SemEval-2024 Task 8: Black-Box Detection of Machine-Generated Text via Prompt-Tuning

Giacomo Fidone
Università di Pisa
g.fidone1@studenti.unipi.it

February 2024

Abstract

SemEval-2024 Task 8 introduces several challenges involving black-box machine-generated text detection. In this work I address two sequence classification sub-tasks by leveraging prompt tuning on a Large Language Model (Mistral-7B) and comparing its performance with a zero-shot classifier and a state-of-the-art classification system, i.e. RoBERTa fine-tuned with Low Rank Adaptation (LoRA).

1 Introduction

In recent years, Large Language Models (LLMs) have achieved performance levels in solving language generation tasks comparable to those of humans, making increasingly challenging to discriminate between AI-generated and human-authored text. This difficulty is not a concern for sole research purposes, as it paves the way to potential misuses of LLMs for spreading disinformation and causing disruptions in the education system (Crothers et al. 2023).

The remarkable fluency of LLMs and their ethical implications have revitalized interest both in the academia and the industry in improving state-of-the-art detection tools and devising up-to-date, more reliable solutions. This renewed effort has followed two distinct directions: black-box detection, where the detector has only an API-level access to the LLM; and white-box detection, where the detector can rely on secret watermarks embedded in the LLM's output (Tang et al. 2023). Although the latter may be more accurate, it assumes the cooperation of human developers for ensuring the effectiveness of output traceability. For this reason, despite the ever-growing capabilities of LLMs in emulating human-style writing, black-box detection still stands as the most explored approach.

Nonetheless, there is still limited literature about the exploitation of the latest soft-prompting techniques for approaching black-box machine-generated text detection, such as (notably) prompt tuning (Lester et al. 2021). Prompt tuning has indeed proved to be highly competitive with state-of-the-art classification systems if provided with a sufficiently large LLM (at least 1 billions of parameters); and also outperforms other parameter-efficient approaches in terms of computational efficiency by granting the minimum number of trainable parameters. Prompt tuning represents therefore a valuable candidate for addressing a challenging task like black-box machine-generated text detection, further attesting the benefits of situating a traditional NLP problem into the newest, promising generation-based paradigm (Raffel et al. 2020).

The focus of this paper is the development of two prompt-tuned LLMs based on Mistral-7B (Jiang et al. 2023) for solving two different sub-tasks proposed in the context of «Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection» (Wang et al. 2023) at the SemEval-2024 competition:

1. Sub-task A (monolingual), consisting in binary human-written *vs* machine-generated text classification;

2. Sub-task B, consisting in multi-class text classification, where each text can be written by a human or generated by a specific language model (ChatGPT, Cohere, Davinci, Bloomz or Dolly).

For a comparative evaluation I also consider the development of two baseline models: a zero shot classifier (Mistral-7B), effective for highlighting the gap between hard prompting and soft prompting; and a state-of-the-art classification model, i.e. RoBERTa (Liu et al. 2019) fine-tuned with Low Rank Adaptation (LoRA) (Hu et al. 2021).

2 Related Work

Machine-generated text detection has been a long-standing task in NLP. Black-box detection has been traditionally addressed by training statistical Machine Learning models – typically on top of bag-of-words text representations – or neural architectures (feed-forward and recurrent). Most recently detection systems have leveraged the fine-tuning of deep learning transformer-based models (Solaiman et al. 2019; Fagni et al. 2021), such as BERT (Devlin et al. 2019) and RoBERTa (Liu et al. 2019). Latest works have also experimented with zero-shot learning, such as (Mitchell et al. 2023), which leverages the log probabilities of LLMs; and (Su et al. 2023), which leverages log rank information. On the other hand, white box detection has mainly focused on specializing watermarking techniques (Kirchenbauer et al. 2023; Zhao et al. 2023), which were already implemented for protecting the intellectual property of LLMs with closed-source license.

3 Method

This section briefly outlines the specifics of each classification system, along with the tools and the data employed in the experimental process.

3.1 Systems

As anticipated above, I consider three types of classification systems: Mistral-7B prompt-tuned (PT-Mistral-7B), Mistral-7B conditioned as a zero-shot classifier (ZS-Mistral-7B) and RoBERTa fine-tuned with Low Rank Adaptation (LoRA-RoBERTa).

Mistral-7B (v. 0.1) is a LLM large about 7 billions of parameters, engineered for improving computational efficiency and performance w.r.t. larger LLMs.[1] This result is mainly achieved by improving the attention mechanism with Grouped Query Attention (GQA) and Sliding-Window Attention (SWA) (Jiang et al. 2023). Due to GPU limitations, in this work Mistral-7B has been loaded with half precision (FP16) for both zero-shot classification and prompt tuning.

In zero-shot classification, given labeled task-specific test data $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$, a pre-trained task-agnostic LLM with parameters θ and an unlabeled prompt template $[\mathbf{d}; \mathbf{x}_i]$ – where $\mathbf{d} = \{d_0, d_1, ..., d_k\}$ are task-specific tokens – the LLM is conditioned by each input prompt to generate the correspondent label:

$$\hat{y}_i = \arg\max_{y \in V} P_\theta(y|\mathbf{d}, \mathbf{x}_i) \quad (1)$$

where V is the vocabulary.

At inference time (Equation 1), there is no difference between ZS-Mistral-7B and PT-Mistral-7B. The latter, however, assumes that prompt tokens $\mathbf{d}$ – which shape a labeled prompt template $[\mathbf{d}; \mathbf{x}_i; y_i]$ – are mapped onto an additional trainable module θ_d within the embedding layer, such that the optimal prompt can be found by keeping θ frozen and searching for optimal parameters θ_d^* through standard language modelling:

$$\theta_d^* = \arg\max_{\theta_d} P_{\theta;\theta_d}(y_i|\mathbf{d}; \mathbf{x}_i) \quad \forall i \quad (2)$$

The second baseline, Robustly optimized BERT approach (RoBERTa) (Liu et al. 2019), shares the same transformer-based encoder-only architecture of the well known BERT model (Devlin et al. 2019), but greatly improves its performance through the implementation of dynamic masking, the removal of Next-Sentence Prediction (NSP) objective and the augmentation of data, batch size and training time. The version of RoBERTa I consider is a base version whose size amounts to about 125 millions of parameters.[2]

Despite its modest size, training all the parameters of RoBERTa can be still computationally expensive. In this case, a more suitable parameter-efficient approach is Low Rank Adaptation (LoRA), which allows to reduce the number of trainable parameters by decomposing weight updates ΔW into low-rank matrices (Hu et al. 2021):

$$h(\mathbf{x}_i) = W_0\mathbf{x}_i + \Delta W\mathbf{x}_i = W_0\mathbf{x}_i + BA\mathbf{x}_i \quad (3)$$

so that A and B are updated keeping W_0 frozen.

3.2 Tools

As a well-established practice in Deep Learning, all systems were implemented in Python (v. 3.8.8). For data manipulation I mainly used `dataset` (v. 2.16.0) from HuggingFace. Both pre-trained models (Mistral-7B, RoBERTa) were loaded from the HuggingFace API via `transformers` (v. 4.36.2) and trained using the `torch` framework (v. 2.2.0) in combination with `peft` (v. 0.7.1). For evaluation I opted for the dedicated modules in `scikit-learn` (v. 1.3.2). On the hardware side, training and inference of each system were executed on NVIDIA V100 (32 GB) GPU.

[1] https://huggingface.co/mistralai/Mistral-7B-v0.1
[2] https://huggingface.co/FacebookAI/roberta-base

To ensure reproducibility and facilitate experimentation, all non-deterministic operations implemented by the aforementioned dependencies have been made deterministic through Pseudo-Random Number Generators (PRNGs) with a seed value set to 42. The same holds for all CUDA Deep Neural Network (cuDNN) operations, except for the benchmark mode, as the cuDNN auto-tuner behaves deterministically if provided with fixed model architecture and fixed input size – which is our case.

3.3 Data

Data has been limited to the data supplied by the task organizers – based on an extension of the M4 dataset (Wang et al. 2023) – which include a training set (TR) and a validation set (VL) for each sub-task.[3] Given the absence of labeled test data, a sample of 10% of TR has been separated as test data (TS). The remainder TR has been additionally sampled to alleviate the computational effort of training processes: for sub-task A it has been reduced of 50%; for sub-task B it has been reduced of 20%. All sampling operations have reasonably preserved the distribution of data w.r.t. target classes.

Hard-prompting and prompt tuning both require prompt design, which has followed the template:

$$\text{Text: “}\mathbf{x}_i\text{”. \textbackslash nLabel: } y_i.$$

where $\mathbf{x}_i$ is the i-th text input and y_i is its target value.[4] This template is post-pended to k prompt tokens conveying task-specific instructions (Table 1).

Table 1: Prompt tokens

Task	k	
A	24	Decide if the following text has been written by a human (0) or by a language model (1).
B	51	Decide if the given text has been written by a human (0) or by a language model among: ChatGPT (1), Cohere (2), Davinci (3), Bloomz (4) or Dolly (5).

The choice of the input size has attempted to maximize informativeness within the limits of computational feasibility: for prompt tuning systems it has been set to 143 (A) and 116 (B), i.e. 167 (for both A and B) if we include the number of prompt tokens (k). Since the amount of input information is a decisive factor for correctly recognizing machine-generated text (Gaggar et al. 2023), the comparative evaluation between PT-Mistral-7B and LoRA-RoBERTa could be biased by the different availability of textual information. For this reason, the same input size constraints have been imposed on LoRA-RoBERTa.

Data has been batched for computational efficiency, again within the limits of GPU capabilities: for prompt tuning systems the batch size has been set to 8; for LoRA-RoBERTa to 32.

4 Experiments

This section describes the experimental setup for LoRA-RoBERTa and PT-Mistral-7B.

4.1 LoRA-RoBERTa

LoRA has been configured with parameters $r = 8$ (rank), $\alpha = 16$ (scaling factor for the weight matrices) and $p = 0.1$ (dropout probability of the LoRA layers). This configuration allows to decrease trainable parameters to about 0.71% of total parameters in both sub-tasks.

The loss function employed is standard cross-entropy and the optimizer is AdamW (Loshchilov and Hutter 2017) with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$. Regularization is carried out not only by the aforementioned decoupled weight decay (AdamW) – which significantly improves Tikhonov regularization by decoupling weight decay from gradient computation; but also by early stopping with a patience of 2 epochs and a minimum loss (VL) variation (Δ) of 0.01.

Hyper-parameter selection has been performed manually on learning rate (η) and weight decay coefficient (λ). Selected hyper-parameters for each task can be found in Table 2.

Table 2: LoRA-RoBERTa hyper-parameters

Task	η	λ
A	1×10^{-6}	0.9
B	1×10^{-6}	0.7

Learning curves and classification performance metrics (VL) over epochs – i.e. accuracy and F1 macro – for each sub-task can be visually inspected in Figures 1 and 2.

[3] https://github.com/mbzuai-nlp/SemEval2024-task8

[4] Naturally, the target value y_i must be omitted at inference time when using deterministic (greedy-search) autoregressive generation.

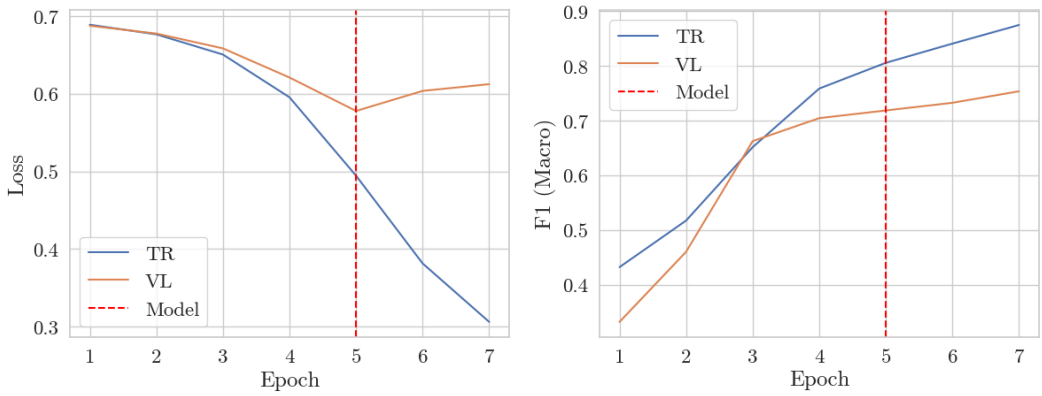

Figure 1: LoRA-RoBERTa: learning curves (left) and F1 (Macro) over epochs (right) for task A.

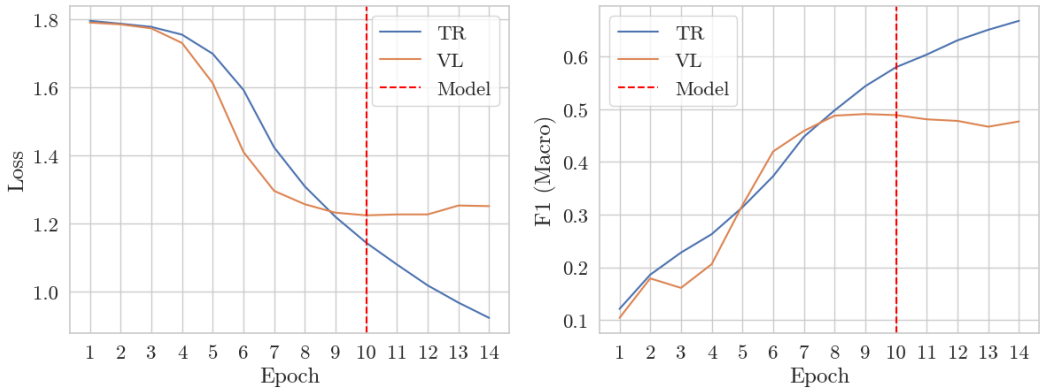

Figure 2: LoRA-RoBERTa: learning curves (left) and F1 (Macro) over epochs (right) for task B.

4.2 PT-Mistral-7B

A preliminary design choice in prompt tuning concerns the initialization of prompt parameters θ_d (Equation 2). Even if random initialization is a valid option, initializing prompt parameters with actual vocabulary embeddings (Table 1) is generally preferred as it situates the model in a favorable region of the parameter space. Given that each embedding has dimensionality 4096, this translates into 98,304 trainable parameters for task A – accounting for approximately 0.0014% of total parameters; and 208,896 trainable parameters for task B – accounting for approximately 0.0028% of total parameters.

The training of Mistral-7B has been performed via language modelling (teacher forcing), using standard cross-entropy as loss function and AdamW as optimizer with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$. Given the large computational cost, also in this case hyper-parameter search has been performed manually on a relatively restricted hyper-parameter space (Table 3), including learning rate (η), weight decay coefficient (λ) and learning rate decay factor (γ). A constant learning rate has indeed resulted in moderate instability (in both tasks) and an effective strategy for preventing loss divergences has been the adoption of an exponential learning rate schedule:

$$\eta^{(t)} := \eta^{(t-1)} \exp(\gamma t)$$

As with LoRA-RoBERTa, regularization is also carried out by early stopping with a patience of 2 epochs and $\Delta = 0.01$.

Table 3: Prompt-tuning hyper-parameters

Task	η	λ	γ
A	0.004	0.3	0.95
B	0.01	0.2	0.95

Learning curves can be visually inspected in Figures 3 and 4. Since the loss is computed as the mean cross-entropy over each teacher forcing iteration, it only provides a global estimation of the LLM's adaptation to the input prompts. To gain a local estimation of the model's performance in solving the downstream task, I also consider task-specific classification metrics, i.e. accuracy and F1-macro. In this regard, it must also be noticed that it is not *a priori* guaranteed that at inference time (Equation 1) $\hat{y}$ matches an expected label. Therefore, whenever this happens it is assumed that the LLM is acting as a random classifier.

This assumption allows to preserve input-output mapping without affecting the reliability of evaluation metrics.

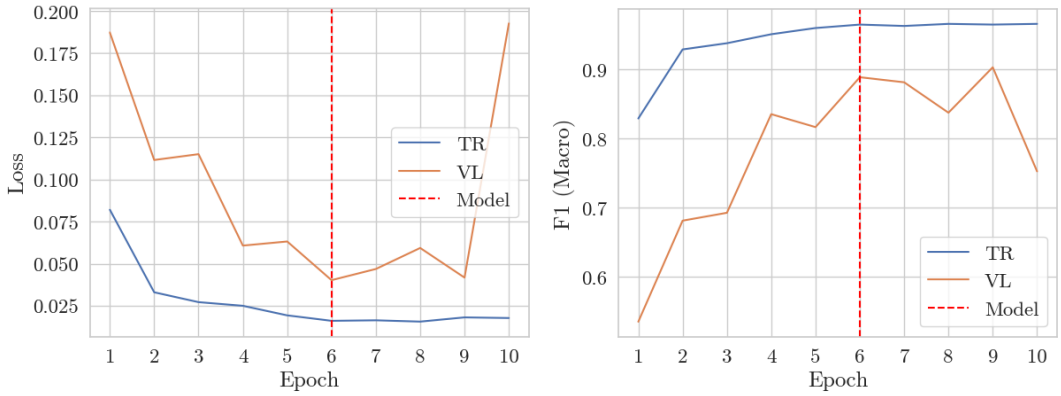

Figure 3: PT-Mistral-7B: learning curves (left) and F1 (Macro) over epochs (right) for task A.

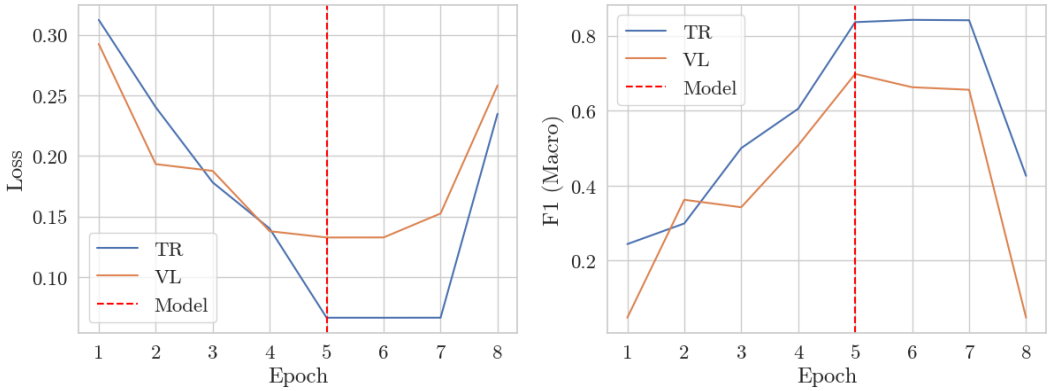

Figure 4: PT-Mistral-7B: learning curves (left) and F1 (Macro) over epochs (right) for task B.

5 Results

For evaluating each classification system I consider as performance metrics the F1-score (of each class), the F1-score macro and the accuracy – given a sufficiently balanced distribution of TS data w.r.t. target classes. Except for ZS-Mistral-7B, which is not able to outperform random classification, all systems demonstrate good performance (TS) in both tasks (Tables 4, 5). As before, performance metrics for prompt-based systems are computed assuming that, whenever $\hat{y}$ (Equation 1) does not match any expected label, $\hat{y}$ is selected randomly – yet, in practical experimentation this circumstance was observed only in ZS-Mistral-7B's inference.

As evident, for both LoRA-RoBERTa and PT-Mistral-7B performance estimation on VL lags behind performance estimation on TS. This discrepancy, however, cannot be attributed neither to imbalances in data distribution w.r.t. target classes, as both TS and VL exhibit no significant imbalances; neither to a fortunate sampling of TS from TR, as it persists with different samplings; neither by a specific model configuration, as it is observed consistently across both classification systems. Since TS was sampled from TR, it is plausible that VL is not fully representative of TR/TS data, thus leading to an under-estimation of model's performance.

The poor results of ZS-Mistral-7B are not surprising: hard-prompting classification is effective with scale (Brown et al. 2020) and becomes more competitive if also providing task demonstrations (few-shot learning). The significant gap in performance between ZS-Mistral-7B and PT-Mistral-7B, on the other hand, further confirms that the crucial factor lies in prompt optimization, which allows to effectively steer the model for generating correct downstream outputs.

As expected, PT-Mistral-7B is able to outperform LoRA-RoBERTa in both tasks and w.r.t. each performance metric considered. Remarkably, this result has been achieved within the context of GPU limitations, such as half-precision – which could limit the model's capability in effectively represent relevant information – and con-

strained input size – which decreases the amount of input information.

Confusion matrices (Figure 5) provide some final insights on PT-Mistral-7B's performance. PT-Mistral-7B is highly effective in recognizing human-authored texts, as in only few cases it mistakes a machine for a human. However, PT-Mistral-7B frequently mistakes Dolly for a human – which seems to be the language model which most effectively simulates human linguistic behaviour; and also struggles in correctly recognizing Davinci.

Table 4: Results (TS) of task A

Model	F1			Accuracy
	human	machine	macro	
ZS-Mistral-7B	0.650	0.165	0.407	0.507
LoRA-RoBERTa	0.773	0.819	0.796	0.798
PT-Mistral-7B	**0.992**	**0.960**	**0.975**	**0.975**

Table 5: Results (TS) of task B

Model	F1							Accuracy
	human	ChatGPT	Cohere	Davinci	Bloomz	Dolly	macro	
ZS-Mistral-7B	0.271	0.015	0.021	0.021	0.017	0	0.057	0.158
LoRA-RoBERTa	0.736	0.592	0.586	0.342	0.725	0.642	0.604	0.615
PT-Mistral-7B	**0.890**	**0.872**	**0.888**	**0.735**	**0.908**	**0.842**	**0.856**	**0.861**

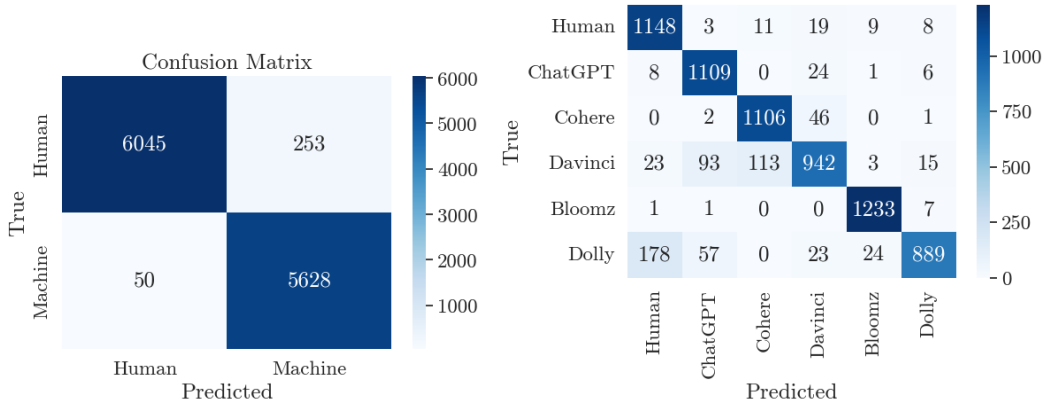

Figure 5: PT-Mistral-7B: confusion matrices (TS) for task A (left) and B (right).

6 Conclusions

Black-box machine-generated text detection has been usually handled with a fine-tuning of deep learning models, remarkably transformer-based encoder-only architectures like BERT and its variants. This work has proved that RoBERTa still stands as an effective detection method, especially in the context of limited computational resources. However, prompt tuning has proved to be highly competitive, even with some computational constraint.

In this respect, a crucial aspect in comparing BERT-based detection and prompt tuning based detection lies in the potential for improvement, as a task like machine-generated text detection is supposed to become increasingly challenging over time. Aside the computational limitations of this work, the performance of a prompt-tuned detection system can significantly benefit from leveraging larger LLMs, which will become always more accessible and efficient with the ever-increasing advancements in computational and storage capabilities. For this reason, prompt-tuning is likely to become a leading state-of-the-art system for solving classification tasks – including black-box machine-generated text detection – also providing all the advantages (e.g. multi-tasking) of the new generation-based paradigm.

References

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. and Agarwal, S. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, pp.1877-1901.

Crothers, E., Japkowicz, N. and Viktor, H.L. (2023). Machine-generated text: A comprehensive survey of threat models and detection methods. IEEE Access.

Devlin, J., Chang, M., Lee K. and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language under- standing. In North American Association for Computational Linguistics (NAACL).

Fagni, T., Falchi, F., Gambini, M., Martella, A. and Tesconi, M. (2021). TweepFake: About detecting deepfake tweets. Plos one, 16(5), p.e0251415.

Gaggar, R., Bhagchandani, A. and Oza, H. (2023). Machine-Generated Text Detection using Deep Learning. arXiv preprint arXiv:2311.15425.

Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. and Chen, W. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D.D.L., Bressand, F., Lengyel, G., Lample, G., Saulnier, L. and Lavaud, L.R. (2023). Mistral 7B. arXiv preprint arXiv:2310.06825.

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I. and Goldstein, T. (2023). A watermark for large language models. arXiv preprint arXiv:2301.10226.

Lester, B., Al-Rfou, A. and Constant, N. (2021). The Power of Scale for Parameter-Efficient Prompt Tuning. arXiv preprint arXiv:2104.08691v2

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D. and Finn, C. (2023). Zero-shot machine-generated text detection using probability curvature. arXiv:2301.11305.

Raffel, C., Shazeer N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research, 21(1), pp.5485-5551.

Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., Radford, A., and Wang, J. (2019). Release Strategies and the Social Impacts of Language Models. CoRR, abs/1908.09203.

Su, J., Zhuo, T. Y., Wang, D., and Nakov, P. (2023). DetectLLM: Leveraging Log Rank Information for Zero-Shot Detection of Machine-Generated Text. arXiv preprint arXiv:2306.05540.

Tang, R., Chuang, Y.N. and Hu, X. (2023). The science of detecting llm-generated texts. arXiv preprint arXiv:2303.07205.

Wang, Y., Mansurov, J., Ivanov, P., Su, J.,Shelmanov, A., Tsvigun, A., Whitehouse, C., Afzal, O. M., Mahmoud, T., Aji, A. F., and Nakov, P. (2023). M4: Multi-generator, Multi-domain, and Multi-lingual Black-Box Machine-Generated Text Detection. arXiv preprint arXiv:2305.14902v1

Zhao, X., Wang, Y.X. and Li, L. (2023). Protecting language generation models via invisible watermarking. arXiv preprint arXiv:2302.03162.