting methods may affect the performance of the detectors. Therefore, we conducted additional per-subset experiments to investigate whether those differences influenced the detecting performance. The findings are detailed in Appendix C.2.

For GPT family When collecting AIGTs with GPT-3.5-turbo and GPT-4o, we used the following prompts except for the PubMedQA dataset.

```
Please provide a continuation for
the following content to make it
coherent: {first 30 tokens}
```

For PubMedQA, we used the following prompts:

```
Please answer the question:
{question}
```

For Llama 3-70B and Gemini-1.5-Flash When collecting AIGTs with Llama 3-70B and Gemini-1.5-Flash, we used the following prompts except for the PubMedQA dataset.

```
Please provide a continuation for
the following content to make it
coherent: {first 30 tokens}
Provide the continuation without
any prefix.
—
answer:
```

	T_0	T_1	T_2
Human	267.95	258.47	270.38
GPT-3.5-T	107.48	89.85	83.08
GPT-4o	260.03	253.59	262.56
Llama3	152.33	133.94	127.69
Gemini-1.5	131.32	116.74	110.25

Table 4: Average number of words after rephrasing

	Mac F1	Xsum	SQuad	Reddit	PubMed
SeqXGPT*	63.0	75.1	57.0	58.2	61.7
DART_{SVM}	88.8	80.0	92.4	93.2	89.8
DART_{DT}	98.6	99.0	98.4	98.6	98.4

Table 5: Performance of AIGT detectors across different subsets in a Multi-Candidate setting

For PubMedQA, we used the following prompts:

```
Please answer the question:
{question}
Provide the continuation without
any prefix.
—
answer:
```

A.2 DART's *rephraser*

When rephrasing a text into another rephrased version, we used the following prompt in the rephraser module.

```
Please rewrite the following
paragraph in {n} words: {paragraph}
```

We used this prompt because we observed some semantic meanings of rephrased texts were largely changed without any prompting method in our pre-experiment. For example, some rephrased texts were much longer or shorter than the original texts, which was enough to distort the core message of the origins. As such distortion leads to unintended trivial semantic differences, we wanted to avoid such too-short or too-long texts. Thus, we restricted the word counts of rephrased texts by using prompts. Table 4 on page 8 shows the average number of words in the original and rephrased texts that we collected. It shows that the number of words slightly changed after rephrasing. We believe that such changes are minor to affect the performance of DART.

B Experimental setting

B.1 Environment

Hardware configuration: The experiments were conducted on a system with an AMD Ryzen Threadripper 3960X 24-Core Processor and four NVIDIA RTX A6000 GPUs. The four NVIDIA RTX A6000 GPUs are used to train existing detectors and execute AMR parsers. The semantic gap scoring module was run on a single core of the CPU.

LLM APIs: We used commercial APIs of LLMs to collect AIGTs and rephrased texts. GPT models are called with OpenAI's official API. Llama 3-70B is called with a free API provided by `groq.com`. Lastly, Gemini-1.5-Flash is called with OpenRouter's API.

Implementation We used Python 3.11.7 for implementing DART🎯. Using `scikit-learn` library, we implemented DART_{SVM} and DART_{DT} with `SVC` and `DecisionTreeClassifier`. We mostly used the basic settings of those classes without conducting a hyperparameter search. The only exception is the depth of the pruned tree in DART_{DT}, and we set it as 5 based on our heuristic.

B.2 Dataset statistics

Table 7 in page 10 shows the statistics of the collected dataset. We used four datasets, which belong to different domains: Xsum (Narayan et al., 2018), SQuAD (Rajpurkar et al., 2018), Reddit (Fan et al., 2018), and PubMedQA (Jin et al., 2019). Xsum is a dataset of news articles and summaries. SQuAD is a question-answering dataset whose questions are based on Wikipedia articles. Reddit is a dataset of human-written stories with writing prompts. PubMedQA is a question-answering dataset on a specialized medical domain.

The statistics show that the average lengths of texts in each dataset are different. For example, Gemini-1.5 usually generates long texts on the PubMedQA dataset, while the model generates short texts on the Xsum and Reddit datasets. On average, it seems that the length of a given text is not a significant factor for discriminating origin.

C Additional analysis

C.1 Precision, Recall

As we discussed in Section 3, DART computes precision p and recall r scores with SEMA. Note that p_i and r_i refer to the semantic similarity between the original text T_0 and the i-th rephrased text T_i. DART assumes that the differences between those rephrased texts in terms of p and r values can be used to identify AIGTs. In this section, we provide evidence that supports the assumption by comparing the trend of p and r values.

	p_1	p_2	r_1	r_2
Human	0.619	0.582	0.600	0.561
GPT-3.5-T	0.645	0.605	0.631	0.595
GPT-4o	0.636	0.596	0.623	0.587
Llama3	0.648	0.610	0.631	0.594
Gemini-1.5	0.651	0.615	0.633	0.596

Table 6: Precision and Recall values for text comparisons between T_0, T_1 and T_0, T_2

Table 6 on page 9 illustrates the average of precision and recall values we collected. On average, the table shows that p_2 and r_2 are smaller than p_1 and r_1, respectively. This indicates that T_2 was semantically far from T_0 than T_1. So, as we apply rephraser more times on T_0, the semantics of rephrased text becomes farther from T_0.

Also, the result shows that p and r values are lower in human-written texts than AIGTs. For example, human-written text showed p_1 of 0.619, which is lower than AIGTs (ranging from 0.636 to 0.651). So, it is reasonable to use these values to distinguish between human-written texts and AIGTs.

C.2 Effect of prompt and domain changes

Since we used different prompting methods and datasets in generating AIGTs, we conducted the per-subset experiment to investigate whether those differences affected the performance of detectors. Specifically, we conducted multi-candidate experiments for each subset. For example, instead of using all data, we trained and tested models only with texts from PubMedQA.

Table 5 on page 8 shows the results of the per-subset experiment. Though the domains and prompting methods are different across those subsets, DART_{DT} achieved consistently high-performance scores by showing 98.6% macro F1. Also, DART_{SVM} (ranging from 80.0% to 93.2%) showed better consistency than SeqXGPT (ranging from 57.0% to 75.1%). This result indicates that DART🎯 models are robust on changes of domains or prompting methods compared to SeqXGPT.

C.3 Principal components of features

Figure 4 and 5 in page 11 display PCA plots of features used in DART. The figures show that each source makes several clusters. Here, we attempt to interpret DART's experimental results by analyzing the PCA results. The distribution of feature vectors may affect the performance of SVM and DT classifiers. As SVM seeks a global decision boundary that maximizes margin, SVM may not find a clear decision boundary between multiple mini clusters. Meanwhile, DT can split such mini clusters based on multiple criteria. So, DT could achieve high performance in discriminating AIGTs from human-written texts. For example, we can easily discriminate humans from others and iteratively build different decision boundaries between smaller clusters. As a result, DART_{DT} can clearly discriminate sources and showed higher performance than DART_{SVM}.

C.4 Training efficiency on single-candidate setting

Figure 6 in page 12 shows the training efficiency on the single-candidate experiment. In general, the performance drops as the size of the dataset decreases. Among those models, DART_{DT} demonstrates the best performance across all models, even with small datasets. DART_{SVM} experiences a more rapid decrease in its performance.

We suspect that the distribution of the data may affect the classification performance. In other words, SVM or a neural network may not have sufficient data to distinguish small clusters whose features are close to each other when we use a small dataset.

	# char	# tokens	# sample
PubMedQA dataset			
Human	265.9	41.8	995
LLMs	1132.4	188.1	3980
GPT-3.5T	496.2	78.2	995
GPT-4o	1181.4	192.6	
Llama 3-70B	1327.5	212.7	
Gemini-1.5F	1524.7	268.9	
Xsum dataset			
Human	2194.5	428.9	999
LLMs	909.5	160.5	3996
GPT-3.5T	773.7	136.5	999
GPT-4o	1627.8	282.4	
Llama 3-70B	671.9	121.6	
Gemini-1.5F	564.7	101.5	
Reddit dataset			
Human	2962.7	641.0	997
LLMs	1135.5	237.3	3988
GPT-3.5T	852.3	176.7	997
GPT-4o	1986.7	413.5	
Llama 3-70B	1009.5	213.0	
Gemini-1.5F	691.4	146.0	
SQuAD dataset			
Human	740.2	135.1	998
LLMs	947.5	157.0	3992
GPT-3.5T	503.4	79.1	998
GPT-4o	1803.1	303.6	
Llama 3-70B	809.7	142.4	
Gemini-1.5F	673.9	102.8	

Table 7: Statistics of collected datasets

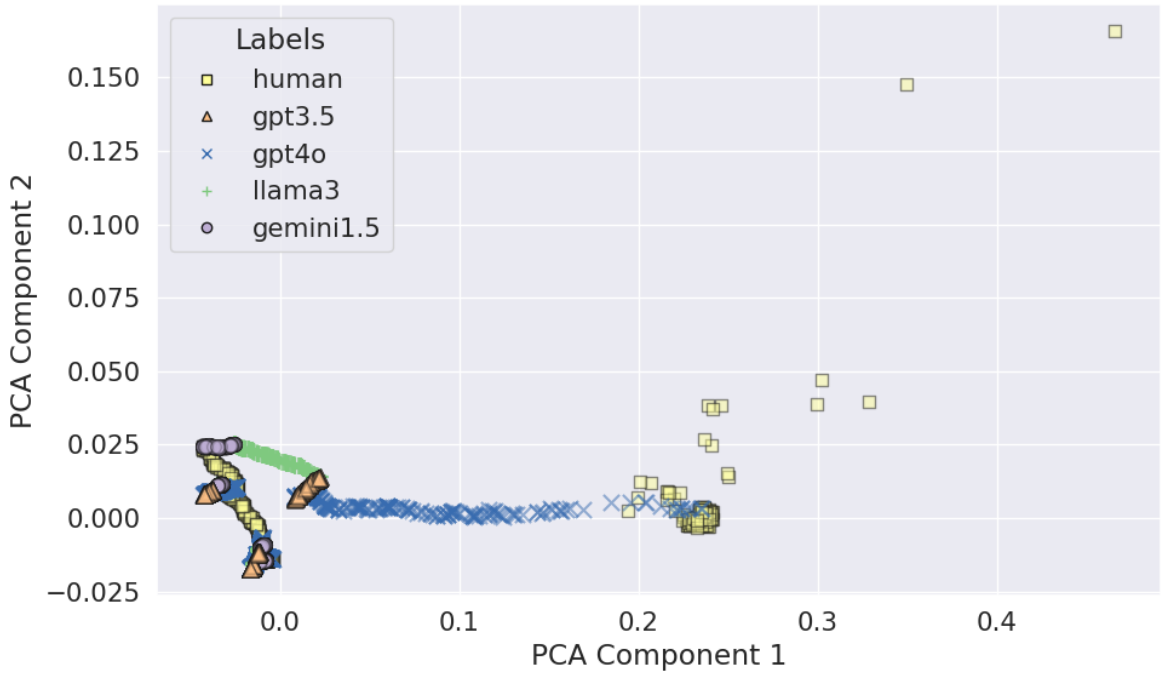


Figure 4: PCA Plot between the first principal component and the second

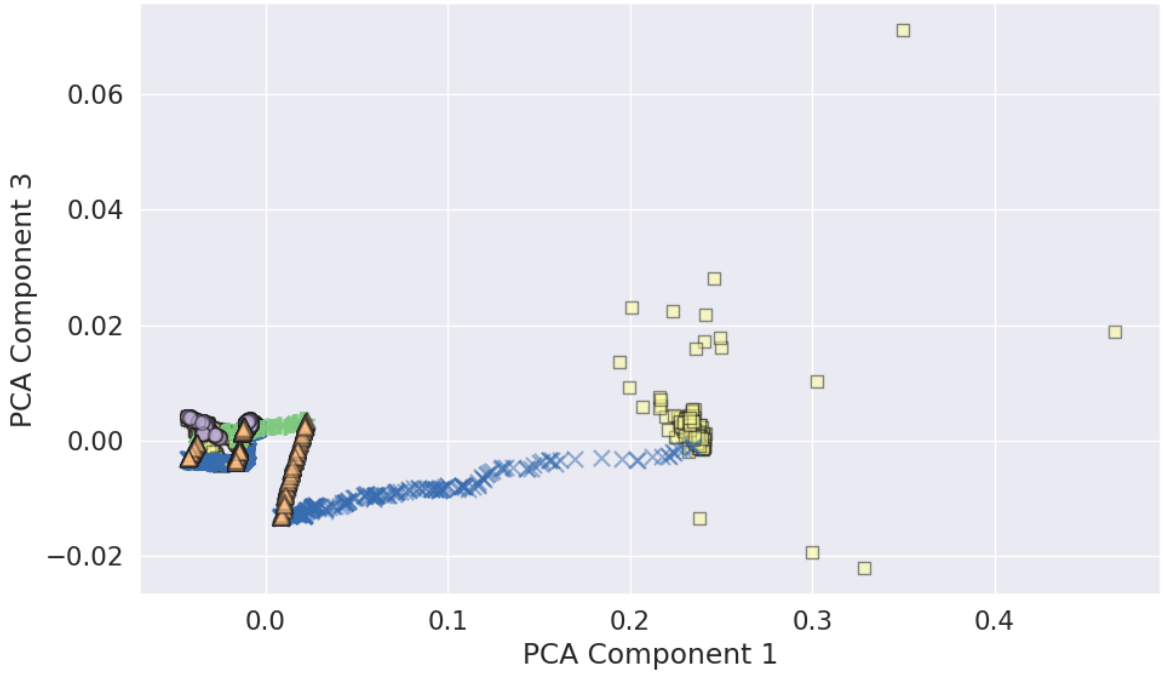
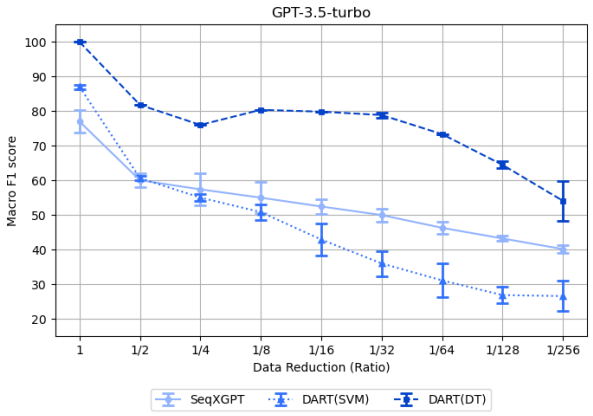
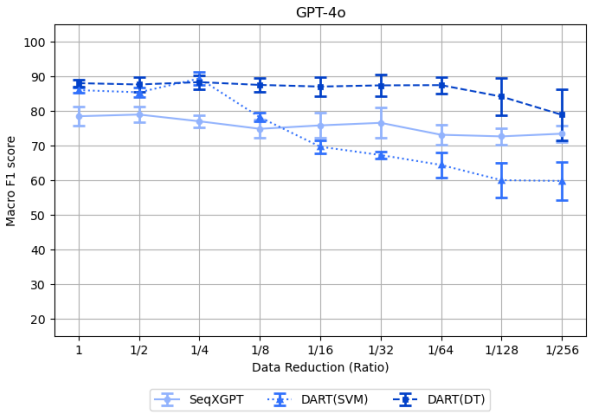


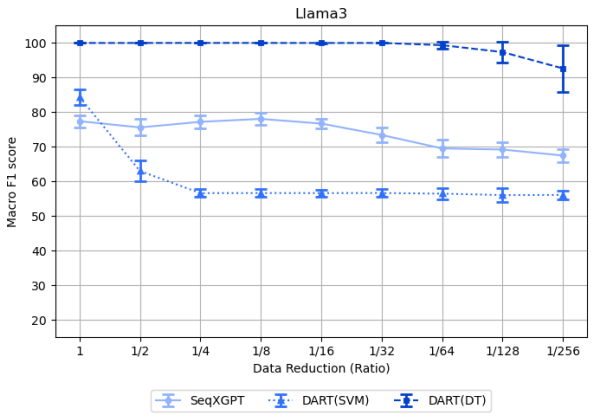
Figure 5: PCA Plot between the first principal component and the third



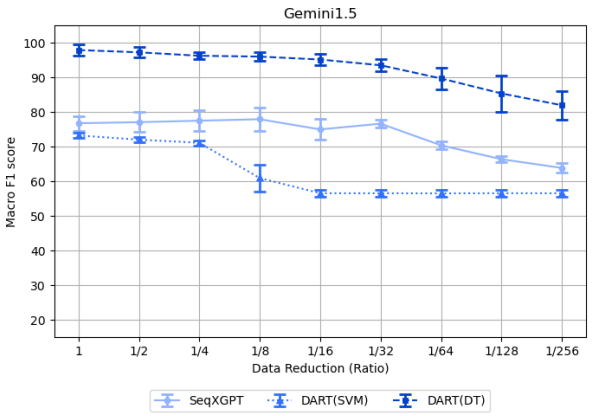
(a) GPT-3.5-turbo



(b) GPT-4o



(c) Llama 3-70b



(d) Gemini-1.5-Flash

Figure 6: Training efficiency on the single-candidate experiment