CR-GCL: Comprehensive Robust Graph Contrastive Learning for Machine-Generated Text Detection

Anonymous ACL submission

Abstract

This document is a supplement to the general instructions for *ACL authors. It contains instructions for using the LaTeX style files for ACL conferences. The document itself conforms to its own specifications, and is therefore an example of what your manuscript should look like. These instructions should be used both for papers submitted for review and for final versions of accepted papers.

1 Introduction

These instructions are for authors submitting papers to *ACL conferences using LaTeX. They are not self-contained. All authors must follow the general instructions for *ACL proceedings,[1] and this document contains additional instructions for the LaTeX style files.

The templates include the LaTeX source of this document (`acl_latex.tex`), the LaTeX style file used to format it (`acl.sty`), an ACL bibliography style (`acl_natbib.bst`), an example bibliography (`custom.bib`), and the bibliography for the ACL Anthology (`anthology.bib`).

2 Related Works

2.1 AI-Generated Content Detection

With the continuous improvement of large language models(LLMs) in natural language generation capabilities and the huge success of commercialization, the hidden dangers of misuse and abuse of LLMs have also increased,ranging from academic fraud(Perkins et al., 2023) to the spread of fake news(Hanley and Durumeric, 2024).As a result, AI text detection has stepped forward and become an indispensable gatekeeper for the correct development of LLMs. The existing AI text detectors are mainly divided into statistical based(Mitchell et al., 2023),watermark and entropy based(Kirchenbauer et al., 2024),classifier based(Guo et al., 2023),and retrieval based(Krishna et al., 2023),yet researchers have found that their robustness still have room for increase when facing text disturbances of different granularities(Zhou et al., 2024b),which implies the potential improvement of the existing detectors.

2.2 Adversarial Learning

In fact, many researchers have conducted extensive adversarial attacks on AI text detectors, which not only proves the effectiveness of adversarial attacks but also shows that adversarial learning is a possible direction to improve the robustness of AI text detectors.Sadasivan et al. (2024)designed a paraphraser to rewrite LLM's generated text, successfully fooling the detector and escaping the detection. Furthermore,Kirchenbauer et al. (2024) set adversarial attack by mixing human texts with machine-generated texts, hoping to verfify the anti-disturbance ability of watermark detectors in specific scenarios. On the basis of adversarial attacks,Hu et al. (2023) break the ice by introducing the GAN framework to train a text detector and enhancing the robustness of the detector by the design of a paraphraser.Zhou et al. (2024a) go a step deeper by introducing dynamic iteration based on adversarial attacks with word level substitutions.We furtherly improve the granularity of adversarial attacks by introducing sentence and document level substitutions to make the detector more robust against various attacks.

2.3 Contrastive Learning

Cheng et al. (2023a)'s work shows that contrastive learning has excellent performance in the field of natural language.(Cheng et al., 2023b) also propose a two-layer asymmetric contrastive learning framework to handle noise and improve models' robustness. Liu et al. (2023)'s success in improving model accuracy and robustness proves that con-

[1] http://acl-org.github.io/ACLPUB/formatting.html

trastive learning can effectively learn data representation and improve the generalization ability of detector models.Since the paraphrasing attack strategy is almost the most effective means to resist,with Sadasivan et al. (2024) saying that paraphrasing attack can "can break a whole range of detectors",we innovatively introduce paraphrased text into contrastive learning,the authorship triple construction and multi-faceted comparative learning successfully improve the robustness of the detector in the face of rewriting attacks.

3 Proposed Method

In this section, we will first explain the motivation for adopting graph neural networks, along with the corresponding sample pair design and contrastive learning strategy. We will then provide a detailed analysis of the proposed CR-GCL method.

3.1 Motivation

Detection of Machine-Generated Texts (MGT) has become a prominent research area. However, existing methods largely fail to adequately consider the diversity of adversarial attacks, or only address single attack types. Zhou(Zhou et al., 2024b) summarized twelve potential attack methods and tested them on currently widely recognized MGT detection techniques, finding that most methods struggle to withstand these attacks. Therefore, there is an urgent need to develop robust detection methods capable of comprehensively defending against various MGT attacks. After undergoing multiple attack types, the sentence structures and phrases within texts undergo significant alterations, rendering watermark-based and statistical-based methods notably ineffective. Consequently, classifier-based detectors have emerged as the optimal choice. Graph Neural Networks (GNNs) can effectively capture coherence representations within texts, and there are significant differences in the coherence representations between human-written texts and machine-generated texts. These differences remain difficult to completely eliminate even after various attacks or rewrites. Therefore, we employ Graph Neural Networks as the baseline discrimination method. To effectively address multiple attack types, we adopt Multi-Faceted Contrastive Learning (MFCL), designing two distinct contrastive learning strategies to achieve targeted contrastive learning.

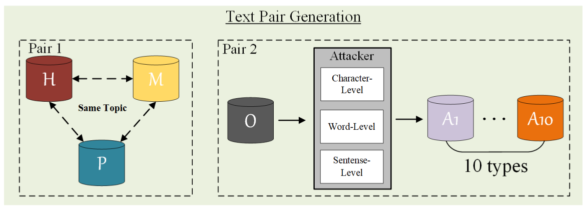


Figure 1: The text pairs we need

3.2 Text Pair Generation

As shown in Figure 1, to fulfill the requirements of comprehensive and robust MGT detection, we have designed two types of text triplets for subsequent contrastive learning.

The first type consists of triplet pairs, each containing three texts on the same topic: one written by humans (T_h), one generated by machines (T_m), and one machine-generated text that has been modified by a paraphrasing tool (T_p). The generation process of these triplets is as follows: initially, a human-written text T_h is obtained. Then, either a summary of its content or its original title is provided to a target model to generate the machine-generated text T_m. Subsequently, a paraphrasing tool is applied to T_m to produce the paraphrased text T_p. These three texts collectively form an authorship-based sample triplet, denoted as (T_h, T_m, T_p). The second type consists of the original text and its corresponding versions that have undergone various attack methods from the Attacker layer. Each original text is paired with 10 different versions, each altered by a distinct attack method.

The design of such sample pairs aims to serve as foundational data preparation for the subsequent Multi-Faceted Contrastive Learning (MFCL) framework.

3.3 Data Preprocessing

In this part, we will elucidate the necessity of preprocessing input data and the methods to implement such preprocessing.

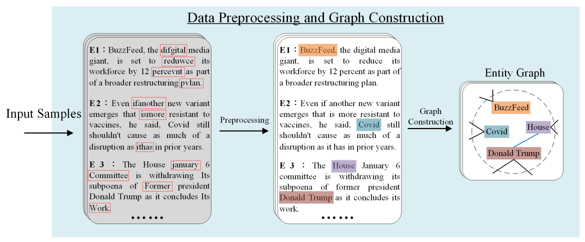
According to Zhou's(Zhou et al., 2024b) summary of twelve attack methods, six of them involve simple and easily detectable attack techniques (e.g., spelling errors, keyboard typos, word merging). These attack methods significantly impact text recognition but can be easily mitigated through straightforward measures. Therefore, we preprocess these texts before inputting them into the model to rectify these easily identifiable attacks.

To prevent inadvertent modifications of important proper nouns during the spelling correction process, we have designed and implemented a proper noun protection mechanism. The acquisition of the proper noun set is achieved through the construction of a proper noun dictionary. This dictionary is created by traversing the sentences in the original dataset prior to training and extracting words that begin with capital letters. After establishing the proper noun protection, we employ the TextBlob library for spelling correction. TextBlob utilizes probabilistic language models to automatically identify and rectify spelling errors based on contextual information. Specifically, TextBlob's correct() method calculates the conditional probability of candidate words given the context and selects the most probable correct spelling. This process can be mathematically represented as:

$$\hat{w} = \arg\max_{w'} P(w'|C) \tag{1}$$

where $\hat{w}$ is the corrected word, w' represents candidate words, and $P(w'|C)$ is the conditional probability of the candidate word w' given the context C.

This method specifically enhances robustness against various attack types, particularly targeting simple attacks such as named entity merging and spelling errors.



E1 : BuzzFeed, the difgital media giant, is set to reduwce its workforce by 12 percevnt as part of a broader restructuring pvlan.

E2 : Even ifanother new variant emerges that ismore resistant to vaccines, he said, Covid still shouldn't cause as much of a disruption as ithas in prior years.

E 3 : The House january 6 Committee is withdrawing Its subpoena of Former president Donald Trump as it concludes Its Work.

......

E1 : BuzzFeed, the digital media giant, is set to reduce its workforce by 12 percent as part of a broader restructuring plan.

E2 : Even if another new variant emerges that is more resistant to vaccines, he said, Covid still shouldn't cause as much of a disruption as it has in prior years.

E 3 : The House January 6 committee is withdrawing its subpoena of former president Donald Trump as it concludes its work.

......

Figure 2: The data preprocessing and graph construction, E1 to E3 are examples of texts that have undergone different types of attacks, with the red boxes highlighting the parts that have been modified as a result of the attacks.

3.3.1 Coherence Graph Construct

According to Centering Theory(Grosz and Sidner, 1986), text coherence can be modeled through the interactions of sentences around central entities. To better reflect the structure of the text and avoid semantic overlap, we propose constructing an undirected graph with entities as nodes. Our approach improves upon existing methods by providing a deeper understanding of the coherence between entities. Specifically, we first utilize the ELMo-based NER model TagLM(Peters et al., 2017), with the help of the AllenNLP NER toolkit, to extract entities from the document. Entities in each sentence are treated as nodes in the graph, with edges categorized into intra-sentence connections ("inner") and inter-sentence connections ("inter"). Intra-sentence edges form a linear structure by sequentially connecting keyword nodes within the same sentence, while inter-sentence edges are established based on the contextual similarity scores between entities. The calculation of contextual similarity relies on the text window mechanism we proposed. Let C_i represent the context window for node v_i, which contains a subset of neighboring nodes based on both syntactic proximity and semantic similarity. The coherence between two entities v_i and v_j can then be quantified by a function Coherence(v_i, v_j) that depends on their contextual overlap:

$$\text{Coherence}(v_i, v_j) = f(v_i, v_j) \tag{2}$$

where $f(v_i, v_j)$ is a function that combines both the syntactic similarity and the semantic relationship between entities v_i and v_j. This function can be further defined as:

$$f(v_i, v_j) = \text{sim}(C_i, C_j) + \alpha \cdot \text{contextual}(v_i, v_j) \tag{3}$$

where α is a weight factor that controls the relative importance of the syntactic and semantic components.

Additionally, for a more refined measurement, we introduce a weighted adjacency matrix where weights are assigned to edges based on the coherence measure:

$$A_{ij} = \begin{cases} \text{Coherence}(v_i, v_j), & v_i \text{related to} v_j \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Through this approach, we can more accurately capture the coherence between entities across sentences. This enables us to better model the relationships between entities and their contextual dependencies in a more granular manner.

3.4 Coherence-Based Detect

3.4.1 Model Overview

The general training process is illustrated in Figure 3. Each data entry in the dataset contains its corresponding ID, the attack method applied to the data (with "original" indicating no attack), the original text of the data, the extracted keyword entities, and the weighted coherence graph. During training, the text and entity relationship graph from the paired

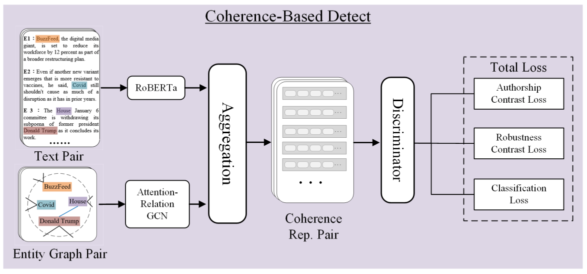


Figure 3: Overview of our Coherence-Based Detection Method

samples are processed separately by RoBERTa and Relation-Aware GCN. The resulting coherence representations are aggregated and used for discrimination. Finally, the model is optimized through backpropagation under the evaluation of three types of loss functions.

3.4.2 Attention-Relation GCN

To enhance the model's ability to analyze text coherence, we introduce the Graph Attention Layer (GATLayer) into the GCN. The GATLayer facilitates this by allowing the model to assign varying degrees of importance to different edges based on their relevance to the overall coherence of the text. By dynamically weighting the connections between nodes, the GATLayer ensures that the model emphasizes relationships that contribute significantly to maintaining a coherent narrative, while deemphasizing less relevant or redundant connections. This selective attention mechanism enhances the model's ability to capture the nuanced dependencies and logical structures inherent in coherent text. Consequently, the GATLayer enables the model to discern the hierarchical and relational dynamics between entities, leading to more accurate and effective coherence assessments. Mathematically, given a node representation matrix $H \in \mathbb{R}^{N\times d}$ and an adjacency matrix $\mathbf{A}$, the attention mechanism is formulated as follows:

$$\mathbf{H}' = \text{softmax}\left(\text{ReLU}(\mathbf{HW}) \odot \mathbf{A}\right)\mathbf{W} + \mathbf{b} \quad (5)$$

where $\mathbf{W} \in \mathbb{R}^{d\times d}$ is the learnable weight matrix, $\mathbf{b} \in \mathbb{R}^{d}$ is the bias term, $\odot$ denotes element-wise multiplication.

Our relational graph extends the standard GCN by incorporating multiple relation-specific GAT layers. For each relation type $r \in R$, a distinct GATLayer is instantiated, enabling the model to learn unique attention weights for different types of edges. The node representations are then aggregated across all relations to form a comprehensive representation:

$$\mathbf{H}^{(l+1)} = \sum_{r\in R} \text{GATLayer}_r\left(\mathbf{H}^{(l)}, \mathbf{A}_r\right) \quad (6)$$

where $\mathbf{H}^{(l)}$ is the node representation at layer l, $\mathbf{A}_r$ is the adjacency matrix corresponding to relation r.

To stabilize training and ensure proper scaling of node features, the model employs a normalized Laplacian matrix $\tilde{\mathbf{L}}$, defined as:

$$\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \quad (7)$$

where $\mathbf{D}$ is the degree matrix. The normalized Laplacian facilitates balanced information propagation across the graph, preventing certain nodes from dominating the information flow due to high degree.

This relational graph method enables the model to effectively disentangle and integrate information from multiple types of relations, enhancing its ability to model complex graph structures.

3.4.3 Sentence Representation

Afterward, we aggregate updated node representation from the last layer of Attention-Relation GCN into sentence-level representation to prepare for concatenation with sequence representation from RoBERTa. .

3.4.4 Sequential Modeling and Aggregation

We adopt the Transformer module to capture long-range dependencies and rich contextual information within the sequence. The Transformer's self-attention mechanism allows it to dynamically focus on different parts of the input sequence, thereby enhancing the model's understanding of text coherence. After Transformer encoding, the model generates the final node representations through weighted average pooling (Mean Pooling):

$$\mathbf{H}_{\text{final}} = \frac{1}{|\mathcal{D}|}\sum_{i\in\mathcal{D}} \mathbf{H}_i^{\text{RoBERTa}} \oplus \mathbf{H}_i^{\text{Transformer}} \quad (8)$$

where $\mathcal{D}$ is the set of document positions, and $\oplus$ denotes concatenation. This aggregation ensures that both graph-based relational information and sequence-based contextual information are effectively integrated.

3.4.5 Multi-Faceted Contrastive Learning

As shown in Figure 4, we adopt Multi-Faceted Contrastive Learning to achieve Comprehensive Robust Machine-Generated Text Detection, namely

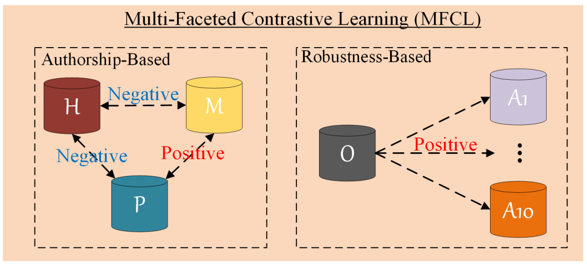


Figure 4: Multi-Faceted Contrastive Learning

Authorship-Based Contrast and Robustness-Based Contrast.

In Authorship-Based Contrast, the first set of text pairs mentioned earlier is used for positive and negative sample comparison. The relationship between positive and negative samples is as follows: T_h is a negative sample with both T_m and T_p, while T_p and T_m are positive samples to each other. When the input sample is T_h, we select another T_h with the same label as a positive sample, while one of the T_m and T_p samples will serve as a negative sample. When the input is T_m, the corresponding T_h and T_p are selected as negative and positive samples, respectively. In Robustness-Based Contrast, second set of text pairs mentioned earlier is used as positive samples. Specifically, one randomly selected instance from the 10 types of attacked original texts is chosen as a positive sample to participate in the training.

3.4.6 Loss Function

Based on the previously mentioned design of contrastive learning, our contrastive loss function is designed as follows:

$$\mathcal{L}_{MFCL} = \lambda_A \mathcal{L}_A + \lambda_R \mathcal{L}_R \tag{9}$$

The designs of $\mathcal{L}_A$ and $\mathcal{L}_R$ are as follows:

$$\mathcal{L}_A = \frac{d(T,T^+)}{\tau} - \frac{d(T,T^-)}{\tau} + \beta \tag{10}$$

$$\mathcal{L}_R = \frac{d(T,T^+)}{\tau} + \beta \tag{11}$$

where, β represents a constant offset, T^+ and T^- represent positive samples and negative samples, respectively. Apart from instance-level learning mechanism, a linear classifier combined with cross-entropy loss $\mathcal{L}_{CE}$ is employed to provide the model with class- level separation ability. L_{CE} is calculated by

$$\mathcal{L}_{\text{CE}} = \frac{1}{N}\sum_{i=1}^{N} -[y_i \log(p_i) + (1-y_i)\log(1-p_i)] \tag{12}$$

where p_i is the prediction probability distribution of i-th sample. The final loss L_{total} is a weighted average of $\mathcal{L}_{MFCL}$ and $\mathcal{L}_{CE}$ as:

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{MFCL} + (1-\alpha)\mathcal{L}_{CE} \tag{13}$$

where the hyperparameter α adjusts the relative balance between instance compactness and class separability.

References

Xuxin Cheng, Bowen Cao, Qichen Ye, Zhihong Zhu, Hongxiang Li, and Yuexian Zou. 2023a. ML-LMCL: Mutual learning and large-margin contrastive learning for improving ASR robustness in spoken language understanding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6492–6505, Toronto, Canada. Association for Computational Linguistics.

Xuxin Cheng, Zhihong Zhu, Yaowei Li, Hongxiang Li, and Yuexian Zou. 2023b. Das-cl: Towards multimodal machine translation via dual-level asymmetric contrastive learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 337–347, New York, NY, USA. Association for Computing Machinery.

Barbara J Grosz and Candace L Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational linguistics*, 12(3):175–204.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *Preprint*, arXiv:2301.07597.

H. W. A. Hanley and Z. Durumeric. 2024. Machine-made media: Monitoring the mobilization of machine-generated articles on misinformation and mainstream news websites. *Proceedings of the International AAAI Conference on Web and Social Media*, 18(1):542–556.

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*, volume 36, pages 15077–15095. Curran Associates, Inc.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. 2024. On the reliability of watermarks for large language models. In *The Twelfth International Conference on Learning Representations*.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Preprint*, arXiv:2303.13408.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. 2023. CoCo: Coherence-enhanced machine-generated text detection under low resource with contrastive learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16167–16188, Singapore. Association for Computational Linguistics.

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.

Mike Perkins, Jasper Roe, Darius Postma, James McGaughran, and Don Hickerson. 2023. Detection of gpt-4 generated text in higher education: Combining academic judgement and software to identify generative ai tool misuse. *Journal of Academic Ethics*, 22(1).

Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108*.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2024. Can ai-generated text be reliably detected? *Preprint*, arXiv:2303.11156.

Ying Zhou, Ben He, and Le Sun. 2024a. Humanizing machine-generated content: Evading AI-text detection through adversarial attack. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8427–8437, Torino, Italia. ELRA and ICCL.

Ying Zhou, Ben He, and Le Sun. 2024b. Navigating the shadows: Unveiling effective disturbances for modern ai content detectors. *arXiv preprint arXiv:2406.08922*.

A Example Appendix

This is an appendix.