L3i++ at GenAI Detection Task 1: Can Label-Supervised LLaMA Detect Machine-Generated Text?

Tien Nam Nguyen [1] and **Hanh Thi Hong Tran**[2]
[1] University of La Rochelle, L3i, La Rochelle, France
[2] Arkhn, Paris, France
tnguye28@univ-lr.fr,
hanh.tran@arkhn.com

Abstract

The widespread use of large language models (LLMs) influences different social media and educational contexts through the overwhelming generated text with a certain degree of coherence. To mitigate their potential misuse, this paper explores the feasibility of finetuning *LLaMA* with label supervision (named *LS-LLaMA*) in unidirectional and bidirectional settings, to discriminate the texts generated by machines and humans in monolingual and multilingual corpora. Our findings show that unidirectional *LS-LLaMA* outperformed the sequence language models as the benchmark by a large margin (up to 7.39 and 5.29 percentage points in F1 increase in monolingual and multilingual corpora, respectively). Our code is publicly available at https://github.com/honghanhh/llama-as-a-judge.

1 Introduction

The blooming of large language models (LLMs) has led to a significant step forward in producing different machine-generated content across diverse channels and platforms (e.g., news, social media, question-answering forums, educational, and even academic contexts). The generated texts become increasingly fluent and coherent with the advent of recent models (e.g., GPT-4o, Claude 3.5). However, this also resulted in concerns regarding their potential misuse, such as spreading misinformation and causing disruptions in the education system. Consequently, there is a need to develop automatic systems to identify machine-generated text to mitigate its potential misuse.

Inspired by the work of Tran et al. (2024), we investigate the feasibility of training a binary sequence classifier that can reliably differentiate between text generated by humans and text that appears human-like but is generated by machines but leverage the performance with the integration of a *LLaMA-as-a-judge* in three different settings on the larger monolingual and multilingual corpora from Wang et al. (2025).

The main contribution of this paper is as follows:

- We study a label-supervised adaptation configuration for *LLaMA-as-a-judge* to discriminate between human-written (HW) and machine-generated (MG) texts.
- We investigate the feasibility of employing latent representations in LLaMA with three settings: masked unidirectional, masked bidirectional, and unmasked ones for discriminant label prediction in the classification tasks.
- Our solution is publicly available on GitHub to encourage openness, transparency, and reproducibility in the research community.

2 Related Work

The success of LLMs in various downstream NLP tasks (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). Researchers have developed several automatic detection methods (Zellers et al., 2019; Uchendu et al., 2021) that can identify the MG texts from the HW texts, which initially can be divided into two categories, i.e., metric-based and model-based methods.

Metric-based methods Metric-based methods leverage LLMs to process the text and extract its distinguishable features. Then, predicted distribution entropy determines whether a text belongs to MG or HW texts. Some metric-based detection methods include Log-Likelihood, Rank, Entropy, GLTR, Log-Rank, and DetectGPT (He et al., 2023), to cite a few.

Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect), pages 155–160
January 19, 2025.

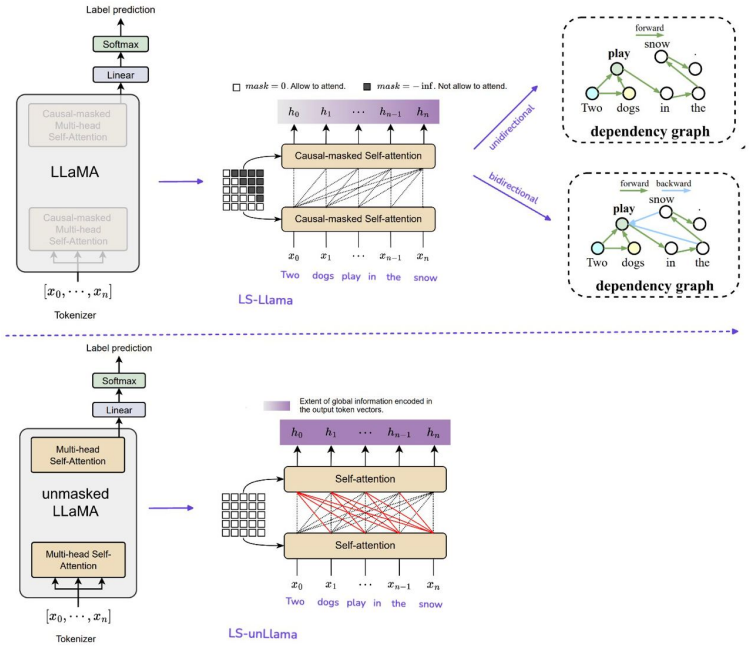


Figure 1: Our general LLaMA architecture in three different settings.

Model-based methods The model-based methods (Habibzadeh, 2023; Guo et al., 2023) are often trained using a corpus that contains both MG and MW texts to make predictions, for example, ChatGPT Detector (Guo et al., 2023), GPTZero (Habibzadeh, 2023), and LM Detector (Ippolito et al., 2020). Regarding Wang et al. (2024b), RoBERTa (Liu et al., 2019) and XLM-R (Conneau et al., 2020) are two baseline language models for these specific tasks.

The comparative studies of both categories can be found at the work of Tran et al. (2024).

3 Data

We evaluate the feasibility of our approach with English[1] and multilingual[2] corpora from Wang et al. (2025). Both corpora are the continuation and improvement of Wang et al. (2024a) with additional training and testing data generated from novel LLMs and including new languages.

4 Methodology

This section tackles the problem by formulating it as supervised sequence classification tasks. We then introduce our proposed architecture and present how we fine-tune them before indicating how we assessed their performance.

[1] Jinyan1/COLING_2025_MGT_en

[2] Jinyan1/COLING_2025_MGT_multingual

4.1 Problem Formulation

We formulate the problem as a binary supervised classification task, whose objective is to learn a mapping between a text representation 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., $\hat{l}_i = f(t_i)$.

4.2 Our architecture

Our general architecture of the label-supervised *LLaMA-as-a-judge* (short form: *LS-LLaMA*[3]) from MG text detection with three different settings is visualized in Figure 1.

4.2.1 Masked Unidirectional LS-LLaMA

The tokens T from the input sequence S were fed into pretrained models to extract the latent representation H from *LLaMA* for sequence classification. First, we compute its embedding:

$$t = Tokenizer(S) \quad (1)$$

then

$$x = Embedding(t) \quad (2)$$

the transformer decoder layers are computed as

$$\text{Attn}_i^{LLaMA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{SoftMax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \mathcal{M}\right)\mathbf{V} \quad (3)$$

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

$$\mathbf{Q} = W_q x + b, \quad \mathbf{K} \quad = W_k x + b, \quad \mathbf{V} = W_v x + b$$

$\mathcal{M}$: denotes the causal mask.

We modify the *LLaMA* model to obtain all the sequence representations:

$$h_{LLaMA} = LLaMA(T) \tag{4}$$

The pooling operation is applied to the latent representation to obtain the vector representation h for sequence classification. After passing through fully connected layers and a softmax layer, vector representation h is mapped to the label space. Cross-entropy loss is calculated based on the output logits and the ground-truth label.

4.2.2 Masked Bidirectional LS-LLaMA

To address the missing dependency information in autoregressive LLMs, we explore how backward dependencies affect sentence embedding learning. This is done by converting certain attention layers in the transformer decoder from unidirectional to bidirectional, removing the causal masks. However, if we keep all the causal masks, performance decreases significantly. Therefore, only the last attention layer is converted to bidirectional.

Mathematically speaking, with input sentence S and its embedding x as computed in *LS-LLaMA*, the embeddings are fed to the transformers to obtain $\overleftrightarrow{\text{LLaMA}}^{1:n}$:

$$\text{Attn}_i^{LLaMA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{SoftMax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \mathcal{M}\right)\mathbf{V} \tag{5}$$

Then, we detach and transform it from uni- to bi-directional to obtain $\overleftrightarrow{\text{BiLLaMA}}^{n-1:n}$

$$\text{Attn}_i^{BiLLaMA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{SoftMax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}})\mathbf{V} \tag{6}$$

The final representation can be formulated as:

$$\mathbf{h} = \overrightarrow{\text{LLaMA}}^{1:n}(\mathbf{x}) + \overleftrightarrow{\text{BiLLaMA}}^{n-1:n}(\mathbf{x}) \tag{7}$$

4.2.3 Unmasked Unidirectional LS-LLaMA

Instead of removing only the causal mask of the last transformer layer, the causal masks will be removed in all transformer layers with the assumption to be replenished in token representations during fine-tuning as all the tokens can attend to each other. The computation of the transformer layer is computed as:

$$\text{Attn}_i^{unLlaMa}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{SoftMax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}})\mathbf{V} \tag{8}$$

Moreover, using bidirectional combining with max-over-time pooling yields better performance than average pooling and last-token pooling in classification tasks. The formula of unmasked unidirectional *LS-LLaMA* can be represented as follows:

$$h_{\text{unLLaMA}} = UnLLaMA(x) \tag{9}$$

without causal masks.

4.3 Hyperparameters

We fine-tuned *LLaMA-2-7b-hf*[4] with the same configuration for all three settings: batch size = 16, learning rate = 1e-5, number of epochs = 5 with max length = 128, and Lora = 12. All the experiments were implemented on an NVIDIA RTX H100 with a CUDA Version of 12.4 (95000MiB).

4.4 Evaluation metrics

We use *Accuracy*, *macro-F1*, and *micro-F1* as the evaluation metrics to measure our classifiers' performance. These are also the standard metrics in Wang et al. (2025), which makes our work more comparable with other solutions.

5 Results

Table 1 and 2 report the evaluation of *LS-LLaMA* with three different learning settings in comparison with the baselines on the monolingual and multilingual subsets, respectively, in the development phase before the test set was released.

Methods	Accuracy	Micro F1	Macro F1
LS-LLaMA	0.9166	0.9166	0.9146
biLS-LLaMA	0.8887	0.8928	0.8928
LS-unLLaMA	0.8725	0.8725	0.8682
Baseline	0.8483	0.8483	0.8407

Table 1: Evaluation on monolingual set in dev. phase.

Overall, *LS-LLaMA* demonstrates strong performance in monolingual and multilingual corpora, particularly excelling in accuracy and micro F1 metrics. However, the significant drop in macro F1 scores for the multilingual evaluation suggests that while the model performs well on average, it may

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

Methods	Accuracy	Micro F1	Macro F1
LS-LLaMA	0.8703	0.8703	0.6715
biLS-LLaMA	0.8514	0.8514	0.6540
LS-unLLaMA	0.8025	0.8025	0.5890
Baseline	0.8561	0.8561	0.6186

Table 2: Evaluation on multilingual set in dev. phase.

have difficulty with less frequent classes, indicating a potential area for improvement in handling multilingual data where there exists an imbalance in HW and MG classes in different languages and resources. While providing some performance, the other models do not surpass *LS-LLaMA*, reinforcing their effectiveness in this evaluation phase.

Based on the subset's performance in the development phase, we applied *LS-LLaMA* to the test set in the test phase, which achieved 0.7463 in macro F1 and 0.7554 in accuracy for the monolingual test set, 0.7427 in macro F1 and 0.744 in accuracy for the multilingual test set.

6 Discussion

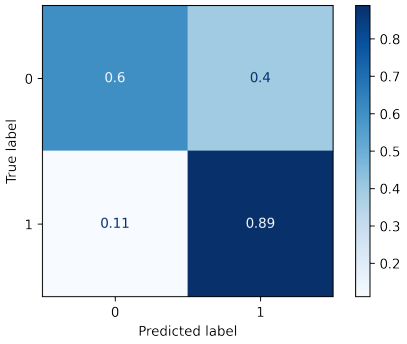
Unidirectional vs. Bidirectional Context. The unidirectional *LS-LLaMA*'s focus on sequential learning, coherence recognition, and specialized training objectives makes it particularly well-suited for the task of MG text detection. In contrast, the bidirectional *LS-LLaMA*, while powerful in capturing overall context, may struggle with the specific sequential dependencies that are critical for effectively distinguishing HW from MG texts. This fundamental difference in architecture and training approach likely contributes to the observed performance advantage of unidirectional *LS-LLaMA*.

Masking Strategy. The "masked" aspect refers to how models are trained to predict missing parts of the input. In unidirectional masked *LS-LLaMA*, the focus is often on learning to predict the next token or fill in gaps based on prior context. This can enhance their ability to understand coherent patterns typical in HW texts, which explains the higher performance of masked *LS-LLaMA* compared to unmasked settings, which can potentially suffer from data leaks.

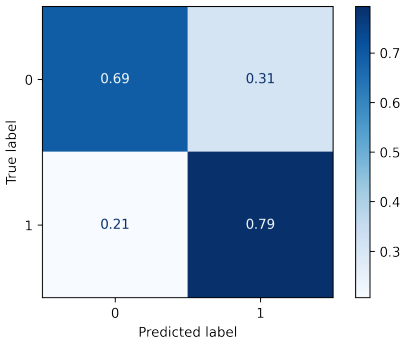
7 Error Analysis

We conduct several analyses to investigate how different factors would affect the detection performance of our best classifier, namely *LS-LLaMA*.

Figure 2 illustrates the confusion matrices for the English and multilingual test sets. These matrices reveal a notable tendency for higher error rates in detecting MG content. This observation suggests the model may be calibrated to prioritize detecting MG (label 1) instead of HW texts (label 0).



(a) The monolingual test set.



(b) The multilingual test set.

Figure 2: Confusion matrices for *LS-LLaMA*.

We elaborated our analysis regarding the error rate by text length and textual analysis of misclassification. The results suggest that error rates may not consistently increase or decrease with longer or shorter texts; instead, they vary based on data. However, there is a tendency for the classifier to have higher errors when the length of the text is from 10,000 to 20,000 words (see Figure 3).

8 Conclusions

In conclusion, we conducted a comparative study of label supervision *LLaMA*, so-called *LS-LLaMA* to highlight the potential and feasibility of fine-tuning an LLM to discriminate between HW and MG texts. Three different settings have been applied, including unidirectional masked, unidirectional unmasked, and bidirectional. Our findings suggest that unidirectional masked *LS-LLaMA* outperformed two other settings and the benchmarks for both monolingual and multilingual sets.

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.

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*.

Hanh Thi Hong Tran, Tien Nam Nguyen, Antoine Doucet, and Senja Pollak. 2024. L3i++ at semeval-2024 task 8: Can fine-tuned large language model detect multigenerator, multidomain, and multilingual black-box machine-generated text? In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 13–21.

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, and Thomas Arnold. 2024a. 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 2057–2079, Mexico City, Mexico. Association for Computational Linguistics.

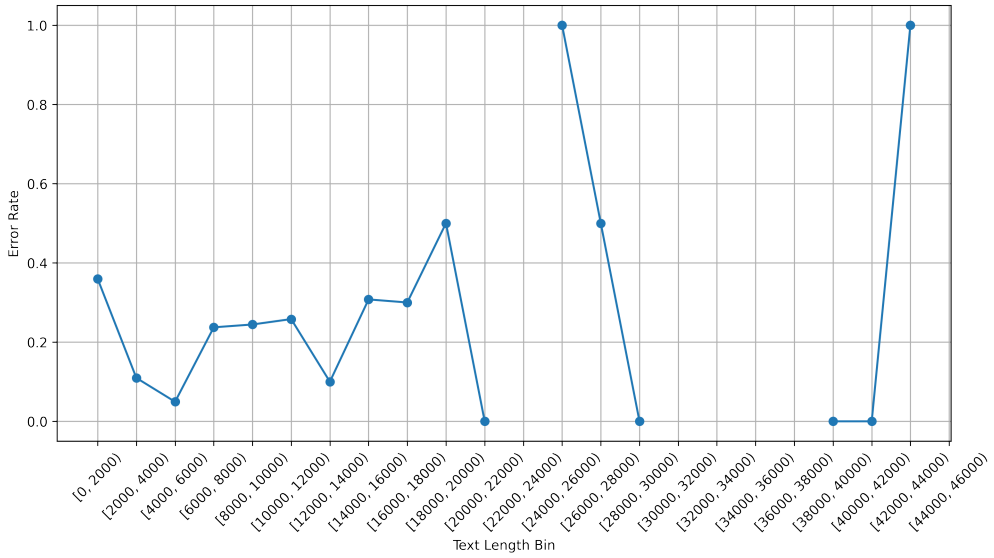
Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. Semeval-2024 task 8: Multidomain, multimodel and multilingual 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, Artem Shelmanov, Jonibek Mansurov, Akim Tsvigun, Vladislav Mikhailov, Rui Xing, Zhuohan Xie, Jiahui Geng, Giovanni Puccetti, Ekaterina Artemova, Jinyan Su, Minh Ngoc Ta, Mervat Abassy, Kareem Elozeiri, Saad El Dine Ahmed, Maiya Goloburda, Tarek Mahmoud, Raj Vardhan Tomar, Alexander Aziz, Nurkhan Laiyk, Osama Mohammed Afzal, Ryuto Koike, Masahiro Kaneko, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2025. Genai content detection task 1: English and multilingual machine-generated text detection: Ai vs. human. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, Abu Dhabi, UAE. Association for Computational Linguistics.

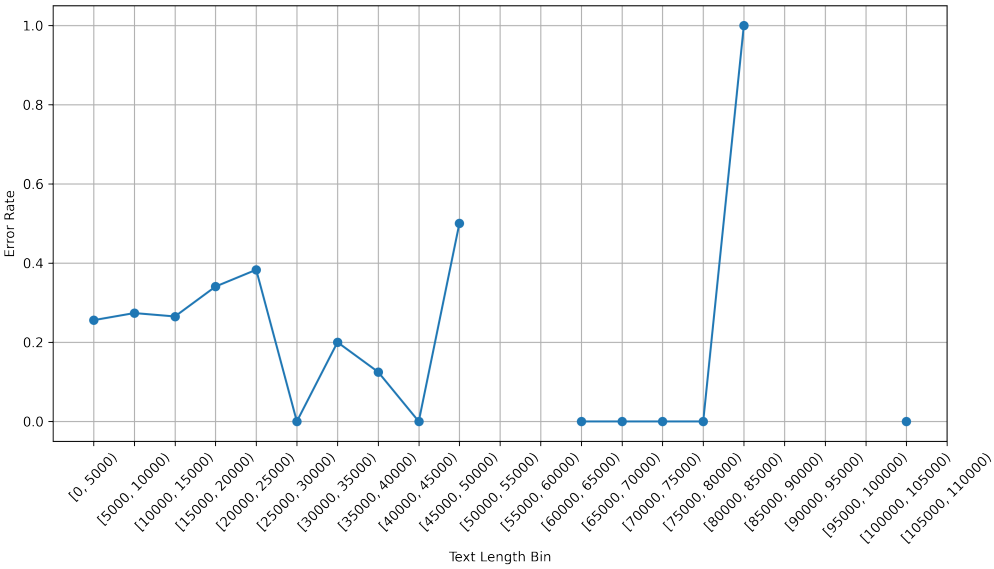
Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

A Error Rate by Text Length

Figure 3a and 3b show fluctuating error rates across text lengths of the English set and multilingual test set, we can not see a clear linear relationship between text length and error rate. This suggests that errors may not consistently increase or decrease with longer or shorter texts; instead, they vary based on data. However, there is a tendency for the classifier to have higher errors when the length of the text is from 10,000 to 20,000 words



(a) The monolingual test set.



(b) The multilingual test set.

Figure 3: Error rates based on text length using *LS-LLaMA*.