L3i++ at SemEval-2024 Task 8: Can Fine-tuned Large Language Model Detect Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text?

Hanh Thi Hong Tran[1,2,3] and **Tien Nam Nguyen**[1]
and **Antoine Doucet**[1] and **Senja Pollak**[2]
[1] University of La Rochelle, L3i, La Rochelle, France
[2] Jožef Stefan Institute, Ljubljana, Slovenia
[3] Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
{firstname.lastname}@univ-lr.fr, senja.pollak@ijs.si

Abstract

This paper summarizes the participation of the L3i laboratory of La Rochelle University (L3i++) in *SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection*. In this task, we aim to solve two over three Subtasks: (1) Monolingual and Multilingual Binary Human-Written vs. Machine-Generated Text Classification; and (2) Multi-Way Machine-Generated Text Classification. We propose a comparative study among three groups of methods to trigger the detection: (1) Using metric-based models; (2) Using a fine-tuned sequence-labeling language model (LM); and (3) Using a fine-tuned large-scale language model (LLM). Our findings show that LLM surpassed the performance of traditional sequence-labeling LM as the benchmark and metric-based approaches. We ranked $5^{th}/62$ in Multilingual Binary Human-Written vs. Machine-Generated Text Classification and $6^{th}/70$ Multi-Way Machine-Generated Text Classification on the leaderboard. Our code is publicly available at https://github.com/honghanhh/semeval8.

1 Introduction

The rise of large language models (LLMs) has led to a significant step forward in producing remarkably controllable, fluent, and grammatical text, triggering a surge in machine-generated content across diverse platforms such as news, social media, question-answering forums, educational, and even academic contexts. Notably, recent LLMs like ChatGPT[1] and GPT-4 (OpenAI, 2023) exhibit a remarkable ability to generate coherent and contextually appropriate responses to a wide array of user queries.

Unfortunately, use and abuse come hand in hand. Although the fluency of these generated texts positions LLMs as potential candidates for replacing human labor in numerous applications, this has also raised concerns about their potential for misuse, particularly in spreading misinformation and causing disruptions within the education system. Given that humans struggle to distinguish between machine-generated and human-written text, it becomes imperative to develop automated systems capable of identifying machine-generated text to curb the risks associated with its misuse.

In this paper, as the participants in *SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection* (Wang et al., 2024), we investigate the feasibility of training a classifier that can reliably differentiate between text generated by humans and text that appears human-like but is generated by machines in two paradigms:

- Subtask A: Given a full text, determine whether it is human-written or machine-generated in monolingual (only English sources) and multilingual versions.
- Subtask B: Given a full text, determine who generated it (human-written or generated by a specific language model).

To address these problems, we explore the performance of diverse methodologies, which can be divided into three categories, including:

- Five different metric-based methods: Log-Likelihood, Rank, Log-Rank, Entropy, and DetectGPT (He et al., 2023).
- Two traditional sequence-labeling language models: monolingual RoBERTa$_{large}$[2] (Liu et al., 2019) and multilingual XLM-R$_{large}$[3] (Conneau et al., 2020).
- A large language model (LLM): $LLaMA-2-7b-hf$[4] (LLaMA-2) (Touvron et al., 2023).

[1] https://chat.openai.com/

[2] FacebookAI/roberta-large

[3] FacebookAI/xlm-roberta-large

[4] NousResearch/Llama-2-7b-hf

This paper is organized as follows. We present related work in Section 2, followed by Section 3, where we introduce the data used to solve this challenge. Our proposed methods are described in Section 4 before we present our findings and an error analysis in Section 5. Finally, in Section 6 we present our conclusions, and future work and discuss the limitations of the proposed methods.

2 Related Work

The success of LLMs in various downstream NLP tasks (Perez et al., 2021; Vilar et al., 2022; Hegselmann et al., 2023) leads to the overuse and abuse of the information generated by LLMs. However, it is essential to acknowledge that the outputs generated by LLMs are not always accurate, giving rise to the issue of hallucination (Azamfirei et al., 2023). Consequently, there is a need for clear differentiation in addressing this concern.

To address these issues, researchers have developed several automatic detection methods (Badaskar et al., 2008; Zellers et al., 2019; Ippolito et al., 2020; Uchendu et al., 2021) that can identify the machine-generated text from the human-written text, which initially can be divided into two categories, i.e., metric-based methods and model-based methods.

2.1 Metric-based methods

Metric-based methods leverage pre-trained LLMs to process the text and extract distinguishable features from it, e.g., the rank or entropy of each word in a text conditioned on the previous context. Then, predicted distribution entropy determines whether a text belongs to machine-generated or human-written texts. Some metric-based detection methods include Log-Likelihood, Rank, Entropy, GLTR, Log-Rank, and DetectGPT (He et al., 2023), to cite a few.

2.2 Model-based methods

In the model-based methods (Zellers et al., 2019; Habibzadeh, 2023; Guo et al., 2023), the classification models are trained using a corpus that contains both machine-generated or human-written texts to make predictions, for example, ChatGPT Detector (Guo et al., 2023), GPTZero (Habibzadeh, 2023), LM Detector (Ippolito et al., 2020), to mention a few.

Regarding *SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection* (Wang et al., 2024), RoBERTa (Liu et al., 2019) and XLM-R (Conneau et al., 2020) are two language models that can be considered as the baseline for these specific tasks.

2.3 Challenges

Yet, there is currently no existing framework capable of automatically distinguishing between human-written and machine-generated texts at both binary and multi-way paradigms outlined in the described tasks as well as no existing free available architecture taking advantage of recent open-sourced LLMs to tackle the issue.

3 Data

We work on two datasets provided by *SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection* (Wang et al., 2024), whose statistics covering the number of examples for each source and each label are presented in Tables 1 and 2 for Subtask A and B, respectively.

Labels	**Human**				**Machine**			
Source	**Monolingual**		**Multilingual**		**Monolingual**		**Multilingual**	
	Train	**Dev**	**Train**	**Dev**	**Train**	**Dev**	**Train**	**Dev**
arxiv	15,498	500	15,998	-	11,999	500	14,999	-
peerread	2,357	500	2,857	-	9,374	500	11,708	-
reddit	15,500	500	16,000	-	12,000	500	14,999	-
wikihow	15,499	500	15,999	-	12,000	500	15,000	-
Wikipedia	14,497	500	14,997	-	11,033	500	14,032	-
bulgarian	-	-	6,000	-	-	-	6,000	-
chinese	-	-	6,000	-	-	-	5,934	-
urdu	-	-	3,000	-	-	-	2,899	-
indonesian	-	-	2,995	-	-	-	3,000	-
russian	-	-	-	1,000	-	-	-	1,000
arabic	-	-	-	500	-	-	-	500
german	-	-	-	500	-	-	-	500
Total	63,351	2,500	83,846	2,000	56,406	2,500	88,571	2,000

Table 1: Subtask A

In Subtask A of the monolingual version, both the training and development sets are sourced from the same data group for both labels. However, in the multilingual version of Subtask A and Subtask B, the development set is sourced from different places compared to the training set.

For both versions of Subtask A, data were collected from diverse sources, leading to label imbalances. For example, in the monolingual Subtask A training set, there is a notable scarcity of samples from *peerread* compared to the other sources. Conversely, in Subtask B, the dataset is balanced.

Labels	**Source**	**Train**	**Dev**	**Labels**	**Source**	**Train**	**Dev**
Human	arxiv	2,998	-	**davinci**	arxiv	2,999	-
	reddit	3,000	-		reddit	2,999	-
	wikihow	2,999	-		wikihow	3,000	-
	Wikipedia	3,000	-		Wikipedia	3,000	-
	peerread	-	500		peerread	-	500
total		11,997	500				
chatGPT	arxiv	3,000	-	**bloomz**	arxiv	3,000	-
	reddit	3,000	-		reddit	2,999	-
	wikihow	3,000	-		wikihow	3,000	-
	Wikipedia	2,995	-		Wikipedia	2,999	-
	peerread	-	500		peerread	-	500
total		11,995	500	*total*		11,998	500
cohere	arxiv	3,000	-	**dolly**	arxiv	3,000	-
	reddit	3,000	-		reddit	3,000	-
	wikihow	3,000	-		wikihow	3,000	-
	Wikipedia	2,336	-		Wikipedia	2,702	-
	peerread	-	500		peerread	-	500
total		11,336	500	*total*		11,702	500

Table 2: Subtask B

4 Methodology

This section tackles the problem by formulating it as supervised classification tasks. We then introduce our proposed solution architecture for each task, covering the models used, and present how we fine-tuned them with hyperparameter configurations, and how we assessed their performance.

4.1 Problem Statements

4.1.1 Subtask A

We formulate the problem at hand as a binary supervised classification task, whose objective is to learn a mapping between a representation of the text and a binary variable, which is 1 if the text is machine-generated, and 0 otherwise. Mathematically, we learn a function f that, given an input text t_i, represented as a set of features $[f_1^i, ..., f_k^i]$, outputs an estimated label $\hat{l}_i \in \{0, 1\}$, i.e., $\hat{l}_i = f(t_i)$. Note that Subtask A covers two versions: monolingual and multilingual versions.

4.1.2 Subtask B

Similarly, we consider the task as a supervised classification where we aim to learn a function f that, given an input text t_i, represented as a set of features $[f_1^i, ..., f_k^i]$, outputs an estimated label $\hat{l}_i \in \{0, 1, 2, 3, 4, 5\}$, i.e., $\hat{l}_i = f(t_i)$ where 0 refers to the human-written texts and the rests are those generated by different machines, including 1-*ChatGPT*, 2-*cohere*, 3-*davinci*, 4-*bloomz*, and 5-*dolly*, respectively.

Furthermore, we are interested in gaining insights from the classifier's predictions that allow us to understand which features contribute positively to detecting machine-generated text.

4.2 Our architecture

The overall architecture of our proposed approach is demonstrated in Figure 1. The general idea is to use a machine learning model trained to discriminate between text samples generated by a human and text samples generated by LLMs. Different directions could be pursued to extract useful features from a text and perform text classification.

4.2.1 Metric-based models

Inspired the works from He et al. (2023) and Spiegel and Macko (2023), we capture the local information from the texts using the following methods: (1) *Log-Likelihood*, (2) *Rank*, (3) *Log-Rank*, (4) *Entropy*, and (5) *MFDMetric*.

- *Log-Likelihood*: Given a text, we average the token-wise log probability of each word generated from a language model to generate a score for this text.

- *Rank*: For each word in a text, given its previous context, we calculate the absolute rank of this word. Then, for a given text, we compute the score of the text by averaging the rank value of each word.

- *Log-Rank*: Slightly different from the Rank metric that uses the absolute rank, the Log-Rank score is calculated by first applying the log function to the rank value of each word.

- *Entropy*: Similar to the Rank score, the Entropy score of a text is calculated by averaging the entropy value of each word conditioned with its previous context.

- *Multi-Feature Detection Metric* or *MFDMetric*: This is a two-step zero-shot method that (1) considers four distributional information (*Log-Likelihood*, *Log-Rank*, *Entropy*), and statistical information (*LLM-Deviation*) as input features; and (2) classify the text using neural networks.

In *Log-Likelihood*, a larger score denotes the text is more likely to be machine-generated. Meanwhile, in *Rank* and *Log-Rank*, a smaller score denotes the text is more likely to be machine-generated. Similarly, the machine-generated text is more likely to have a lower *Entropy* score. Note that metric-based methods are only applied to Subtask A.

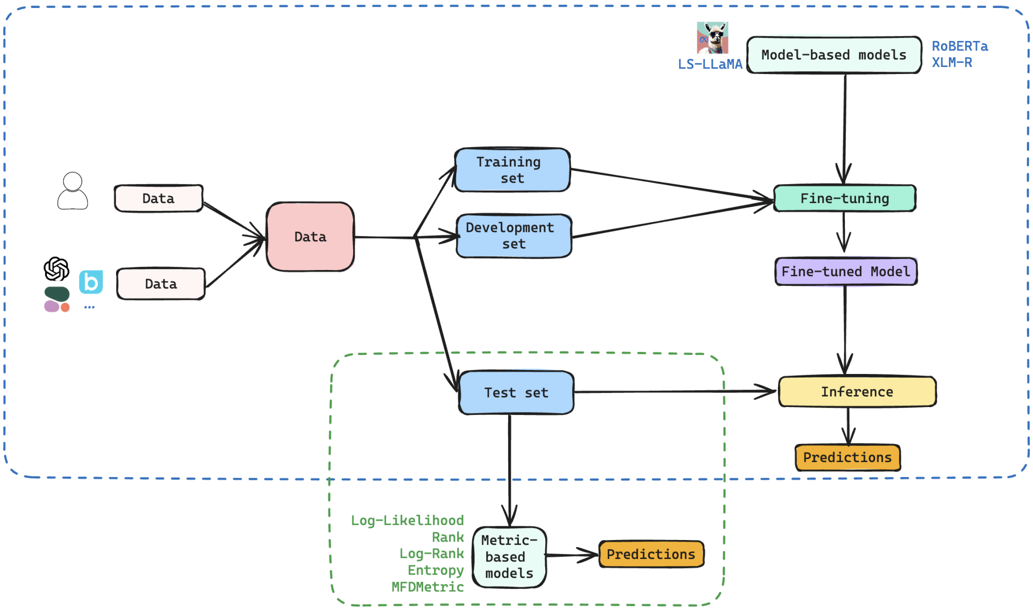

Figure 1: Our general architecture for Subtask A (both blue and green boxes) and Subtask B (only blue box).

4.2.2 Model-based models

LMs Two Transformer-based models have been fine-tuned as sequence classifiers, namely RoBERTa (Liu et al., 2019) and XLM-R (Conneau et al., 2020). RoBERTa is a Transformers model pretrained on a large corpus of English data in a self-supervised fashion using a masked language modeling (MLM) objective. Meanwhile, XLM-R is a multilingual version of RoBERTa that was pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages. These models are also suggested as the baseline methods from *SemEval-2024 Task 8* organizers.

LLMs Given the recent success of the LLMs architectures for solving downstream NLP tasks, we decided to follow the same vein to build our classifier. As such, we start with LLaMA-2 (Touvron et al., 2023), an LLM model pre-trained for the sequence classification task, using its corresponding tokenizer to preprocess data. We then fine-tune the model on the training subset of collected data. Consequently, the fine-tuned model is used for inference on the testing subset. Finally, the obtained classification scores are evaluated against the ground truth.

4.3 Hyperparameters

Metric-based models We took advantage of IMGTB[5] framework with default parameter settings suggested from He et al. (2023) and Spiegel and Macko (2023).

LMs We fine-tuned 2 LMs, namely RoBERTa and XLM-R, using HuggingFace Transformers Pytorch Trainer with the following configuration: batch size = 16, learning rate = 1e-5, weight decay = 0.01, number of epoch = 10.

LLaMA-2 To make the comparison comparable, we fine-tuned LS-LLaMA[6] (version: *LLaMA-2-7b-hf*) using the HuggingFace Transformers PyTorch Trainer class with the same configuration: batch size = 16, learning rate = 1e-5, and the number of epochs = 10 with max length = 256 and Lora = 12.

All the experiments were implemented on an NVIDIA RTX A6000 with CUDA Version of 12.0 and 49140MiB.

4.4 Evaluation metrics

For both Subtasks, we use *Accuracy*, *macro-F1*, and *micro-F1* as the evaluation metrics to measure our classifiers' performance. These are also the standard metrics in *SemEval-2024 Task 8*, which makes our works more comparable with other participants. We assess the performance of the development sets first and apply the best models to the test set. The final leaderboard reported results only for *Accuracy*.

[5] https://github.com/michalspiegel/IMGTB

[6] https://github.com/4AI/LS-LLaMA

Methods	Subtask A - Mono			Subtask A - Multi			Subtask B		
	Accuracy	Micro F1	Macro F1	Accuracy	Micro F1	Macro F1	Accuracy	Micro F1	Macro F1
Metric-based methods									
Log-Likelihood	0.51880	0.40011	0.51880	0.49700	0.46172	0.49700	-	-	-
Rank	0.71760	0.71760	0.71262	0.51000	0.51000	0.47705	-	-	-
Log-Rank	0.51700	0.38751	0.51700	0.49675	0.49675	0.46197	-	-	-
Entropy	0.53880	0.43979	0.53880	0.49475	0.45385	0.49475	-	-	-
MFDMetric	0.65820	0.63645	0.65820	0.49450	0.45875	0.4945	-	-	-
Language model (LM)-based methods - Benchmarks from competition									
RoBERTa	0.65920	0.65920	0.61629	0.49100	0.49100	0.48721	0.73167	0.73167	0.69539
XLM-R	0.75740	0.75740	0.75130	0.52275	0.52275	0.48949	0.60267	0.60267	0.56838
Large language model (LLM)-based methods									
LS-LLaMA$_{2-7b-hf}$	**0.81500**	**0.81500**	**0.80862**	**0.87400**	**0.87400**	**0.87399**	**0.75500**	**0.75500**	**0.73165**

Table 3: Performance of Subtask A (monolingual and multilingual versions) and Subtask B on development set where the training set is split into training and validation set with the ratio of 8:2 for training progress.

5 Results and Discussion

Table 3 demonstrates the evaluation of different methods on the development set before the test set was released, while Table 4 reports our final performance on the test set in comparison with the baseline suggested by *SemEval-2024 Task 8* and our approach ranking on the leaderboard.

Methods	A - Mono	A - Multi	B
Baseline	**0.88466**	0.80887	0.74605
LS-LLaMA$_{2-7b-hf}$	0.85840	**0.92867**	**0.83117**
Our ranking	**25**/125	**5**/62	**6**/70

Table 4: Our performance in *Accuracy* on the test set with the same train-validation-test split of *SemEval Task8*.

5.1 General Observations

We first present different experiment results on the development set in Table 3. We observed that overall, LLM-based methods, such as LS-LLaMA$_{2-7b-hf}$, tend to outperform other approaches across all sequence classification tasks, suggesting the effectiveness of leveraging large pre-trained language models for these tasks. Meanwhile, metric-based methods have varying performance, with *Rank* showing some competitiveness, but generally, they are outperformed by LLM and LM-based methods. Regarding LM-based approaches, XLM-R tends to surpass the performance of RoBERTa in the monolingual version of Subtask A despite RoBERTa being specifically designed for English only.

Based on the performance of the development set, we applied LS-LLaMA$_{2-7b-hf}$, which yields superior performance in these Subtasks compared to other methods, to the test set. As shown in Table 4, despite not surpassing the baseline of Subtask A's monolingual version, our models significantly outperform the baseline of Subtask A's multilingual version and Subtask B with approximately 10% gain on average. While we ranked only 25st over 125 participants in the monolingual version of Subtask A, we demonstrate competitive performance to be ranked 5th over 62 and 6th over 70 participants in the multilingual version of Subtask A and Subtask B, respectively.

We conducted several analyses to investigate how different factors would affect the detection performance of our best classifier.

5.2 Effect of Text Length

We first present the distribution of the number of words (*#. words*) for predicted human-generated and machine-generated texts (*Predictions*) and their ground truth (*GT*) in the dataset in each Subtask (shown in Figure 2).

On ground-truth levels, Figure 2 highlights discrepancies in word distribution between human-written texts and those generated by different LLMs. This is evident in Subtask A by the difference in word count distribution between human and machine-generated labels and in Subtask B by the varying generated performance of individual LLMs compared to human-written ones. For instance, *davinci* can generate long-context answers (more than 2500 words) while others respond in more concise ways (less than 1500 words).

Despite these discrepancies, compared predictions against ground truth, our classifier effectively captures the distribution of generated texts per

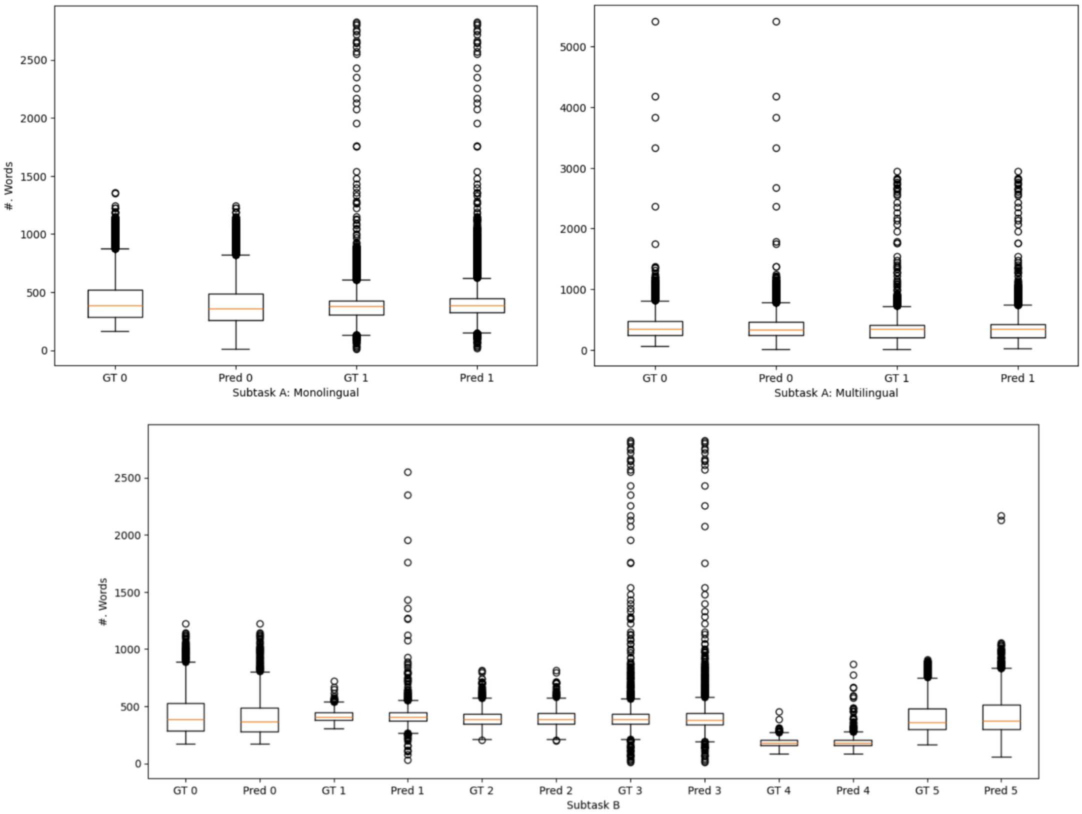

Figure 2: The distribution of words (*#. words*) for human-written and machine-generated texts of our predictions (*Pred*) and the ground truth (*GT*) on different datasets of different tasks (Subtask A: 0-*Human*, 1-*Machine*; Subtask B: 0-*Human*, 1-*ChatGPT*, 2-*cohere*, 3-*davinci*, 4-*bloomz*, and 5-*dolly*).

class, resulting in comparable word distributions between predictions and ground truth except in *ChatGPT* and *dolly* where most of the examples we misclassified are outliers.

5.3 Class-wise Performance

To better investigate the detection performance of different classes, we visualize the normalized confusion matrix of different tasks when we used our LLaMA-2 classifier as shown in Figure 3.

On one hand, in terms of Accuracy, unlike the multilingual version of Subtask A where all the classes can be well detected with up to 94% in Accuracy, the monolingual version suffers significantly from misclassifying human-written texts into machine-generated ones, which reduces the performance of the overall classifier (the accuracy of the human-written class falls into around 76%). Most of the misclassified texts are human-written that our classifier mistakenly took for the machine-generated ones.

On the other hand, when it comes to multi-way machine-generated text classification as Subtask B, the predictive performance of our classifier varies depending on the type of LLMs used to generate texts. Although LLaMA-2 has a good performance in identifying human-written and machine-generated texts generated by *ChatGPT*, *bloomz*, and *dolly*, the performance in attributing machine-generated texts from other LLMs (e.g., *cohere*, and *davinci*) is largely limited. For example, the prediction accuracy of *ChatGPT*, *bloomz* is almost perfect (99.53% and 99.70%, respectively). Meanwhile, that of *cohere* is just above the average (around 60%) and its texts are often misclassified as machine-generated texts from *davinci*, followed by *ChatGPT*. This is expected due to potential overlap in the distribution of the metric among various LLMs, which introduces extra challenges in attribution.

Broadly speaking, our findings suggest that the fine-tuned LLMs (e.g., LLaMA-2) excel in detecting machine-generated multilingual texts and accurately classifying machine-generated texts within a specific category, (e.g., *ChatGPT*, *bloomz*, *dolly*). However, they do exhibit challenges in detecting

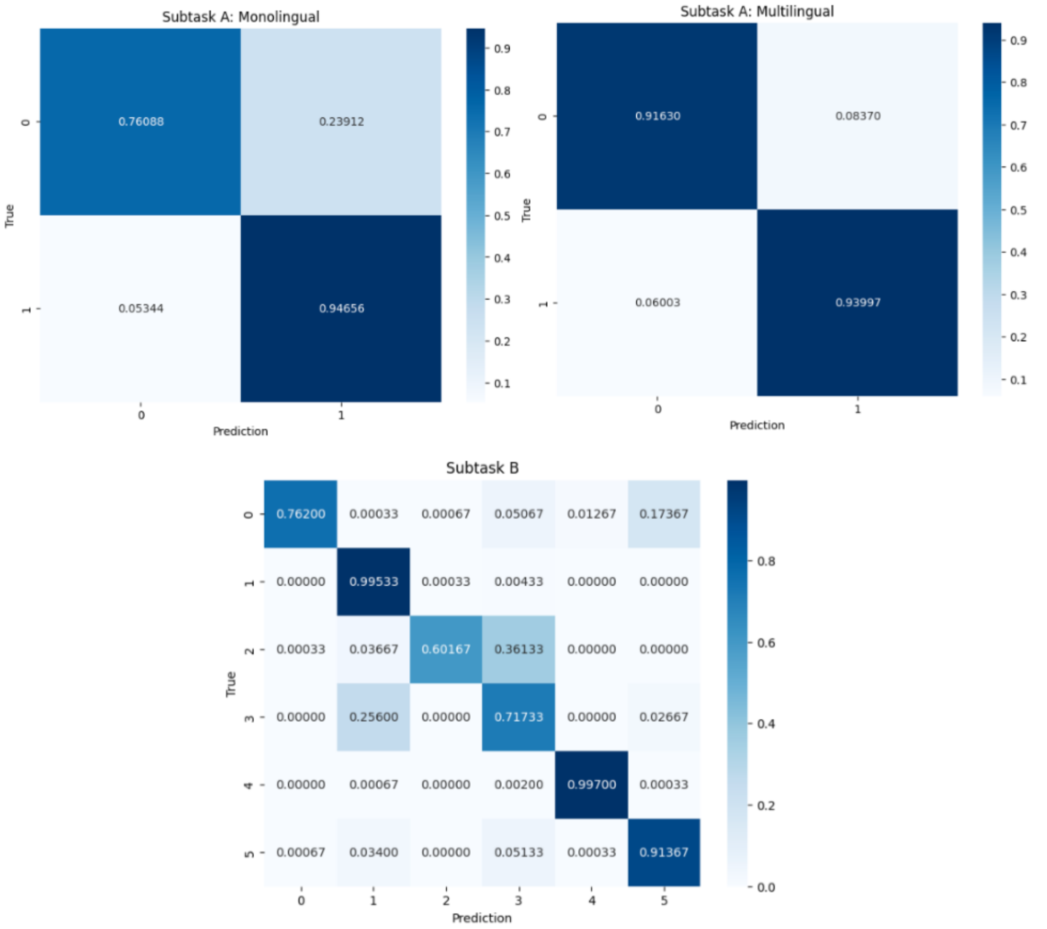

Figure 3: Normalized confusion matrix of LLaMA-2 methods on different tasks. Note that the values in the diagonal represent the class-wise accuracy (Subtask A: 0-*Human*, 1-*Machine*; Subtask B: 0-*Human*, 1-*ChatGPT*, 2-*cohere*, 3-*davinci*, 4-*bloomz*, and 5-*dolly*).

them in other categories (e.g., *cohere*, and *davinci*). Further studies are needed to improve the lower-performing classes.

6 Conclusions

In conclusion, this paper outlines our contribution to the first two Subtasks of *SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection*, namely Monolingual and Multilingual Binary Human-Written vs. Machine-Generated Text Classification and Multi-Way Machine-Generated Text Classification. We conducted a comprehensive comparative study across three methodological groups: Five metric-based models (Log-Likelihood, Rank, Log-Rank, Entropy, and MFD-Metric), two fine-tuned sequence-labeling language models (RoBERTA and XLM-R); and a fine-tuned large-scale language model (LS-LLaMA).

Our findings suggest that our LLM outperformed both traditional sequence-labeling LM benchmarks and metric-based approaches. Furthermore, our fine-tuned classifier excelled in detecting machine-generated multilingual texts and accurately classifying machine-generated texts within a specific category, (e.g., *ChatGPT*, *bloomz*, *dolly*). However, they do exhibit challenges in detecting them in other categories (e.g., *cohere*, and *davinci*). This is due to potential overlap in the distribution of the metric among various LLMs. Overall, we ranked 6^{th} in both Multilingual Binary Human-Written vs. Machine-Generated Text Classification and Multi-Way Machine-Generated Text Classification on the leaderboard.

In future work, we would like to take a step further to evaluate whether our classifier is robust enough against adversarial attacks (e.g., paraphrasing, random spacing, adversarial perturbation) as well as investigate how to make our model more interpretable and explainable, which is important, but insufficiently addressed when detecting machine-generated contents.

Limitations

Regarding specificity and domain dependence, our classifier might not effectively distinguish among different types of machine-generated texts, such as texts generated by different models, for different purposes, or in specific domains (which can be seen in the case of detecting texts generated by *cohere* and *davinci*).

Acknowledgements

The work was partially supported by the Slovenian Research and Innovation Agency (ARIS) core research program Knowledge Technologies (P2-0103) and projects Linguistic Accessibility of Social Assistance Rights in Slovenia (J5-50169) and Embeddings-based techniques for Media Monitoring Applications (L2-50070). The work has also been supported by the ANNA (2019-1R40226) and TERMITRAD (2020-2019-8510010) projects funded by the Nouvelle-Aquitaine Region, France. Besides, the work was supported by the project Cross-lingual and Cross-domain methods for Terminology Extraction and Alignment, a bilateral project funded by the program PROTEUS under the grant number BI-FR/23-24-PROTEUS006.

References

Razvan Azamfirei, Sapna R Kudchadkar, and James Fackler. 2023. Large language models and the perils of their hallucinations. *Critical Care*, 27(1):1–2.

Sameer Badaskar, Sachin Agarwal, and Shilpa Arora. 2008. Identifying real or fake articles: Towards better language modeling. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

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*.

Farrokh Habibzadeh. 2023. Gptzero performance in identifying artificial intelligence-generated medical texts: A preliminary study. *Journal of Korean Medical Science*, 38(38).

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. Mgtbench: Benchmarking machine-generated text detection. *arXiv preprint arXiv:2303.14822*.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2023. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics*, pages 5549–5581. PMLR.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822.

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*.

OpenAI. 2023. Gpt-4 technical report.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *Advances in neural information processing systems*, 34:11054–11070.

Michal Spiegel and Dominik Macko. 2023. Imgtb: A framework for machine-generated text detection benchmarking. *arXiv preprint arXiv:2311.12574*.

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

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. Turingbench: A benchmark environment for turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2001–2016.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2022. Prompting palm for translation: Assessing strategies and performance. *arXiv preprint arXiv:2211.09102*.

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.

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.