FOCUS: Forging Originality through Contrastive Use in Self-Plagiarism for Language Models

Kaixin Lan[1*] **Tao Fang**[1*] **Derek F. Wong**[1†] **Yabo Xu**[2]
Lidia S. Chao[1] **Cecilia Guanfang Zhao**[3]
[1]NLP[2]CT Lab, Department of Computer and Information Science, University of Macau
nlp2ct.{Kaixin,taofang}@gmail.com, {derekfw,lidiasc}@um.edu.mo
[2]Guangdong Hengqin DataStory Information Technology Ltd.
arber@datastory.com.cn
[3]Department of English, Faculty of Arts and Humanities, University of Macau
czhao@um.edu.mo

Abstract

Pre-trained Language Models (PLMs) have shown impressive results in various Natural Language Generation (NLG) tasks, such as powering chatbots and generating stories. However, an ethical concern arises due to their potential to produce verbatim copies of paragraphs from their training data. This is problematic as PLMs are trained on corpora constructed by human authors. As such, there is a pressing need for research to promote the generation of original content by these models. In this study, we introduce a unique "self-plagiarism" contrastive decoding strategy, aimed at boosting the originality of text produced by PLMs. Our method entails modifying prompts in LLMs to develop an amateur model and a professional model. Specifically, the amateur model is urged to plagiarize using three plagiarism templates we have designed, while the professional model maintains its standard language model status. This strategy employs prompts to stimulate the model's capacity to identify non-original candidate token combinations and subsequently impose penalties. The application of this strategy is integrated prior to the model's final layer, ensuring smooth integration with most existing PLMs (T5, GPT, LLaMA) without necessitating further adjustments. Implementing our strategy, we observe a significant decline in non-original sequences comprised of more than three words in the academic AASC dataset and the story-based ROCStories dataset.

1 Introduction

Pre-trained language models (PLMs) have gained widespread recognition for their unparalleled performance in numerous downstream natural language processing (NLP) tasks (Clinchant et al., 2019; Li et al., 2022; Fang et al., 2023a,b,c; Zhang et al., 2023; Pang et al., 2024), especially in text generation (Hua and Wang, 2020; Zhang et al., 2020; Guan et al., 2021; Wang et al., 2024). With the emergence of advanced PLMs, there is an intensifying debate over the distinctiveness of texts produced by these models as opposed to those written by humans, a sentiment highlighted by (McCoy et al., 2023). Within this landscape, generative PLMs primarily fall into two architectural paradigms. On the one hand, BART (Lewis et al., 2020), T5 (Raffel et al., 2020), and other such pre-trained language models (Song et al., 2019) serve as quintessential representatives of the encoder-decoder approach. On the other hand, OpenAI's GPT (Generative pre-trained Transformer) series (Radford et al., 2019; Brown et al., 2020; Ouyang et al., 2022) and Meta AI's LLaMA series pre-trained language models (Touvron et al., 2023a,b), leveraging a decoder-only design, has carved out a unique niche for itself. Both these architectural categories of pre-trained language models have exhibited impressive capabilities in natural language understanding (Ebrahimi et al., 2022), and natural language generation (Rothe et al., 2021; Jiao et al., 2023), solidifying their reputation as premier commercial offerings. Because of their efficiency and adaptability (Zhan et al., 2024), these models have been widely adopted across diverse sectors, including writing, and academic research. Nonetheless, the sophistication these models showcase in text generation has ignited concerns regarding academic integrity, prompting many educational establishments to restrict their utilization in scholarly activities.

We believe that the aforementioned issues mainly occur because the content generated by language models often lacks originality, as their outputs heavily rely on their training data. This might lead the models to replicate or mimic the information or patterns they encountered during training. In areas where originality is highly valued, such as academic writing or story generation, the outputs from these models can sometimes be seen as pla-

* Equal Contribution
† Corresponding Author

Findings of the Association for Computational Linguistics ACL 2024, pages 14432–14447
August 11-16, 2024 ©2024 Association for Computational Linguistics

Input:

"Events" are extracted along with their event participants, e.g., "who did what to whom when and where? "

LLaMA2 output:

Events can have different properties such as the type of event, the time and place of the event and the participants. In order to represent the properties of the event, the Event…

Input:

They are of the following form: [MATH] the distribution over all possible state sequences. The parameter vector [MATH] is trained

GPT2 output:

discriminatively to maximize the conditional log likelihood of the training data [MATH]

Figure 1: Samples of academic writing generated by the fine-tuned LLaMA2 and GPT2 PLMs using the AASC dataset. Upon plagiarism analysis with Turnitin, both models' outputs showed significant overlaps (in Red), implying a conspicuous absence of originality. [MATH] indicates a masked math formula.

giaristic. In fact, earlier research has already highlighted the potential risks of intentional or unintentional leakage of sensitive information within the training sets of language models (Zanella-Béguelin et al., 2020; Carlini et al., 2021; Brown et al., 2022). This concern persists even during the fine-tuning phases, as evidenced by Mireshghallah et al. (2022). To further investigate the potential lack of originality in PLMs outputs, we conduct fine-tuning on two prominent generative PLMs, LLaMA2 and GPT2 pre-trained language models, using an academic paper AASC[1] dataset from the NLP domain. Testing their outputs with Turnitin[2] revealed significant instances where the models reproduced segments from the training set, as illustrated in Figure 1.

Currently, a considerable portion of research is dedicated to determining whether the outputs from PLMs display plagiarism or retain their originality (Ferrero et al., 2017; Wahle et al., 2022; Lee et al., 2023; Wu et al., 2024). However, there is a noticeable lack of research focused directly on the innate originality of content generated by these PLMs. To address this void, we introduce a novel generation approach named the "self-plagiarism" (SP) contrastive decoding strategy, aimed at bolstering the inherent originality of text generated by PLMs. This strategy builds upon the principles delineated by Schick et al. (2021) and Chuang et al. (2023), yet diverges from their methodologies. Initially, our approach shifts its focus from mitigating model bias at the token level to accentuating originality at the paragraph level. A distinctive feature of our method lies in the strategic emphasis on the topmost layer, enriched with high-level knowledge. Coupled with the extension of generation length, our methodology encompasses a broader spectrum of content uniqueness. This proves particularly advantageous in domains like storytelling and academic writing, where the nuances of high-level knowledge hold significant importance.

Secondly, we achieve an amateur model and a professional model by adjusting the prompts of the language models. For the amateur model, we innovatively introduce three prompts, which originate from the three most common research categories in plagiarized literature: verbatim plagiarism, paraphrase plagiarism, and idea plagiarism. The purpose of these prompts is to guide the model to replicate the training data according to specific plagiarism standards. On the other hand, the professional model uses conventional prompts to encourage the model to generate normal text. On this basis, we subtract the probability distribution of the last layer of the two models and impose penalties on tokens that show a higher probability in a regulatory factor function, ensuring a balance between following the prompts and maintaining originality. Finally, we use our method on various language models, including T5, GPT-2, LLaMA1, and LLaMA2, on the academic dataset AASC and ROCStories dataset. The results show a significant reduction in non-original sequences of more than three words generated by PLMs.

Our primary contributions are as follows:

- We reconfirm that even during the fine-tuning phase, pre-trained language models still manifest tendencies of plagiarism. This propensity for un-originality is particularly evident in the domain of academic writing.

- We innovatively introduce "self-plagiarism" contrastive decoding strategy by adjusting the prompts of the language models to achieve an amateur model and professional model, and subsequently penalizes the plagiarized tokens. This approach significantly reduces the plagiarism rate and enhances originality.

- We showcase the efficacy of our approach in augmenting the originality of the content produced by the models on the widely-used academic writing AASC dataset and ROCStories

[1] https://github.com/KMCS-NII/AASC
[2] https://www.turnitin.com/

The following text contains exact copies of words or phrases without transformation language:

(1) Verbatim plagiarism

The following text contains synonymous substitution, word reordering, and back translation language:

(2) Paraphrase plagiarism

The following text contains reuse of the core idea by shortening or summarizing the original content language:

(3) Idea plagiarism

Figure 2: Prompts used for Self-plagiarizing.

dataset, providing valuable guidelines for subsequent text generation endeavors.

2 Methodology

2.1 Definition of Language Modeling

Pre-trained autoregressive language model $p_\theta(y|x)$ is usually built based on the Transformer framework and parametrized by θ. The pre-trained model computes h_i as a function of z_i and the past activations in its left context:

$$h_i = PLM_\theta(z_i), \tag{1}$$

where h_i is the last layer which is used to compute the distribution for the next token:

$$p_\theta(z_{i+1}|h_{\leq i}) = Softmax(W_\theta h_i), \tag{2}$$

where W_θ is the pre-trained parameters matrix.

During the finetuning stage, we use the pre-trained parameters θ to initialize the model where p_θ is a trainable language model distribution. The finetuning performs gradient updates on the log-likelihood objective:

$$\arg\max_\theta p(y|x;\theta) = \sum \log p_\theta(z_i|h_{<i}), \tag{3}$$

2.2 Plagiarism and Self-plagiarism Prompts

In the academic domain, plagiarism involves using someone else's work, ideas, or expressions and presenting them as one's own without proper acknowledgment. However, our work focuses on addressing plagiarism in the context of large models—specifically, reducing the straightforward replication of training data by these models. We refer to this as enhancing model's originality.

Inspired by Lee et al. (2023), they introduce three most commonly studied categories in plagiarism literature: Verbatim, Paraphrase, and Idea plagiarism. They evaluate both pre-trained and fine-tuned models' plagiarism tendencies in these categories. Their findings suggest language models do not just mimic training samples but can also rephrase or borrow ideas from original texts. Informed by their insights, we craft Self-plagiarizing prompts, tailor to guide the model in replicating training data based on the designated plagiarism criteria. Figure 2 illustrates our specific prompt templates.

Formally, let the input text be represented as X. Our self-plagiarizing templates encompass text components denoted by P, which can manifest in one or more of the ways listed below:

- Exact copies of words or phrases without transformation.
- Synonymous substitution, word reordering, and back translation.
- Reuse of the core idea by shortening or summarizing the original content.

2.3 Amateur LM and Expert LM

Upon finalizing the design of the plagiarism prompting template, our primary objective is to construct a robustly flawed amateur model, alongside a proficient expert model. In particular, we denote $p(Y|X)$ as the original predictive probability distribution of the input X, and $p(Y|sp(X,P))$ signifies the probability of the subsequent word given input X in self-plagiarizing prompt templates. This self-plagiarizing input urges the language model to manifest plagiarized behavior, laying the foundation for both the amateur and expert models we have developed. The amateur model is established based on $p(Y|X)$, and it emulates self-plagiarizing behavior to formulate a predictive probability distribution $p_{AMA}(Y|sp(X,P))$ for input X. Conversely, we provide default system prompts for the expert model, or no prompt at all for models that do not support system prompts. The expert model draws insights from the original predictive probability distribution $p_{EXP}(Y|X)$ to deliver more precise predictions on input X.

2.4 Contrastive Decoding

Figure 3 displays the detailed framework of our methodology. The created amateur model tends

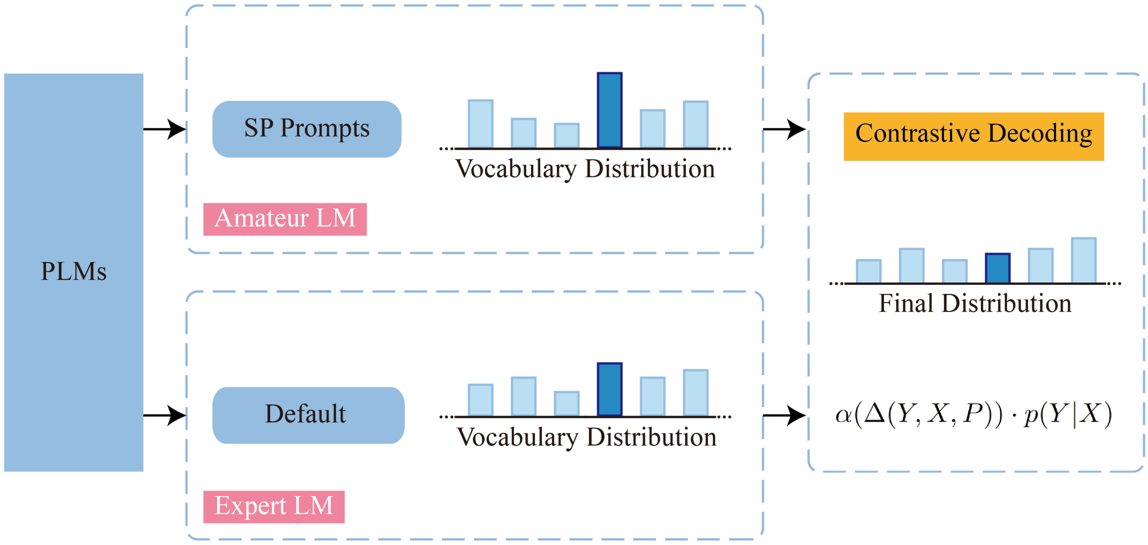

Figure 3: PLMs are prompted to function as both an expert LM and an amateur LM by utilizing SP prompts and default output. The optimized prediction probability is then obtained through contrastive decoding.

to produce plagiarized words or fragments more than the expert model. When both the expert model and the amateur model assign a higher probability score to a "repeating token", the expert model is more likely to also assign high scores to other good tokens with low repetitiveness (Li et al., 2023). However, the amateur plagiarism model does not behave this way, which means that it is more prone to be influenced by plagiarism prompts.

To make the text generated more original, it is necessary to resist the occurrence of plagiarism. Therefore, we propose a contrastive objective function:

$$\Delta(Y,X,P) = p_{EXP}(Y|X) - p_{AMA}(Y|sp(X,P)), \tag{4}$$

In cases of multiple prompts used simultaneously, we keep the largest difference.

$$\Delta(Y,X,P) = \min_{P} \Delta(Y,X,P), \tag{5}$$

A plagiarized sentence is more likely to receive a higher probability from $p_{AMA}(Y|sp(X,P))$ than $p_{EXP}(Y|X)$. Hence, for those exact copies of words or phrases, $\Delta(Y,X,P)$ will be lower than zero. For those original expressions, $\Delta(Y,X,P)$ will be greater than zero.

However, it's not necessary to dismiss all results produced by amateur models. To address this, we follow Schick et al. (2021) to design a regulatory factor, denoted as α, a scale function is used to scale those differences to a number between 0 to 1.

$$\alpha(x) = \begin{cases} 1, & x > 0 \\ e^{\lambda \cdot x}, & otherwise \end{cases} \tag{6}$$

where λ is a hyperparameter, x is contrastive objective. For those original expressions ($\Delta(Y,X,P) > 0$), their probability will be kept the same, while for those exact copies of words or phrases , $e^{\lambda \cdot \Delta(Y,X,P)}$ will be lower than 1.

The result will be used to adjust the original predict the probability distribution $p(Y|X)$. Finally, model will generate output based on $\tilde{p}(Y|X)$:

$$\tilde{p}(Y|X) \propto \alpha(\Delta(Y,X,P)) \cdot p(Y|X). \tag{7}$$

3 Experiments

Considering the vast amount of training data used for pre-trained models and the lack of transparency regarding the specific datasets utilized, it is challenging to detect plagiarize on models that are either not fine-tuned or proprietary. To effectively address the issue, we opt to fine-tune open-source models on publicly available datasets and subsequently assess their plagiarism on these fine-tuned datasets. This strategy allows us to rigorously validate our method and its efficacy in detecting and mitigating plagiarism in PLMs.

3.1 Dataset

We utilize the ROCStories training dataset (Mostafazadeh et al., 2016) and the ACL Anthology

Dataset	Train	Eval	Test
ROCStories	98,161	1572	1,871
AASC	282,332	2812	2,874

Table 1: Summary of the Experimental Datasets

Sentence Corpus (AASC)[3] during the fine-tuning phase of the language models. For the ROCStories dataset, we strictly follow the partition outlined in (Mostafazadeh et al., 2016), dividing it into training, testing, and development sets. The AASC dataset constitutes a curated compilation of text excerpts extracted from scientific papers in the field of natural language processing. Drawn from PDF-format papers published within the ACL Anthology between 2000 and 2018, each paper is segmented into individual sentences, categorized according to their respective originating sections. To construct our training set, we chose sections including Abstract, Introduction, Background, Method, Result, and Discussion. For evaluation and testing, a chosen 1% of sentences were randomly extracted from the corpus. A detailed data statistics can be found in Table 1.

3.2 Model and Training

With due consideration for performance optimization and the judicious allocation of computational resources, our methodological framework revolves around the utilization of LLaMA-7b series pre-trained language models (Touvron et al., 2023a,b), GPT-2 large (Radford et al., 2019) and T5 large (Raffel et al., 2020) as baseline models. Our training process leverages the computational power of four A100 or V100 GPUs. Both GPT-2 and T5 models undergo fine-tuning over a span of five epochs, employing two distinct datasets. Because of limitation of computing resource, LLaMA series are fine-tuned over 3 epochs with Alpaca-LoRA.[4] The foundational pre-trained models and training scripts are sourced from the Huggingface repository.[5] Appendix A presents a detailed account of the hyper-parameters used in the training process.

3.3 Evaluation

We employ three distinct evaluation ways: Generation Originality Test (GOT) (Brooks and Youssef, 2021), Turnitin , and human evaluation. GOT is an n-gram automated test for assessing originality. It constructs an original set by extracting unique n-gram fragments from the training set and subsequently examines whether the fragments generated in the test output are contained within this original set (see Appendix B). We apply GOT to all models on both datasets. To further evaluate if our approach effectively mitigates plagiarism in academic writing and enhances the model's originality, we utilize Turnitin, a widely used academic plagiarism detection software, specifically on the AASC academic dataset. For human evaluation, we enlist the feedback of two volunteers to assess the impact of our method on the coherence (the logical connection and content association between sentences) and fluency (grammar error, naturalness, and writing style) of the model's output. Appendix C presents a detailed definition of coherence and fluency, as well as evaluation template and examples.

4 Results

4.1 Evaluation on GOT

Figures 4 illustrate the comparative analysis of output dissimilarity between the four pivotal models, evaluated using the GOT metric. Appendix D presents illustrative instances of input and output. In terms of overarching trends, a discernible pattern emerges: the resemblance between model-generated texts diminishes in opposite proportion to the length of the identified segments. The occurrence of text displaying similarity, spanning more than seven consecutive words, becomes exceedingly rare. Moreover, the similarity reduction due to the SP contrastive decoding strategy decreases with increasing segment length. All of the four models show a lower similarity rate on AASC than on ROCStories. We think the reason lies on ROCStories has a more common used vocabulary, while the inherent limitations of the SP contrastive decoding strategy inadvertently constrain the range of potential candidates for model predictions, prompting a selection bias towards words that are more prone to detection by the GOT algorithm.

Within the context of the dataset evaluated through ROCStories analysis, LLaMA series show an overall lower similarity, indicates a better originality of large model. GPT-2 large consistently exhibits a reduction of over one percent in similarity across all segment lengths up to five words. In contrast, T5 large deviates from this trajectory,

[3]https://github.com/KMCS-NII/AASC
[4]https://github.com/tloen/alpaca-lora
[5]https://github.com/huggingface/transformers

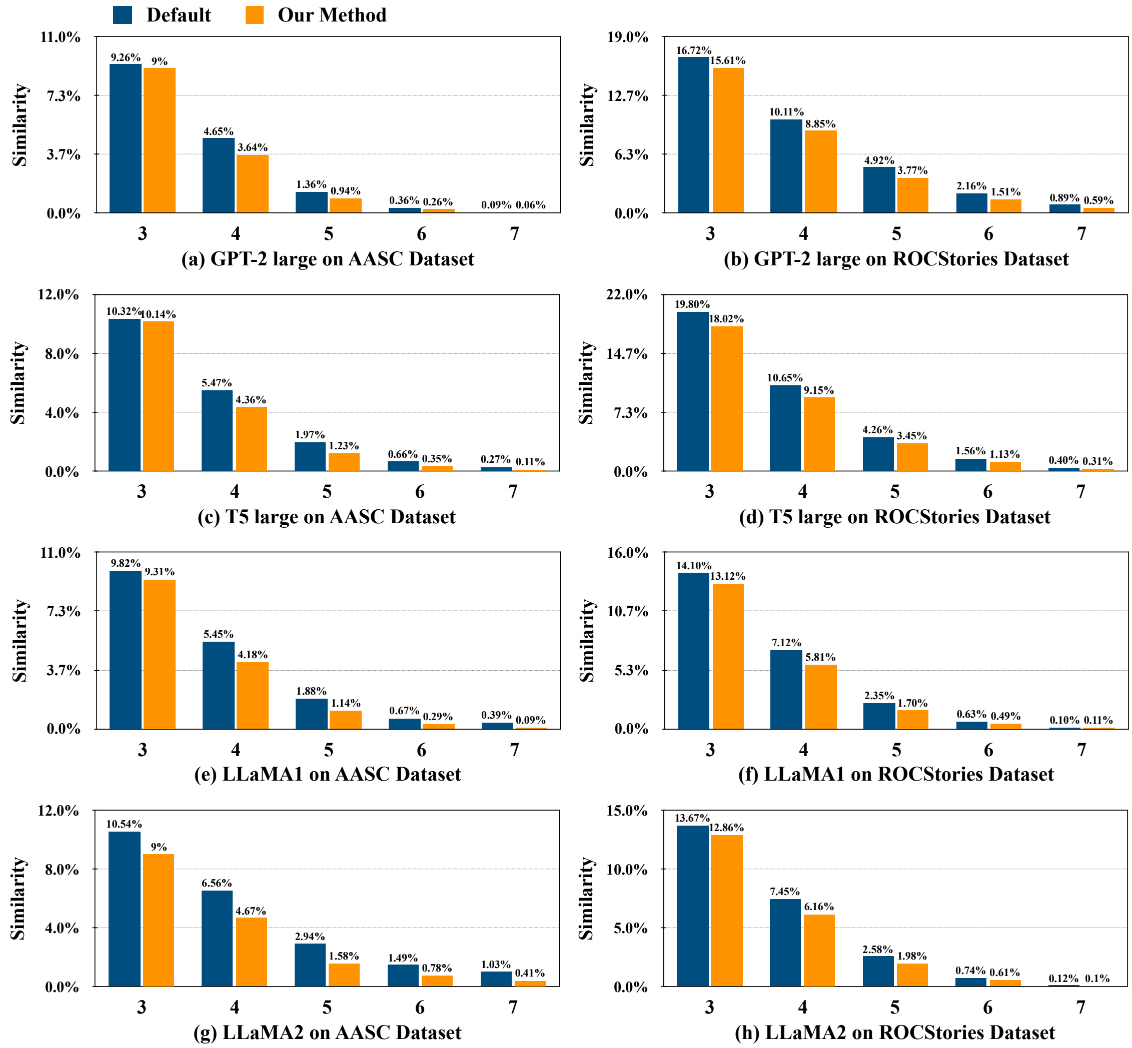

Figure 4: Evaluation results of the fine-tuned GPT-2 large, T5 large, LLaMA1 and LLaMA2 PLMs on the ROCStories and AASC datasets using GOT Metric. Our proposed method exhibits reduced plagiarism across various fragments lengths compared to the default, highlighting its enhanced originality.

exhibiting less similarity decreases when fragment length is higher than five, but maitain a lower similarity rate at the same time, which means T5 takes an advantage on similarity when generating a long sentence, also makes a discount on the performance of SP contrastive decoding strategy.

4.2 Evaluation by Turnitin Check on AASC test set

We collate the output texts generated by LLaMA2, GPT-2 and T5 on the AASC dataset. Subsequently, these generated texts are systematically uploaded onto the Turnitin platform to undergo rigorous originality assessments. Finally, we analyze the outcomes of these Turnitin evaluations. Table 2 showcases the assessment outcomes rendered by Turnitin. For all of the three fine-tuned pre-trained models, our proposed method yields a at least 3% decrease in similarity rates. This empirical validation through Turnitin lends robust support to the practical efficacy of our proposed SP contrastive decoding strategy, confirming its effectiveness within real-world applications.

Model	LLaMA2	GPT2 large	T5 large
Default	12%	5%	6%
Our Method	4%	2%	3%

Table 2: Turnitin Similarity Check on the AASC dataset. Our method consistently achieves a minimum 3% reduction in the similarity of the model's output text.

4.3 Human Evaluation

We average the scores of the two volunteers for the model evaluation. The results, outlined in Table

	Coherence		Fluency	
	default	SPCD	default	SPCD
A_T5	83.5%	88.5%	87.3%	87.2%
A_LLaMA2	86.5%	88.0%	82.0%	83.0%
R_GPT2	76.8%	76.8%	74.5%	75.3%
R_LLaMA2	92.3%	87.5%	87.5%	85.3%

Table 3: Human evaluation of coherence and fluency on LLaMA2, GPT2-Large and T5-Large. The notation "A_model" refers to the fine-tuned model using the AASC dataset, while "R_model" denotes the fine-tuned model using the ROCStories dataset, SPCD means SP contrastive decoding strategy.

3, demonstrate that, in the majority of cases, our method yields comparable results in coherence and fluency. Notably, it even exhibits higher coherence on the AASC dataset. We attribute this to the robustness of pre-trained language models (PLMs), which can generate coherent and fluent text even when certain tokens are penalized. This suggests that our method has minimal negative effects on the coherence and fluency of the model's output, successfully preserving the natural flow and understandability of the generated content.

5 Ablation study

5.1 Impact of Self-plagiarism Prompts

To independently verify the impact of self-plagiarizing prompts, we design ablation experiments using LLaMA2 with both default system prompts and self-plagiarizing prompts, specifically:

- The following text contains exact copies of words or phrases without transformation of language:

We then compare the similarity of the output results. The findings, depicted in Figure 5, indicate that the similarity of the outputs generated with self-plagiarizing prompts is higher across all fragment lengths compared to those generated with default prompts. This demonstrates the effectiveness of the contrastive decoding strategy.

5.2 Impact of Varying Prompt Numbers

To comprehensively comprehend the influence of prompt quantity on the SP contrastive decoding strategy's efficacy, we conduct a meticulous evaluation using GPT-2 large PLM on the ROCStories dataset, modulating the number of prompts as the

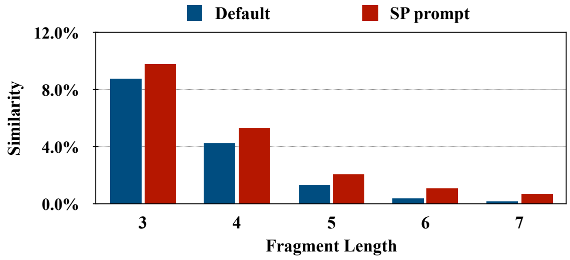

Figure 5: Impact of Self-plagiarism Prompts. When adding only self-plagiarism prompt, the similarity score consistently increase.

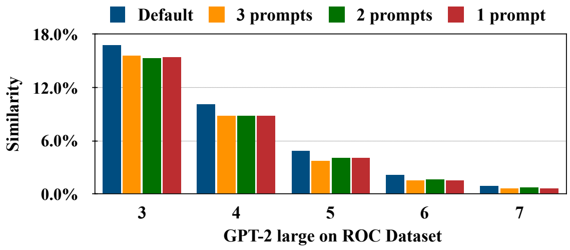

Figure 6: Impact of Varying Prompt Numbers. When adding three prompts, the similarity score reaches its lowest point.

independent variable. As depicted in Figure 6, our findings reveal a consistent trend: the incorporation of the SP contrastive decoding strategy invariably leads to a reduction in the similarity of the generated texts across all test instances. This demonstrates that our approach enhances the originality of content produced by PLMs.

Upon augmenting the prompts to three, a slight augmentation in performance arrives. This enhancement emerges when the fragment length exceeds three words. This empirical observation underscores the pivotal role that sufficiently comprehensive prompts play in adeptly guiding the model's generation process. This nexus between prompt comprehensiveness and algorithmic efficacy underscores that the SP contrastive decoding strategy can achieve heightened performance levels with a more elaborate array of prompts. This augmentation manifests as a reduction in similarity across an array of test cases, reinforcing the integral role of robust and all-encompassing prompts in enhancing the overall efficacy of the SP contrastive decoding strategy.

5.3 Impact of Prompt Template

We evaluate the performance of LLaMA2 on the AASC dataset using various prompts, as depicted in Figure 7. In the "Name Only" template, we streamline the detailed definitions of the three types of

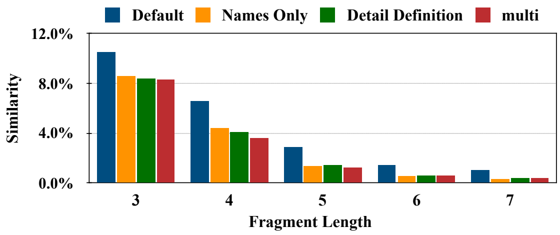

Figure 7: Impact of Prompt Template. A more specific prompt, along with a higher quantity of prompts, contributes to a more significant improvement in performance.

plagiarism to include only their names, while the "Detail Definition" template retains the same format as in the previous experiment. Importantly, the multi-prompt condition, which includes both templates, outperformes other control groups. These findings highlight that a more specific prompt and a higher quantity of prompts contribute to a more substantial improvement in performance.

6 Related Work

6.1 Plagiarism and Memorization in PLMs

Numerous studies have consistently shown that pre-trained language models have a tendency to memorize and plagiarize content from their training data. The study by Brown et al. (2022) underscores the potential risk of intentional or unintentional disclosure of sensitive information from a model's training set. Pioneering research, such as that by Zanella-Béguelin et al. (2020) and Carlini et al. (2021), has revealed the extensive ability of large-scale models to internalize training samples during pre-training, making them especially susceptible to membership inference and data extraction attacks. Importantly, a significant portion of the training datasets used for language model training are culled from the Internet, often without obtaining clear consent from the original content creators Brown et al. (2022). Lee et al. (2023) evidence that PLMs do reproduce content from training samples, encompassing all three classifications of plagiarism. More recently, McCoy et al. (2023) explore the novelty of machine-produced texts and conclude that neural language models have the knack for weaving familiar components into fresh content, rather than merely echoing training samples. Collectively, these studies illuminate the undeniable fact that the outputs of PLMs do draw heavily from their training samples. This sheds light on the deep reliance of pre-trained language models on their training data, underscoring their limitations in terms of innovation and originality.

6.2 Plagiarism Detection for LMs

Plagiarism detection in language models refers to determining whether the model's output replicates content from its training data. To identify if machine-generated texts directly "plagiarize" from its training set, researchers have developed a multitude of detection techniques. For instance, Bensalem et al. (2014) introduce an innovative, language-agnostic plagiarism detection approach, dubbed the n-gram class method, which relies on a novel text representation. Küppers and Conrad (2012) gauge the Dice coefficient for 250-character blocks between passage pairs, while Shrestha and Solorio (2013) employ n-grams to compute the Jaccard similarity. Relying on Convolutional Neural Networks, Agarwal et al. (2018) extract regional information from n-grams and use Recurrent Neural Networks to grasp long-term dependencies. Alzahrani (2015) identify candidate documents by searching for exact duplicated sequences and analyze the similarity of overlapping 8-grams. In the context of generation tasks lacking standard automatic measures, Brooks and Youssef (2021) propose an automated originality testing method. Recently, Lee et al. (2023) embrace a combination of traditional similarity metrics and cutting-edge models, aiming to enhance the efficacy of plagiarism detection. While these studies focus on pinpointing models' plagiaristic behaviors, the question of how to amplify the originality of the generated content has remained unexplored.

6.3 Contrastive Decoding

Li et al. (2023) use an amateur language model (LM) to aid an expert LM in generating coherent text. The work of O'Brien and Lewis (2023) additionally supports the efficacy of contrastive decoding in text generation. In our approach, we integrate their concept into downstream tasks. We use prompts to encourage model plagiarism. This allows us to utilize a single PLM, comparing its predictions when conditioned by opposing prompts, thereby discerning and refining the model's responses. In a related vein, Chuang et al. (2023) propose a contrastive decoding approach, emphasizing the disparity in logits between a higher layer and a lower layer to derive the output probability over the next word. Additionally, Dai et al. (2022) discover the presence of "knowledge neurons" distributed predominantly in the topmost layers of pre-trained BERT models. Building upon these insights, our work adopts a similar idea, but just leveraging knowledge from the topmost layer to penalize plagiarized tokens. This will contribute to the improving of originality of model's generation.

Conclusion

In this study, we emphasize once again the susceptibility of fine-tuned PLMs to display plagiarism tendencies and replicate content from their training sets. This observation applies to both encoder-decoder and decoder-only architectures. These findings highlight the potential risks linked to the utilization of PLMs in sensitive domains like academic writing and storytelling. To tackle the problem of model plagiarism and boost its originality, we introduce an approach involving the introduction of three plagiarism prompts. These prompts guide the model to initially replicate training data in accordance with plagiarism prompts, effectively functioning as an amateur model. Simultaneously, penalties are applied to tokens that show increased probabilities in a designated function, thereby enhancing the overall originality of the generated text. Implementing the proposed approach results in a significant reduction in the generation of non-original sequences, particularly those with over three-word fragments, across both the AASC and ROCStories datasets using PLMs. The alignment of these outcomes offers a robust evaluation of our proposed SP contrastive decoding strategy's ability to ensure the originality of generated text. Furthermore, human evaluation suggests that our method has minimal adverse effects on the coherence and fluency of the output from large pre-trained models. This comprehensive validation highlights the strength and practical utility of our proposed approach in addressing plagiarism risks in NLP tasks.

Limitations

Due to computational limitations, we can not test SP contrastive decoding strategy on larger pre-trained models like GPT-3 (Brown et al., 2020). This will be a focus of our future work. Given the opacity of the training data and the high complexity of the GOT algorithm, we opt not to validate our method on base models without fine-tuning. This is an area we plan to address and improve upon in future work. The primary constraint inherent to our approach lies in its theoretical inability to entirely eradicate plagiarism. Furthermore, given its prompt-based nature, the efficacy of the SP contrastive decoding strategy is contingent upon the model's inherent comprehension capacity. This dependence implies that its performance might be less optimal when applied to models outside the realm of PLMs. Another potential concern with the SP contrastive decoding strategy is its inherent need to prompt the model to generate predictions contrary to the intended outcome. This complicates the algorithm's use for purposes other than counter-plagiarism unless a consistently opposing directive is employed.

Ethics Statement

This research endeavor rigorously adheres to ethical guidelines and principles. We exclusively employ publicly accessible models, datasets, and tools, thereby precluding any involvement or collection of sensitive or private data. All datasets and model parameters are obtained strictly for research purposes from public repositories. Central to this study is the objective to chart a course for effectively guiding the behavior of models, aiming to enhance their capacity to assist users ethically. Our ambition is to foster the ethical advancement of language models, championing their deployment in ways that respect ethical standards and promote societal well-being.

Acknowledgements

This work was supported in part by the Science and Technology Development Fund, Macau SAR (Grant No. FDCT/060/2022/AFJ, the mainland China collaboration project, National Natural Science Foundation of China Grant No. 62261160648), the Science and Technology Development Fund, Macau SAR (Grant No. FDCT/0070/2022/AMJ, the mainland China collaboration project, China Strategic Scientific and Technological Innovation Cooperation Project Grant No. 2022YFE0204900), the Multi-year Research Grant from the University of Macau (Grant No. MYRG-GRG2023-00006-FST-UMDF), and the Tencent AI Lab Rhino-Bird Gift Fund (Grant No. EF2023-00151-FST). This work was performed in part at SICC which is supported by SKL-IOTSC, and HPCC supported by ICTO of the University of Macau. We would like to thank the anonymous reviewers for their insightful comments.

References

Basant Agarwal, Heri Ramampiaro, Helge Langseth, and Massimiliano Ruocco. 2018. A deep network model for paraphrase detection in short text messages. *Information Processing & Management*, 54(6):922–937.

Salha M. Alzahrani. 2015. Arabic plagiarism detection using word correlation in n-grams with k-overlapping approach, working notes for pan-araplagdet at fire 2015. In *Fire*.

Imene Bensalem, Paolo Rosso, and Salim Chikhi. 2014. Intrinsic plagiarism detection using n-gram classes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1459–1464, Doha, Qatar. Association for Computational Linguistics.

Jennifer Brooks and Abdou Youssef. 2021. GOT: Testing for originality in natural language generation. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 68–72, Online. Association for Computational Linguistics.

Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. 2022. What does it mean for a language model to preserve privacy? In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 2280–2292.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.

Stephane Clinchant, Kweon Woo Jung, and Vassilina Nikoulina. 2019. On the use of BERT for neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 108–117, Hong Kong. Association for Computational Linguistics.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Ricardo Ramos, Annette Rios, Ivan Vladimir Meza Ruiz, Gustavo Giménez-Lugo, Elisabeth Mager, Graham Neubig, Alexis Palmer, Rolando Coto-Solano, Thang Vu, and Katharina Kann. 2022. AmericasNLI: Evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6279–6299, Dublin, Ireland. Association for Computational Linguistics.

Tao Fang, Jinpeng Hu, Derek F. Wong, Xiang Wan, Lidia S. Chao, and Tsung-Hui Chang. 2023a. Improving grammatical error correction with multimodal feature integration. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9328–9344, Toronto, Canada. Association for Computational Linguistics.

Tao Fang, Xuebo Liu, Derek F. Wong, Runzhe Zhan, Liang Ding, Lidia S. Chao, Dacheng Tao, and Min Zhang. 2023b. TransGEC: Improving grammatical error correction with translationese. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3614–3633, Toronto, Canada. Association for Computational Linguistics.

Tao Fang, Shu Yang, Kaixin Lan, Derek F. Wong, Jinpeng Hu, Lidia S. Chao, and Yue Zhang. 2023c. Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation. *arXiv preprint arXiv:2304.01746*.

Jérémy Ferrero, Laurent Besacier, Didier Schwab, and Frédéric Agnès. 2017. Deep investigation of cross-language plagiarism detection methods. In *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*, pages 6–15, Vancouver, Canada. Association for Computational Linguistics.

Jian Guan, Xiaoxi Mao, Changjie Fan, Zitao Liu, Wenbiao Ding, and Minlie Huang. 2021. Long text generation by modeling sentence-level and discourse-level coherence. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6379–6393, Online. Association for Computational Linguistics.

Xinyu Hua and Lu Wang. 2020. PAIR: Planning and iterative refinement in pre-trained transformers for long text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 781–793, Online. Association for Computational Linguistics.

Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Xing Wang, and Zhaopeng Tu. 2023. Is chatgpt a good translator? yes with gpt-4 as the engine. *arXiv preprint arXiv:2301.08745*.

Robin Küppers and Stefan Conrad. 2012. A set-based approach to plagiarism detection notebook for pan at clef 2012. In *CLEF(Online Working Notes/Labs/Workshop)*.

Jooyoung Lee, Thai Le, Jinghui Chen, and Dongwon Lee. 2023. Do language models plagiarize? In *Proceedings of the ACM Web Conference 2023*. ACM.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Bei Li, Quan Du, Tao Zhou, Yi Jing, Shuhan Zhou, Xin Zeng, Tong Xiao, JingBo Zhu, Xuebo Liu, and Min Zhang. 2022. ODE transformer: An ordinary differential equation-inspired model for sequence generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8335–8351, Dublin, Ireland. Association for Computational Linguistics.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

R. Thomas McCoy, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. 2023. How much do language models copy from their training data? evaluating linguistic novelty in text generation using RAVEN. *Transactions of the Association for Computational Linguistics*, 11:652–670.

Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David Evans, and Taylor Berg-Kirkpatrick. 2022. An empirical analysis of memorization in fine-tuned autoregressive language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1816–1826, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Sean O'Brien and Mike Lewis. 2023. Contrastive decoding improves reasoning in large language models. *arXiv preprint arXiv:2309.09117*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Jianhui Pang, Fanghua Ye, Derek Fai Wong, Xin He, Wanshun Chen, and Longyue Wang. 2024. Anchor-based large language models. In *Findings of ACL 2024*.

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

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

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in NLP. *Transactions of the Association for Computational Linguistics*, 9:1408–1424.

Prasha Shrestha and Thamar Solorio. 2013. Using a variety of n-grams for the detection of different kinds of plagiarism notebook for pan at clef 2013. In *Conference and Labs of the Evaluation Forum*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jan Philip Wahle, Terry Ruas, Frederic Kirstein, and Bela Gipp. 2022. How large language models are transforming machine-paraphrase plagiarism. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 952–963, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shanshan Wang, Derek F. Wong, Yao Jingming, and Lidia S. Chao. 2024. What is the best way for chatgpt to translate poetry? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Junchao Wu, Runzhe Zhan, Derek F Wong, Shu Yang, Xuebo Liu, Lidia S Chao, and Min Zhang. 2024. Who wrote this? the key to zero-shot llm-generated text detection is gecscore. *arXiv preprint arXiv:2405.04286*.

Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Victor Rühle, Andrew Paverd, Olga Ohrimenko, Boris Köpf, and Marc Brockschmidt. 2020. Analyzing information leakage of updates to natural language models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. ACM.

Runzhe Zhan, Xinyi Yang, Derek F Wong, Lidia S Chao, and Yue Zhang. 2024. Prefix text as a yarn: Eliciting non-english alignment in foundation language model. In *Findings of ACL 2024*.

Yizhe Zhang, Guoyin Wang, Chunyuan Li, Zhe Gan, Chris Brockett, and Bill Dolan. 2020. POINTER: Constrained progressive text generation via insertion-based generative pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8649–8670, Online. Association for Computational Linguistics.

Yue Zhang, Leyang Cui, Deng Cai, Xinting Huang, Tao Fang, and Wei Bi. 2023. Multi-task instruction tuning of llama for specific scenarios: A preliminary study on writing assistance. *arXiv preprint arXiv:2305.13225*.

A Hyper-parameters for training

As shown in table 5, in the fine-tuning process of both LLaMA1 and LLaMA2 using both the AASC and ROCStories datasets, specific hyperparameters were employed. These include a learning rate of 1e-4, a batch size of 4, a LORA-R of 8, a LORA-Alpha of 16, and a LORA-Dropout of 0.05. For the fine-tuning of GPT-2 with the AASC dataset, a learning rate of 5e-4 is adopted. The block size is configured at 128, and a batch size of 16 per device is instituted, further compounded by a gradient accumulation step count of 2. Conversely, for the fine-tuning of GPT-2 using the ROCStories dataset, default learning rate settings are maintained. The block size is established at 60, with a batch size of 128 per device and a gradient accumulation step count of 2. As for the T5 model, the batch size is set to 24 per device. During the fine-tuning phase on the AASC dataset, a learning rate of 0.001 is employed, coupled with a gradient accumulation step count of 4. Correspondingly, for the ROCStories dataset, the learning rate is maintained at 1e-4, and the gradient accumulation step count is retained at 2.

B GOT Algorithm

Table 4 provides the pseudo code of the Generation Originality Test (GOT) algorithm. We utilized this as the basis for implementing the Python script of the GOT algorithm for conducting similarity testing.

GOT Algorithm: Detect similar fragments

Input: Input Sentences X, List of the sentences S, Original set O
Result: Similar fragments R
Initialize R as empty list
foreach sentence **in** X **do**
    Get length of sentence as sl
    // Define a sliding window of length wl
    for wl = 2 to sl **do**
        // Move the window, and extract fragments
        for i = 0 to sl - wl +1 **do**
            Extract fragment from sentence
            if fragment **in** O **then**
                Add fragment to R
return R

Table 4: GOT Algorithm.

C Human Evaluation

C.1 Evaluation Metric

In this section, we provide definitions for coherence and fluency, breaking them down into specific components:

Coherence:

- Content Association: Indicates whether the output sentence is related to the content of the source sentence.
- Logical Coherence: Rates the natural degree of logical connection between the output sentence and the source sentence.

Fluency:

- Grammar Errors: Indicates whether the output sentence contains grammar errors.
- Naturalness and Vividness: Assesses whether the language of the output sentence is natural and vivid, as opposed to being stiff or verbose.
- Writing Style: Examines whether the output sentence aligns with the style of academic papers or narrative storytelling.

C.2 Template of Human Evaluation

Figure 8 illustrates the template utilized for manual evaluation alongside four concrete examples. It is pertinent to note that due to the recruitment of two volunteer assessors who are native Chinese-speaking doctoral candidates, certain rule explanations are presented in Chinese. The test sentences were randomly sampled from model outputs. Additionally, the identities of the three models have been concealed to uphold the integrity and fairness of the evaluation process. This methodology ensures an unbiased assessment of model performance, safeguarding against potential predispositions or biases associated with prior knowledge of specific model identities.

D Example outputs

Table 6 and 7 elucidates a collection of exemplars encompassing input texts and corresponding outputs, both with and without the integration of the SP contrastive decoding strategy, derived from both fine-tuned LLaMA2, GPT-2 large and T5 large pre-trained models across the two distinct datasets (ROCStories and AASC datasets). Evidently, all

Config.	LLaMA1 & LLaMA2	GPT-2 Large		T5 Large	
Dataset	AASC & ROCStories	AASC	ROCStories	AASC	ROCStories
GPU	A100 * 4	V100 * 4	V100 * 4	V100 * 4	V100 * 4
Learning Rate	1e-4	5e-4	1e-5	1e-3	1e-4
Batch Size	4	16	128	24	24
Epochs	3	5	5	5	5
Block Size	-	128	60	-	-
Gradient Accu.	-	2	2	4	2
LORA-R	8	-	-	-	-
LORA-Alpha	16	-	-	-	-
LORA-Dropout	0.05	-	-	-	-

Table 5: Hyper-parameters for training LLaMA1, LLaMA2, GPT-2 Large and T5 Large models. Gradient Accu. means gradient accumulation steps. - means not applicable or default setting.

SRC	Output 1					Output 2				
	Coherence		Fluency			Coherence		Fluency		
	Content Association	Logical Coherence	Grammar Errors	Naturalness and Vividness	Writing Style	Content Association	Logical Coherence	Grammar Errors	Naturalness and Vividness	Writing Style
	0 or 1	1--5	0 or 1	1--5	1--5	0 or 1	1--5	0 or 1	1--5	1--5
Rex had always loved birds. One day his friend recommended they go birding at a local park. Rex and his friend saw finches, sparrows, and other birds. They talked about all the birds they had seen over ice cream later	home and I should go with him. We got in his van and I sat next to him in the front seat. He drove and I listened to music on the radio.					that day. The next week they went to see a different type of bird.				
	0	1	1	4	4	1	4	1	2	3
Naomi and Andy were riding their bikes in downtown SF. Naomi heard a loud crash and turned around to look at Andy. Andy had swerved away from an oncoming vehicle and into a trolley. On impact Andy was flung over his handlebars and landed on his wrist	and head. The next day Naomi and Andy went to the hospital.					, fracturing the scaphoid. Andy was in a lot of pain and couldn't even lift his arm.				
	1	3	1	3	3	1	5	1	5	5
It was a very hot day. But Ivan wanted to eat chicken. So he baked some chicken in the oven. Everyone else in the house complained that it was too hot	to cook. But he insisted. He wanted chicken so badly.					to turn the oven on. Ivan said, "It doesn't matter. I am going to do this, so just leave me alone.				
	1	5	1	4	4	1	1	1	3	3
Kelly was playing her new Mario game. She had been playing it for weeks. She was playing for so long without beating the level. Finally she beat the last level	. She felt so happy. Her mother was very proud.					of her game! She was so happy. She was going to go to her friend's house and show her. Kelly ran down the stairs to tell her mother.				
	1	4	1	4	4	1	2	1	2	2

Figure 8: Template and examples of human evaluation.

instances of SP contrastive decoding strategy generated outputs aptly maintain logical coherence and textual fluency while effectively circumventing plagiarism. This substantiates the inherent compatibility of the SP contrastive decoding strategy with models underpinning disparate architectures, as well as its efficacy within divergent text generation domains. This robustness further underscores the algorithm's commendable generalization potential. However, it is imperative to acknowledge a potential drawback: the proposed SP contrastive decoding strategy's direct influence on altering the content and conceptual structure of the output text. Consequently, the algorithm may exhibit suboptimal performance within certain generation contexts where the content or conceptual direction is explicitly specified or mandated.

LLaMA2	Input	I was sick of being unemployed. I decided it was time to get a job. I scoured listings on the internet and sent out resumes. Finally, a company asked me to come in for an interview.
	Default Output	I was so happy. When I got there, the receptionist gave me a clipboard with a bunch of paperwork on it and said I had to fill it out
	SPCD Output	I was thrilled. I prepared for the meeting. I made sure to have my hair and clothes looking great. I had to look my best to get the job.
GPT-2	Input	Thomas lost his job three weeks ago. He had no idea how he was going to pay his rent. His friend told him about a website that paid five dollars per blog. Thomas didn't think he could raise five hundred dollars there.
	Default Output	He asked his friends what they thought would happen. They told him that they were going to hire him.
	SPCD Output	He got five hundred dollars together and started the website. He now has a website and is paid to write about things he knows!
T5	Input	Tim was fishing on the bank. He felt a bit on his line. He yanked up real hard. The fish on his line flew out of the water.
	Default Output	Tim let go of his hook. He was happy he was finally able to reel in his fish.
	SPCD Output	He yanked it again, this time a lot harder. The line snapped in half.

Table 6: Example outputs from fine-tuned LLaMA2, GPT-2 large and T5 large PLMs on ROCStories. Text segments in red indicate plagiarism fragments identified by GOT metric. Results suggest that our proposed SP contrastive decoding strategy enhances the originality of text generated by PLMs, effectively mitigating plagiarism.

LLaMA2	Input	This paper deals with predictions which extend to the next several words in the text.
	Default Output	The model is based on an artificial neural network with a hidden layer of sigmoidal units. The network has two outputs, one of which predicts the next character and...
	SPCD Output	Predictive text is the technology behind the texting feature in many cellphones and other smartphones.
GPT-2	Input	If the classified user tweets achieved an accuracy above a threshold (0.7), the user model was applied on the authored
	Default Output	users in the test set with equal weight. To assess the generalization capability of our method we performed additional experiments in which we used a single group
	SPCD Output	user tweets. The user tweets classified with greater accuracy were then sent to the corresponding authored user tweets
T5	Input	First we computed baseline scores using the approaches described in Section 3.2. All resulting scores are listed in Table 2.
	Default Output	We also included baselines computed using the same approach as in the English subtask. We also included the results obtained when using the baselines
	SPCD Output	The first row of Table 2 shows the baseline scores obtained using the naive approach (Naive).

Table 7: Example outputs from fine-tuned LLaMA2, GPT-2 large and T5 large PLMs on AASC. Text segments in red indicate plagiarism fragments identified by GOT metric. Results suggest that our proposed SP contrastive decoding strategy enhances the originality of text generated by PLMs, effectively mitigating plagiarism.