NCL-UoR at SemEval-2024 Task 8: Fine-tuning Large Language Models for Multigenerator, Multidomain, and Multilingual Machine-Generated Text Detection

Feng Xiong[1] and **Thanet Markchom**[2] and **Ziwei Zheng**[1] and **Subin Jung**[1] and **Varun Ojha**[1] and **Huizhi Liang**[1]
[1]School of Computing, Newcastle University, Newcastle upon Tyne, UK
[2]Department of Computer Science, University of Reading, Reading, UK
xf199912@163.com, t.markchom@pgr.reading.ac.uk, {z.zheng21, s.jung4, varun.ojha, huizhi.liang}@newcastle.ac.uk

Abstract

SemEval-2024 Task 8 introduces the challenge of identifying machine-generated texts from diverse Large Language Models (LLMs) in various languages and domains. The task comprises three subtasks: binary classification in monolingual and multilingual (Subtask A), multi-class classification (Subtask B), and mixed text detection (Subtask C). This paper focuses on Subtask A & B. To tackle this task, this paper proposes two methods: 1) using traditional machine learning (ML) with natural language preprocessing (NLP) for feature extraction, and 2) fine-tuning LLMs for text classification. For fine-tuning, we use the train datasets provided by the task organizers. The results show that transformer models like LoRA-RoBERTa and XLM-RoBERTa outperform traditional ML models, particularly in multilingual subtasks. However, traditional ML models performed better than transformer models for the monolingual task, demonstrating the importance of considering the specific characteristics of each subtask when selecting an appropriate approach.

1 Introduction

Large Language Models (LLMs) are sophisticated natural language processing (NLP) models extensively trained on vast textual datasets (Wang et al., 2023). These models demonstrate an impressive proficiency in generating human-like text based on the input they receive. However, using LLMs for generating texts has raised concerns about potential misuse, such as disseminating misinformation and disruptions in the education system (Wang et al., 2023). Thus, urgent development of automated systems to detect machine-generated texts is essential (Mitchell et al., 2023; Wang et al., 2023).

Recently, several LLMs have been developed such as ChatGPT[1] Brown et al. (2020), Cohere[2], Davinci[3], BLOOMZ[4] (Muennighoff et al., 2022), and Dolly[5] (Conover et al., 2023). The versatility of these models extends across various domains, such as news, social media, educational platforms, and academic contexts, in multiple languages not only English (Wang et al., 2023). This wide application poses a challenge in developing an automated system capable of detecting machine-generated texts from various generators, across multiple domains and languages.

To tackle this challenge, SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection (Wang et al., 2024) introduces the task of detecting machine-generated texts obtained from different LLMs, in various domains and languages. This task consists of three subtasks: Subtasks A, B, and C. Subtask A involves binary classification of text as either human-written or machine-generated, with two tracks: monolingual (English only) and multilingual. Subtask B focuses on multi-class classification of machine-generated text, aiming to identify the source of generation, whether human or a specific language model. Subtask C addresses the detection of human-machine mixed text, requiring the determination of the boundary where the transition from human-written to machine-generated occurs in a mixed text. This paper focuses on Subtasks A and B. To tackle these tasks, we propose two approaches: (1) classical machine learning, leveraging NLP techniques for feature extraction, and (2) fine-tuning LLMs for the classification of human-written and machine-generated texts.

2 Related Work

Researchers have employed a variety of methods and tools to detect AI-generated texts. Broadly,

[1]https://chat.openai.com/
[2]https://cohere.com
[3]https://platform.openai.com/docs/models/gpt-base
[4]https://huggingface.co/bigscience/bloomz
[5]https://huggingface.co/databricks/dolly-v2-12b

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 163–169
June 20-21, 2024 ©2024 Association for Computational Linguistics

these approaches can be categorized into two main types: black-box and white-box detection methods (Tang et al., 2023). Black-box detection relies on API-level access to LLMs, utilizing textual samples from both human and machine sources to train classification models (Dugan et al., 2020). The study by Guo et al. (2023) integrated existing question-and-answer datasets and leveraged fine-tuning of pre-trained models to investigate the characteristics and similarities between human-generated and AI-generated texts.

As for white-box detection, Kirchenbauer et al. (2023) introduced a novel approach involving the embedding of watermarks in the outputs of LLMs to facilitate the detection of AI-generated text. Additionally, a variety of tools and methodologies, including XGBoost, decision trees, and transformer-based models, have been evaluated for their efficacy in detecting texts produced by AI (Zaitsu and Jin, 2023). These techniques incorporate multiple stylistic measurement features to differentiate between AI-generated and human-generated texts (Shijaku and Canhasi, 2023).

Specific tools and techniques in this domain include the GLTR tool developed by Gehrmann et al. (2019), which analyzes the usage of rare words in texts to distinguish between those generated by the GPT-2 model and human writers. The DetectGPT method posits that minor rewrites of LLM-generated texts tend to reduce the log probability under the model, a hypothesis that has been explored in depth (Mitchell et al., 2023). Furthermore, intrinsic dimension analysis, including methods like the Persistent Homology Dimension estimator (PHD), has been applied to distinguish between authentic texts and those generated artificially (Tulchinskii et al., 2023). Detectors specifically designed for certain LLMs, such as the GROVER detector for the GROVER model (Zellers et al., 2019) and the RoBERTa detector using the RoBERTa model (Liu et al., 2019), also play a significant role in this field.

In summary, the combination of statistical analysis with advanced language models is being employed by researchers to more effectively differentiate between content generated by humans and machines. The continuous evolution and refinement of these techniques reflect the dynamic nature of the field and the complexities involved in distinguishing between the increasingly nuanced outputs of LLMs and human-authored texts.

3 Methods

To tackle these tasks, we employ two distinct strategies. The first is classical machine learning, tailored for natural language preprocessing (NLP). The second approach involves transformer-based LLMs, with an emphasis on LoRA (Low-Rank Adaptation of Large Language Models) fine-tuning (Hu et al., 2021). We then enhance our results by integrating these methods through ensemble techniques.

3.1 Machine Learning Models

Our approach for textual data analysis in machine learning involves a concise yet comprehensive preprocessing pipeline. Initially, URLs and excess whitespace are removed from the text. Next, all punctuation is eliminated, focusing solely on alphanumeric characters. The text is further refined by excluding common stopwords and numeric characters. Emojis are decoded into text, providing additional context. Lemmatization standardizes words to their base forms, ensuring consistent analysis. Texts are then converted to lowercase for uniformity.

The final step involves using a *Term Frequency-Inverse Document Frequency* (TF-IDF), configured to handle a maximum of 8000 features and considering unigrams to trigrams. This vectorizer excludes terms appearing in less than 10 documents, balancing feature representation with computational efficiency. Furthermore, we enhance the feature set for machine learning by incorporating esteemed readability metrics such as the *Gunning fog index* (Scott, 2023) and *Flesch reading ease score* (Kincaid et al., 1975) into our text analysis, which assess the complexity and readability of the text respectively. This preprocessing strategy transforms raw text into a structured numerical format, ready for machine learning model analysis.

Expanding our feature extraction capabilities, we introduce additional dimensions of analysis including perplexity measures, sentiment analysis, document and error analysis, text vector features, the AI Feedback Query feature, and list lookup features. Perplexity measures assess text complexity through language models, offering insights into predictability. Sentiment analysis is deepened to reveal emotional tones and subjective nuances, providing a fuller understanding of the text's emotional landscape and authorial intent. Document and error analysis afford a detailed look at structure and linguistic accuracy, enhancing content quality assessment. Text vector features, leveraging Sentence-BERT embeddings, enable sophisticated semantic content capture, facilitating nuanced thematic analysis. The AI Feedback Query feature is a binary response achieved through a structured inquiry where the AI model is presented with the text and asked to determine its generative source. List lookup features, examining elements like stop word frequency and special character use, offer stylistic and structural insights. Collectively, these advancements enable a comprehensive and detailed interpretation of textual data, significantly broadening our analytical capabilities by combining them.

In our study, we employed four distinct machine learning algorithms for both binary and multi-class classification tasks: Logistic Regression (LR), Multinomial Naive Bayes Classifier (MultinomialNB), eXtreme Gradient Boosting (XGBoost) (Chen and Guestrin, 2016) and Random Forest (RM).

- LR: A linear model used for classification tasks. It models the probability that a given input belongs to a certain class. Logistic Regression is particularly effective for binary classification due to its simplicity and efficiency in estimating probabilities.

- MultinomialNB: This algorithm is based on the Bayes theorem and is particularly suited for classification with discrete features (like word counts for text classification). It assumes independence between predictors and is highly scalable to large datasets.

- XGBoost: This is an efficient and scalable implementation of gradient-boosted decision trees. It is known for its performance and speed, especially in structured or tabular data, and can handle both binary and multi-class classification problems effectively.

- RF: A versatile ensemble learning method that builds multiple decision trees for classification or regression tasks. It improves accuracy by averaging or taking the mode of predictions from all trees, effectively reducing overfitting. Suitable for both binary and multi-class problems, it excels in handling large, high-dimensional datasets.

By integrating these algorithms, our approach leverages the strengths of linear modeling, probabilistic classification, and ensemble learning, aiming to enhance predictive accuracy and robustness across diverse classification scenarios.

3.2 XLM-RoBERTa

In our approach, we established XLM-RoBERTa[6] (Conneau et al., 2019) as the baseline model among transformer-based architectures. XLM-RoBERTa represents a multilingual adaptation of the original RoBERTa (Liu et al., 2019) model, specifically designed to understand and process a diverse range of languages. XLM-RoBERTa is pre-trained on a substantial dataset: 2.5TB of filtered CommonCrawl data (Zhang et al., 2020), encompassing text in 100 different languages. This extensive pre-training enables the model to capture nuanced language features and patterns across a broad linguistic spectrum, making it highly effective for tasks involving multiple languages. The use of such a diverse training dataset aids in achieving a robust understanding of various linguistic structures and vocabularies, which is crucial for accurate language processing and analysis in a multilingual context.

3.3 LoRA-RoBERTa

To improve the predictive performance of LLMs, we use LoRA for fine-tuning RoBERTa[7] model. LoRA is a technique enhancing the efficiency of fine-tuning large models with reduced memory consumption. It modifies the weight updates in neural networks using two smaller matrices derived through low-rank decomposition. These matrices adapt to new data while the original weights remain unchanged. The final output combines the original and adapted weights. In transformer models, LoRA is often applied to attention blocks for efficiency. The number of trainable parameters depends on the low-rank matrices' size, influenced by the rank and the original weight matrix's shape (Hu et al., 2021), as shown in Figure 1.

3.4 Majority Voting

The Majority Voting ensemble in this study combines the predictions of two transformer-based models: XLM-RoBERTa and LoRA-RoBERTa. The final prediction is determined by the majority vote of these two models, offers several advantages over a single-model approach. This technique, applicable in scenarios with N classifiers $(C_1, C_2, \ldots, C_N)$, determines the final out-

[6]https://huggingface.co/xlm-roberta-base
[7]https://huggingface.co/roberta-base

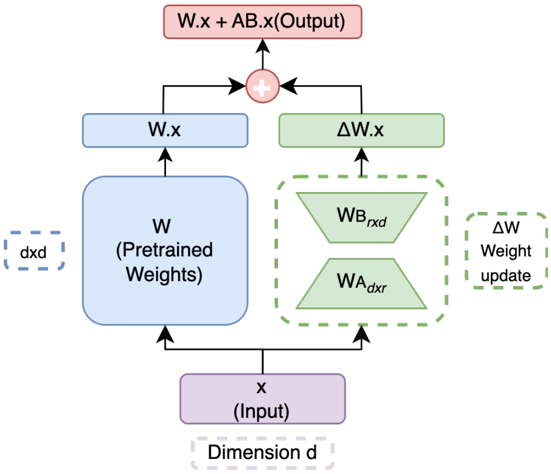

Figure 1: LoRA-based fine-tuning streamlines the process by freezing the original weights of LLMs and training a minimal number of parameters.

put $V(x)$ as the class receiving the most votes: $V(x) = \text{mode}\{C_1(x), C_2(x), \ldots, C_N(x)\}$. This method effectively reduces variance by balancing out individual model errors, leading to more stable predictions. Furthermore, it generally achieves higher accuracy due to the diverse perspectives of different models. Its robustness against overfitting is enhanced, as it combines various models' strengths, making it suitable for a wider range of data scenarios. The flexibility in model selection allows for a blend of different algorithms, each capturing unique data patterns, which contributes to better generalization on unseen data. Thus, majority voting stands out as a robust, accurate, and flexible approach in machine learning.

3.5 DistilmBERT

RoBERTa and XLM-RoBERTa are both powerful but computationally expensive. Therefore, we investigate an alternative model that is more computationally efficient, aiming to compare its performance against these models. We adopted *DistilBERT base multilingual cased*[8] (DistilmBERT) (Sanh et al., 2019), a distilled version of the BERT base multilingual model. It was pretrained on the concatenation of Wikipedia in 104 different languages. DistilmBERT consists of 6 layers, each with 768 dimensions and 12 attention heads, totaling 134 million parameters. This configuration balances model efficiency while retaining significant representational power Sanh et al. (2019).

[8] https://huggingface.co/distilbert-base-multilingual-cased

4 Experiments

In our study, subtask A focuses on distinguishing between human-written (label 0) and machine-generated text (label 1), offered in both monolingual (119,757 train, 5,000 dev, 34,272 test) and multilingual versions (172,417 train, 4,000 dev, 42,378 test), across various sources and languages are given in Table 1. Subtask B, with 71,027 train, 3,000 dev, and 18,000 test, goes further by identifying the specific model (including ChatGPT, Cohere, DaVinci, BloomZ, and Dolly) that generated the text, or if it's human-generated. Both tasks utilize datasets with an identifier, label, text content, model name, and source, focusing on the nuanced classification of texts.

Subtask	#Train	#Dev	#Test
A - Monolingual	119,757	5,000	34,272
A - Multilingual	172,417	4,000	42,378
B	71,027	3,000	18,000

Table 1: Dataset for text classification subtasks

4.1 Parameter Settings

In our experimentation, hyperparameter settings varied between classical machine learning models and LLMs. For the classical machine learning models, we adhered to default parameter settings during training. This approach simplifies the process and relies on the general applicability of these preset parameters.

In contrast, for LLMs, specific hyperparameters were carefully chosen. When training the XLM-RoBERTa baseline model, we set the batch size to 16 and the learning rate to $2.0e-5$ with the model being trained for 3 epochs. This configuration ensures efficient handling of data and optimal learning speed. For fine-tuning the LoRA-RoBERTa base model, the learning rate was adjusted to $1.0e-3$ over 5 epochs, a setting conducive to the specific demands of fine-tuning.

Furthermore, we employed configuration for the LoRA fine-tuning, defined with the following parameters: *task_type* set to *SEQ_CLS* indicating a sequence classification task, *r* (rank of the low-rank matrices) set to 4, *lora_alpha* (scaling factor for learning rate) at 32, *lora_dropout* to manage overfitting set at 0.01, and *target_modules* focused on the *query* module. These configurations are critical in guiding the fine-tuning process, ensuring that the

Method	Subtask A - Monolingual		Subtask A - Multilingual		Subtask B	
	Dev	Test	Dev	Test	Dev	Test
LR	0.673	0.764	0.473	0.721	0.251	0.393
MultinomialNB	0.555	**0.832**	0.483	0.717	0.435	0.511
XGBoost	0.692	0.800	0.515	0.738	0.540	0.545
RF	0.650	0.825	-	-	0.471	0.524
XLM-RoBERTa	**0.783**	0.717	0.679	**0.875**	**0.735**	0.600
LoRA-RoBERTa	**0.783**	0.811	0.726	0.672	**0.735**	**0.699**
Majority voting	0.735	0.828	**0.728**	0.862	0.717	0.602
DistilmBERT	0.702	0.730	0.670	0.810	0.629	0.619

Table 2: Performance comparison of ML and transformer models on text classification subtasks

adjustments to the model are precisely tailored to enhance performance on the specified task.

As for DistilmBERT, the maximum length of input sequences was set to 512. The AdamW optimizer was employed for training with a learning rate set to $1.0e - 4$ and a batch size of 20. This model was trained for 5 epochs.

4.2 Results and Discussions

In our experiments, we evaluated various models on three distinct subtasks: Subtask A - Monolingual, Subtask A - Multilingual, and Subtask B. Each subtask involved both development (Dev) and test phases. The models tested included traditional machine learning algorithms - LR, MultinomialNB, XGBoost and RF - as well as advanced transformer-based models like XLM-RoBERTa, LoRA-RoBERTa, and DistilmBERT. However, due to the complexity of RF and time constraints, experiments on this approach for Subtask A - Multilingual are still ongoing, we plan to report the results in future work. Additionally, we employed a majority voting ensemble method combining XLM-RoBERTa and LoRA-RoBERTa.

The results, detailed in Table 2, reveal significant variations in model performance across the subtasks, highlighting the strengths and weaknesses of each model. One notable observation is the large performance gap between the dev and test sets for some ML approaches. This discrepancy could be attributed to several factors, such as overfitting, differences in data distribution between the dev and test sets, or the limited complexity of some ML models in capturing the intricacies of the task. Further investigation and error analysis are necessary to fully understand and address these issues.

Subtask A - Monolingual In the monolingual Subtask A, MultinomialNB emerged as a strong performer with the highest test score of 0.832. RF and XGBoost also showed robust performance with test scores of 0.825 and 0.800, respectively. The success of these ML models in the monolingual setting suggests that they can effectively capture relevant features and patterns when dealing with a single language. However, their performance on the dev set was notably lower, indicating potential overfitting or limitations in generalizing to unseen data. Among the transformers, LoRA-RoBERTa was notable with a test score of 0.811, outperforming XLM-RoBERTa, which scored 0.717. DistilmBERT, while not leading, still demonstrated a commendable test score of 0.730, indicating its effectiveness in monolingual contexts. The performance of transformer models in this subtask highlights their ability to capture complex language representations and generalize well to new data.

Subtask A - Multilingual In the challenging multilingual Subtask A, XLM-RoBERTa excelled with the highest test score of 0.875. The Majority Voting ensemble was also highly effective, achieving a test score of 0.862. These results demonstrate the strength of transformer models in handling diverse language inputs and their ability to learn language-agnostic representations. DistilmBERT, with a test score of 0.810, also showed notable effectiveness in multilingual text classification, outperforming traditional models and reflecting its potential in handling complex, diverse language data.

Subtask B In Subtask B, LoRA-RoBERTa led with a Test score of 0.699, follwed by DistilmBERT, achieving a test score of 0.619 and XLM-RoBERTa with 0.600. The strong performance of transformer models in this subtask underscores their versatility and adaptability across different text classification scenarios. Among the traditional models, XGBoost was the most effective, with a test score of 0.545. However, the performance gap between ML models and transformers in Subtask B suggests that the latter are better equipped to handle the specific challenges and complexities of this task.

At the model level, we observed that ML models often struggled with handling rare or out-of-vocabulary words, leading to misclassifications. Transformer models, on the other hand, showed better resilience to such challenges, likely due to their subword tokenization and ability to capture broader context. However, transformers sometimes struggled with very short or noisy inputs, indicating room for improvement in their robustness.

5 Conclusions

The results showed that transformer models, particularly LoRA-RoBERTa and XLM-RoBERTa, performed exceptionally well in most text classification tasks. DistilmBERT represented a more streamlined transformer approach and was also proven to be efficient, especially in multilingual task. Contrary to popular belief, traditional ML models such as MultinomialNB and XGBoost can outperform transformers in monolingual tasks. These findings highlight the importance of carefully considering the characteristics of the task and the trade-offs between model complexity and performance when selecting an appropriate approach.

Our results contribute to the understanding of model selection strategies for text classification and emphasize the need for a nuanced approach that takes into account the specific demands of each subtask. Future research could explore the development of hybrid models that combine the strengths of traditional ML techniques and transformer architectures, as well as the design of more efficient and lightweight transformer models for resource-constrained environments. These findings reflected the dynamic nature of NLP tools and the importance of selecting models based on the specific requirements of the task.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, and Chris Callison-Burch. 2020. Roft: A tool for evaluating human detection of machine-generated text. *arXiv preprint arXiv:2010.03070*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

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. *arXiv preprint arXiv:2301.07597*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. *arXiv preprint arXiv:2301.10226*.

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

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Brian Scott. 2023. The gunning's fog index (or fog) readability formula.

Rexhep Shijaku and Ercan Canhasi. 2023. Chatgpt generated text detection. *Publisher: Unpublished*.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting llm-generated texts. *arXiv preprint arXiv:2303.07205*.

Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Serguei Barannikov, Irina Piontkovskaya, Sergey Nikolenko, and Evgeny Burnaev. 2023. Intrinsic dimension estimation for robust detection of ai-generated texts. *arXiv preprint arXiv:2306.04723*.

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*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

Wataru Zaitsu and Mingzhe Jin. 2023. Distinguishing chatgpt (-3.5,-4)-generated and human-written papers through japanese stylometric analysis. *arXiv preprint arXiv:2304.05534*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Hao Zhang, Jae Ro, and Richard Sproat. 2020. Semi-supervised url segmentation with recurrent neural networks pre-trained on knowledge graph entities. *arXiv preprint arXiv:2011.03138*.