Werkzeug at SemEval-2024 Task 8: LLM-Generated Text Detection via Gated Mixture-of-Experts Fine-Tuning

Youlin Wu*, **Kaichun Wang***, **Kai Ma, Liang Yang, Hongfei Lin**†
School of Computer Science and Technology, Dalian University of Technology, China
{wuyoulin, wangkc, ostmbh}@mail.dlut.edu.cn
{liang, hflin}@dlut.edu.cn

Abstract

Recent advancements in Large Language Models (LLMs) have propelled text generation to unprecedented heights, approaching human-level quality. However, it poses a new challenge to distinguish LLM-generated text from human-written text. Presently, most methods address this issue through classification, achieved by fine-tuning on small language models. Unfortunately, small language models suffer from anisotropy issue, where encoded text embeddings become difficult to differentiate in the latent space. Moreover, LLMs possess the ability to alter language styles with versatility, further complicating the classification task. To tackle these challenges, we propose **G**ated **M**ixture-**o**f-**E**xperts **F**ine-tuning (GMoEF) to detect LLM-generated text. GMoEF leverages parametric whitening to normalize text embeddings, thereby mitigating the anisotropy problem. Additionally, GMoEF employs the mixture-of-experts framework equipped with gating router to capture features of LLM-generated text from multiple perspectives. Our GMoEF achieved an impressive ranking of #8 out of 70 teams. The source code is available on https://gitlab.com/werkzeug1/gmoef.

1 Introduction

The advancements in Large Language Models (LLMs) have made generating human-level text more accessible and cost-effective than ever before. These advancements, coupled with techniques such as chain-of-thought (Wei et al., 2022) and instruction tuning (Zhang et al., 2023), have enabled LLMs in producing high-quality text on various topics. However, in the real world, using LLM-generate text is not always acceptable. Thus, there is an urgent need for an easy yet reliable way to detect LLM-generated text.

The SemEval-2024 task 8 (Wang et al., 2024) aims to find methods that can detect machine-generated text. In this work, we followed the most common black-box detection paradigm, which regards such problem as a classification task. We argue that current methods all suffer from the following issues: (1) Anisotropy of the text embeddings (Li et al., 2020; Jiang et al., 2022; Gao et al., 2021). Using small pretrained language models (PLMs) to encode text is the very first step for all classification models, however, PLM may suffer from anisotropy issue, which makes text embeddings clustering in a small cone in the latent space, and compromise the classification performance. (2) Language style of LLM-generated text is dynamic. As aforementioned, LLM can generate text that accommodates various topics and contexts; different LLM may have different optimization targets during pre-training *w.r.t.* text generation. In other words, finding a regular pattern for LLM-generated text is difficult.

To this end, we propose **G**ated **M**ixture-**o**f-**E**xperts **F**ine-tuning (GMoEF) to tackle these problems. GMoEF first uses the PLM to encode the text, then employs parametric whitening transformation to normalize the embedding distribution, in order to mitigate the anisotropy issue; furthermore GMoEF adopts Mixture-of-Experts equipped with gating router to capture features of LLM-generated text from multiple perspectives. Our GMoEF achieved an impressive ranking of #8 out of 70 participating teams on subtask B.

2 Related Work

Typically, LLM-generated text detection is regarded as a classification task aimed at distinguishing between LLM-generated text and human-written text (Jawahar et al., 2020). With the advancement of LLMs, their text generation capabilities have reached a level comparable to hu-

*Equal contribution.
†Corresponding author.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 547–552

man writing (Achiam et al., 2023), making it even challenging for humans to differentiate between LLM-generated text and human-written text. Consequently, there is a need to develop effective detectors to mitigate the potential misuse of LLM (Wu et al., 2023). Recently, owing to the construction of numerous high-quality benchmarks and innovations in detection methods, significant progress has been made in LLM-generated text detection technology.

High-quality datasets play a crucial role in advancing research on detecting LLM-generated text. HC3 dataset (Guo et al., 2023), represents one of the pioneering open-source efforts aimed at comparing ChatGPT-generated text with human-written text.The CHEAT dataset (Yu et al., 2023) comprises academic abstracts written by humans sourced from IEEE Xplore, and is committed to detecting artificially generated deceptive academic content from ChatGPT. Additionally, there are numerous datasets containing text generated by various LLMs, such as monolingual datasets DeepfakeText-Detect-Dataset (Li et al., 2023), GPT-written dataset (Liu et al., 2023b), and M4 (Wang et al., 2023), used in this competition.

Focusing on recently proposed detection methods, these primarily encompass zero-shot (Corston-Oliver et al., 2001), fine-tuning LMs (Qiu et al., 2020), adversarial learning (Hu et al., 2023), and LLMs as detectors (Koike et al., 2023). DetectGPT (Mitchell et al., 2023) is dedicated to the detection of LLM-generated text by analyzing the structural attributes inherent in the probability functions of LLMs. Fagni et al. (2021) noted that fine-tuning RoBERTa (Liu et al., 2019a) resulted in optimal classification outcomes across diverse encoding configurations. Recent studies (Liu et al., 2023a; Chen et al.) have additionally supported the outstanding performance of fine-tuned variants within the BERT family, such as RoBERTa, in discerning LLM-generated text. Yang et al. (2023) conducted an adversarial data augmentation process on LLM-generated text, and the results showed that models trained with augmented data exhibited enhanced robustness.

3 Methodology

In this section, we present our GMoEF in details. We first introduce the overall architecture of the proposed GMoEF, then give a comprehensive insight of the adopted parametric whitening and gated mixture-of-experts

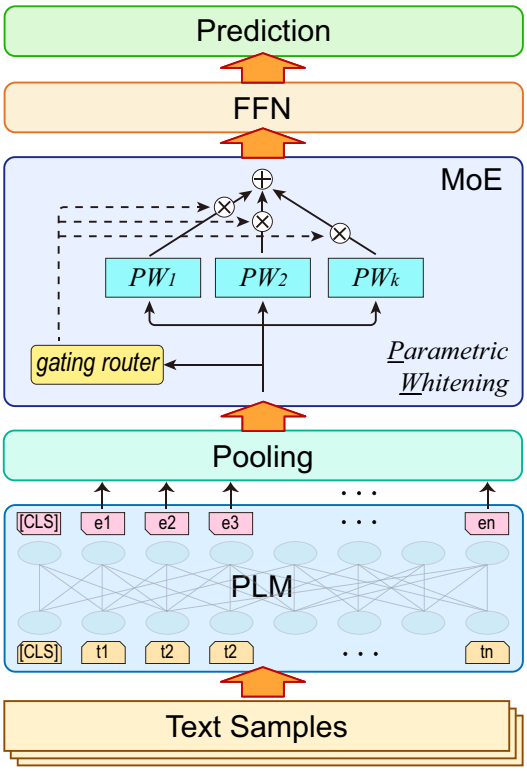

Figure 1: The main architecture of GMoEF.

3.1 System Architecture

The overall architecture is shown in Figure 1. Basically, our GMoEF follows the fine-tuning PLM as the classifier paradigm. We first employ a PLM as the text encoder. For text sample s_i, we take the last layer output at each token position through a mean pooling layer to obtain the text embedding x_i. Notably, we do not take the commonly adopted `[CLS]` position output as the text embedding. Further discussion can be found in section 4.3. On acquiring the text embedding, we put it through a gated mixture-of-expert layer, in which we adopt parametric whitening module as the expert, to learn the language feature of LLM-generated text. Finally, we employ a feed-forward network to give the final probability score $\widehat{y}_i$. We then use the cross-entropy loss as the optimization target.

$$\mathcal{L} = -\sum_{i=1}^{k} y_i \log(\widehat{y}_i) + (1 - y_i) \log(1 - \widehat{y}_i). \quad (1)$$

3.2 Parametric Whitening

While we can utilize a PLM to encode texts into embeddings, current studies have revealed that PLMs induce a non-smooth, anisotropic semantic space

for general texts(Li et al., 2020). Anisotropy issue makes the embeddings occupy a narrow cone, resulting in a high similarity between any embedding pairs. Consequently, this situation can have a negative impact on downstream classification tasks (Jiang et al., 2022). The problem is further exacerbated when mixing texts generated by multiple LLMs and written by humans. Drawing inspiration from recent studies that aim to improve PLM-generated text embeddings through whitening-based methods (Su et al., 2021; Huang et al., 2021), we incorporate a simple linear transformation to transform the original PLM generated embeddings for deriving isotropic representations. Unlike previous whitening-based methods, we make mean and variance as two learnable parameters, for better generalizability. We define the whitening transformation as:

$$\widetilde{x_i} = (x_i - \mathbf{b}) \cdot \mathbf{W}_1, \tag{2}$$

where $x_i \in \mathbb{R}^d$ is the original text embedding, while $\mathbf{b} \in \mathbb{R}^d$ and $\mathbf{W}_1 \in \mathbb{R}^{d \times d'}$ are all parameters to learn. $\widetilde{x_i}$ is the transformed text embedding.

3.3 Gated Mixture-of-Experts

As mentioned earlier, the dynamic language style of LLM-generated text poses a significant challenge for all detecting methods. We contend that conventional methods are only capable of capturing limited or partial aspects of the pattern. To this end, we employ multiple parametric whitening layers to learn a series of whitening embeddings. Each embedding will focus on a certain aspect of the language style, and we make the final decision based on all these embeddings to draw a more robust conclusion.

To implement our idea, we employ the mixture-of-experts (MoE) architecture (Jacobs et al., 1991; Eigen et al., 2013). More sepcifically, we employ k parametric whitening layers as the *experts*, then employ a gating router (Shazeer et al., 2016; Hou et al., 2022) to aggregate them. For text sample s_i, the gated mixture-of-expert output v_i is defined as:

$$v_i = \sum_{j=1}^{k} g_j \widetilde{x_i}^{(j)}, \tag{3}$$

where $\widetilde{x_i}^{(j)}$ represents j-th whitening transformed embedding for text sample s_i. g_j is the weight derived from the gating router, which is defined as follows:

$$\boldsymbol{g} = \text{Softmax}(x_i \cdot \mathbf{W_2} + \boldsymbol{\delta}), \tag{4}$$

$$\boldsymbol{\delta} = \boldsymbol{\epsilon} \cdot \text{Softplus}(x_i \cdot \mathbf{W_3}). \tag{5}$$

where $\boldsymbol{g} \in \mathbb{R}^k$ is the routing vector. We employ two learnable parameters $\mathbf{W_2}$ and $\mathbf{W_3}$ to dynamically adjust the weight for each expert. Inspired by Inoue (2019), we incorporate a series of noises $\boldsymbol{\delta}$ in the gating router to balance these experts and avoid overfitting.

4 Experiment

4.1 Experimental setup

Dataset and Evaluation. A sampled version of M4 (Wang et al., 2023) dataset provided by the organizer was adopted. Comprehensive statistics regarding the dataset can be found in Table 1. Subtask A focuses on detecting single-model generated text while subtask B focuses on the multi-model generated text distinguish. However, subtask C has a very different optimization target comparing to subtask A and B, we opted not to conduct experiments on this particular subtask. As mentioned in the official task description, we employed Accuracy as the evaluation metric to assess the quality of the detection.

Subtask	#Train	#Dev	#Test
A (mono.)	119,757	5,000	34,272
A (multi.)	172,417	4,000	42,378
Subtask B	71,027	3,000	18,000

Table 1: Statistics on subtask A (monolingual & multilingual) and subtask B.

Implementation details. We implemented the GMoEF model based on RoBERTa[1] (Liu et al., 2019b) and XLM-R[2] (Conneau and Lample, 2019) for monolingual and multilingual scenarios respectively, with Pytorch (Paszke et al., 2019) and the Huggingface Transformers library (Wolf et al., 2020). To facilitate distributed training, we utilized the pytorch-lightning framework (Falcon and The PyTorch Lightning team, 2019).

For optimization, we used the AdamW optimizer with an initial learning rate of $2e^{-4}$ for the RoBERTa part and $2e^{-5}$ for the non-RoBERTa parts. The learning rate was linearly decayed with 10% warm-up steps. The hyperparameter settings

[1] https://huggingface.co/FacebookAI/roberta-large
[2] https://huggingface.co/FacebookAI/xlm-roberta-large

Hyperparameter	Symbol	Value
Maximum words (tokens)	-	512
# of experts	k	3
# of epochs	-	3
weight decay	-	$1e^{-2}$
seed	-	42
batch size	-	$32^{\dagger}$
hidden dim	d'	256
PLM embedding dim	d	1024

Table 2: The hyperparameters of the experiment. †: on a single GPU.

we employed are summarized in Table 2. All models are trained on two NVIDIA-SXM4-A100 GPUs.

4.2 Main Results

The main results on test set are shown in Table 3. Our GMoEF exhibits impressive results in both subtask A and subtask B. However, our original submissions (*orig. sub.*) for subtask A is not satisfying as expected. We attribute this discrepancy as two folds: 1) It is possible that we failed to identify the optimal checkpoint for generating predictions on the test set due to a substantial disparity between the number of training and evaluation samples. 2) We searched for the optimal number of experts (k) from 4 up to 10 during submission stage, however, the best result shows up at $k = 3$.

We further find out that our GMoEF shows more significant performance improvements on subtask A (multilingual) and subtask B over the baselines. However, interestingly, the GMoEF does not exhibit significant advantages in subtask A (monolingual). It may indicates that the GMoEF is better suited for complex scenarios, for instance, the texts are *multilingual* and may generated by *multiple models*.

4.3 Ablation Study

In order to validate the unique contribution of each module, we conduct experiments on the following variants of GMoEF:

- Without parametric whitening (*w/o* PW). In this variant, we substitute all parametric whitening layers into the linear layers.
- Using `[CLS]` position output as the text embedding (*alt.* PLM).

As shown in Table 3, all variants will lead to immediate performance drop on all subtasks, which further validates the necessity and effectiveness of all proposed model components. Through these results, we have several noteworthy observations: (1) The multilingual and multi-model cases exhibit more severe anisotropy issue. Removing the PW layer can lead to a substantial decline in performance. (2) The utilization of the `[CLS]` token for text encoding proves to be coarse-grained when it comes to capturing language styles or features in the LLM-generated text detection task. In this context, our token position pooling strategy emerges as a more suitable alternative.

Model	**A** (mono.)	**A** (multi.)	**B**
baseline	0.885	0.809	0.746
orig. sub.	0.806	0.768	0.822
GMoEF	**0.903**	**0.892***	**0.848***
improv.	2.03%	10.3%	13.7%
w/o PW	0.896	0.848	0.732
alt. PLM	0.845	0.808	0.711

Table 3: Experimental results on subtask A (monolingual & multilingual) and subtask B. The best results are marked in **boldface**. *w/o* stands for "without"; *alt.* stands for "alternative". "*" denotes that the improvements are significant at the level of 0.01 with paired *t*-test.

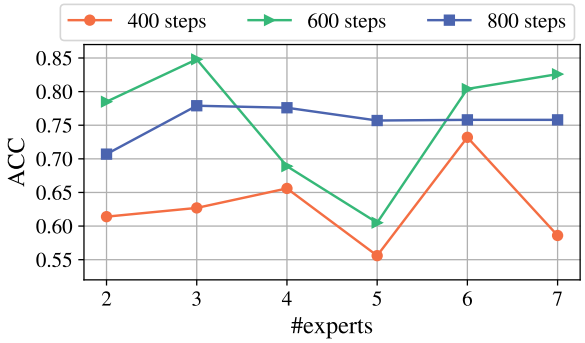

Figure 2: Experimental results on subtask B with different numbers of experts (k). Each line indicates start testing after training for certain steps. Notably, a whole training epoch takes ~1,110 steps under our setup.

4.4 Case Study on Number of Experts (k)

To reveal the effectiveness of our proposed gated mixture-of-experts fine-tuning, we further conduct experiments with different numbers of experts. Detailed results are shown in Figure 2, from these results, we have the following observations: (1) With the assistance of multiple experts, the model tends to converge much earlier, often requiring less

than a full epoch of training. The complete training process for subtask B takes about 1,110 steps. However, as shown in Figure 2, the optimal result is achieved at the 600th step with 3 experts. By the 800th step, the performance becomes expert-agnostic and suboptimal, indicating overfitting. (2) Our GMoEF achieves best performance with $k = 3$. With fewer experts, GMoEF can hardly capture the dynamic language features of LLM-generated text, and revert to conventional fine-tuning models. On the other hand, increasing the number of experts does not necessarily guarantee a better outcome. For instance, when $k = 5$, these experts may reach conflicting conclusions, leading to the worst result. While adding more experts may mitigate this phenomenon, it also introduces additional noise, ultimately resulting in suboptimal performance.

5 Conclusion and Future Work

In this work, we find out that current LLM-generated text detection methods may suffer from anisotropy issue, and they fail to capture the dynamic language features. To this end, we propose GMoEF, which incorporates parametric whitening to mitigate the anisotropy issue. GMoEF further adopts the Mixture-of-Experts equipped with gating router to model the pattern of LLM-generated text from multiple aspects. Our GMoEF exhibits an impressive #8 out of 70 participating teams on the multi-model generated text detection subtask. Extensive experiments show that our GMoEF is suitable for complicated scenarios where texts are multi-lingual and may generated by multiple possible LLMs.

In the future, we aim to extend our observations to other text classification tasks, and incorporate LLM itself to detect machine-generated text.

Acknowledgments

We thank our anonymous reviewers for their helpful comments. This work is supported by the National Natural Science Foundation of China (Grant No. 62376051, 62076046).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. Gpt-sentinel: Distinguishing human and chatgpt generated content.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32.

Simon Corston-Oliver, Michael Gamon, and Chris Brockett. 2001. A machine learning approach to the automatic evaluation of machine translation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics - ACL '01*.

David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. 2013. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *PLOS ONE*, page e0251415.

William Falcon and The PyTorch Lightning team. 2019. PyTorch Lightning.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

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.

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 585–593.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. Whiteningbert: An easy unsupervised sentence embedding approach. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 238–244.

Hiroshi Inoue. 2019. Multi-sample dropout for accelerated training and better generalization. *arXiv preprint arXiv:1905.09788*.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Ganesh Jawahar, Muhammad Abdul-Mageed, and LaksV.S. Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. *arXiv: Computation and Language,arXiv: Computation and Language*.

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022. Promptbert: Improving bert sentence embeddings with prompts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8826–8837.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2023. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2023. Deepfake text detection in the wild. *arXiv preprint arXiv:2305.13242*.

Yikang Liu, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. 2023a. Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A robustly optimized bert pretraining approach. *Cornell University - arXiv,Cornell University - arXiv*.

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

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. 2023b. Check me if you can: Detecting chatgpt-generated academic writing using checkgpt.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, ChristopherD. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, page 1872–1897.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2016. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, et al. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. *arXiv preprint arXiv:2305.14902*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

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

Lingyi Yang, Feng Jiang, and Haizhou Li. 2023. Is chatgpt involved in texts? measure the polish ratio to detect chatgpt-generated text.

Peipeng Yu, Jiahan Chen, Xuan Feng, and Zhihua Xia. 2023. Cheat: A large-scale dataset for detecting chatgpt-written abstracts.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.