Genaios at SemEval-2024 Task 8: Detecting Machine-Generated Text by Mixing Language Model Probabilistic Features

Areg Mikael Sarvazyan and **José Ángel González** and **Marc Franco-Salvador**
Genaios, Valencia, Spain
{areg.sarvazyan, jose.gonzalez, marc.franco}@genaios.ai

Abstract

This paper describes the participation of the Genaios team in the monolingual track of Subtask A at SemEval-2024 Task 8. Our best system, LLMIXTIC, is a Transformer Encoder that mixes token-level probabilistic features extracted from four `LLaMA-2` models. We obtained the best results in the official ranking (96.88% accuracy), showing a false positive ratio of 4.38% and a false negative ratio of 1.97% on the test set. We further study LLMIXTIC through ablation, probabilistic, and attention analyses, finding that (i) performance improves as more LLMs and probabilistic features are included, (ii) LLMIXTIC puts most attention on the features of the last tokens, (iii) it fails on samples where human text probabilities become consistently higher than for generated text, and (iv) LLMIXTIC's false negatives exhibit a bias towards text with newlines.

1 Introduction

The analysis of Machine-Generated Text (MGT) has gained popularity in recent times. This is important for detecting and attributing text to Large Language Models (LLMs) such as LLaMA (Touvron et al., 2023) and GPT (Ouyang et al., 2022), and combating fake-news, intellectual property violations (Henderson et al., 2023), data leakages (Nasr et al., 2023), among other malicious usages (Kasneci et al., 2023). Recent efforts include zero-shot (Bao et al., 2024) and supervised systems (Wang et al., 2023). However, large-scale scenarios that combine domains, data sources, or models are still challenging (Sarvazyan et al., 2023b; Eloundou et al., 2023). As a result, different frameworks to generate high-quality MGT datasets[1] (Sarvazyan et al., 2024) and evaluation campaigns have been released (Shamardina et al., 2022; Sarvazyan et al., 2023a). In this paper, we describe our solution as the Genaios team at SemEval-2024 Task 8: *Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection* (Wang et al., 2024a).

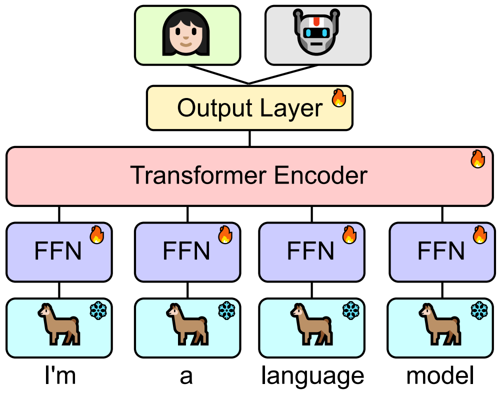

Figure 1: Overview of the proposed system. Modules marked with ❄ are frozen. Those with 🔥 are trainable.

Our starting point is the observation that LLMs assign higher probabilities to MGT than to human text. We propose LLMIXTIC, illustrated in Figure 1, which leverages this via a Transformer encoder (Vaswani et al., 2017) that mixes token-level probabilistic features extracted from four `LLaMA-2` models, both instructed and base flavors: `LLaMA-2-7b`, `LLaMA-2-7b-chat`, `LLaMA-2-13b`, and `LLaMA-2-13b-chat`. For each token, our features are (i) the log probability of the observed token, (ii) the log probability of the predicted token, and (iii) the entropy of the distribution.

These probabilistic features capture MGT style in a precise manner, favouring detection. As a result, we obtained the best results in the official ranking (96.88% accuracy) for the monolingual track of Subtask A: *Binary Human-Written vs. Machine-Generated Text Classification*. Our analysis shows that performance improves as more LLMs and probabilistic features are used. In addition, LLMIXTIC pays more attention to the last tokens of the sequence, where higher probabilities for human texts lead to misclassifications. Finally,

[1] One of these is TextMachina, freely available at https://github.com/Genaios/TextMachina

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 101–107

Split	Label	Model	arXiv	PeerRead	Reddit	WikiHow	Wikipedia	Outfox (Test)
Train	👩		15.5	2.4	15.5	15.5	14.5	16.2
	🤖	Bloomz	-	-	-	-	-	3
		Cohere	3	2.3	3	3	2.3	3
		ChatGPT	3	2.3	3	3	3	3
		Davinci	3	2.3	3	3	3	3
		Dolly	3	2.3	3	3	2.7	3
		GPT4	-	-	-	-	-	3
Dev	👩		0.5	0.5	0.5	0.5	0.5	-
	🤖	Bloomz	0.5	0.5	0.5	0.5	0.5	-

Table 1: Statistics of the Subtask A Monolingual dataset by split, label, model, and domain. Sizes in thousands.

texts with newlines are predominant among false negatives.

2 Background

The monolingual track of Subtask A: *Binary Human-Written vs. Machine-Generated Text Classification* focuses on detecting whether an English text is entirely written by a 👩 **human** or generated by an 🤖 **LLM**. The data is an extension of the M4 dataset (Wang et al., 2024b) and combines texts from different domains and LLMs. We show the statistics of the dataset in Table 1. The official evaluation metric of the Subtask A is accuracy, which we also employ in our experiments.

3 System Overview

It is known that high-quality human text does not follow high-probability distributions over the next tokens (Holtzman et al., 2020). In contrast, LLMs are decoded to sample from regions of high probability, thus assigning higher probability to low-diversity constructions and lower to human texts. In practice, this causes MGT to be measurably different from human texts, e.g., showing less idiomatic expressions, scarce and repetitive discourse markers, or strictly complying with canonical orderings of constituents (Simón et al., 2023).

We developed our system by following these previous findings, and considering that most of the current LLMs share two key components which condition the probability distributions they learn: (i) the underlying backbone, namely Transformer decoder, with few architectural changes and (ii) large portions of their training data both for pre-training and instruction tuning. Our system relies on the hypothesis that token-level probabilistic features extracted from an specific set of LLMs can be used to differentiate human texts and MGT from a potentially different set of LLMs, which has been shown to be very effective in existing MGT detectors (Przybyła et al., 2023; Wang et al., 2023).

As depicted in Figure 1, our final system is a Transformer Encoder that mixes token-level probabilistic features extracted from four `LLaMA-2` models (Touvron et al., 2023), including base and instructed versions: `Llama-2-7b`, `Llama-2-7b-chat`, `Llama-2-13b`, and `Llama-2-13b-chat`. Following (Przybyła et al., 2023), we build feature sequences where each token is represented as the concatenation of three probabilistic features extracted from each LLM. Specifically, we employ the following features.

Log probability of the predicted token. Measures the highest probability assigned by θ to the next token as:

$$\alpha_i = \max_{y \in \mathcal{V}} \log p_\theta(y|x_{<i}) \tag{1}$$

Entropy of the distribution. Measures the uncertainty of θ for choosing the next token:

$$\beta_i = -\sum_{y \in \mathcal{V}} p_\theta(y|x_{<i}) \log p_\theta(y|x_{<i}) \tag{2}$$

Log probability of the observed token. Measures how likely is the observed token x_i according to the model θ and the prefix $x_{<i}$ as:

$$\gamma_i = \log p_\theta(x_i|x_{<i}) \tag{3}$$

Given a text $x = [x_1, ..., x_n]$ and a set of LLMs $\mathcal{L} = \{\theta_1, ..., \theta_m\}$, we represent x as a feature sequence $h = [h_1, \ldots, h_n]$ with each h_i denoting the probabilistic features from all the LLMs for the i-th token, $h_i = [\alpha_i^1; \beta_i^1; \gamma_i^1, \ldots, \alpha_i^m; \beta_i^m; \gamma_i^m]$. For instance, our final system uses four LLMs and three features from each one, $h \in \mathbb{R}^{n \times 12}$. Note that the features are extracted per-token, which constrains us to use LLMs with a shared tokenizer.

The feature vectors in h are projected to 128 dimensions through a feed-forward layer, and then mixed with a Transformer encoder of 1 layer and 4 attention heads. The output of the Transformer layer is averaged and a softmax layer is used to compute a probability distribution over the human and generated classes. This classifier on top of the probabilistic features, LLMixtic's only trainable component, is comprised of solely 85k parameters, being 0.0002% of the total.

4 Experimentation

We focus on the monolingual track of Subtask A, carrying out comparisons among models and ablations of the best system. For these we employ the original training and validation splits provided by the organizers. In the post-evaluation stage, we analyze the errors of LLMixtic in the test set by inspecting the probabilistic features extracted from `LLaMa-2`, the learned attention heads, and text patterns in the misclassified samples.

4.1 Model Comparison

We compare LLMixtic with classical and neural models, while also evaluating different LLMs to extract the probabilistic features. All the models in these comparisons are trained and evaluated on the original training and validation splits provided by the shared task organizers.

Classical baselines. We consider a Logistic Regression classifier, using either TF-IDF features with word n-grams ranging from 1 to 3-grams (`LR+TFIDF`), or readability features (`LR+READ`). For these, we employ `scikit-learn` (Pedregosa et al., 2011) and `readability`,[2] training the model with balanced class weights and default parameters.

Neural baselines. We also compare LLMixtic with two fully fine-tuned Transformer encoders, `roberta-base` (Liu et al., 2019) and `e5-base` (Wang et al., 2022). These models are trained for four epochs, using the cross-entropy loss, a batch size of 32 samples, and a learning rate of 5e-6.

LLMixtic's LLMs. We evaluate LLMixtic with probabilistic features from two LLM families, namely `GPT-2` (Radford et al., 2019; Sanh et al., 2019) and `LLaMA-2` (Touvron et al., 2023). For the `GPT-2` family,[3] we include `gpt2`, `gpt2-medium`, and `distillgpt2`. The `LLaMA-2` family is comprised of `LLaMA-2-7b`, `LLaMA-2-7b-chat`, `LLaMA-2-13b`, and `LLaMA-2-13b-chat`. These are trained for ten epochs, with a maximum text length of 512 tokens, a batch size of 32 samples, a learning rate of 1e-3, and the cross-entropy loss.

All neural models are trained with HuggingFace's Trainer (Wolf et al., 2020) in FP16 mode, employing early stopping, with a patience of 3 evaluation steps, on the validation set. The LLMs used for feature extraction are always frozen, with `LLaMA-2` models also being quantized to 8 bits. We implement LLMixtic in PyTorch (Paszke et al., 2019), and run all the experiments using a single NVIDIA RTX A6000.

Model	Accuracy (%)
`LR+READ`	42.32
`LR+TFIDF`	61.26
`roberta-base`	80.58
`e5-base`	74.48
LLMixtic (w/ `GPT-2`)	67.42
LLMixtic (w/ `LLaMA-2`)	**85.98**

Table 2: Model comparison results on the dev set.

Results are presented in Table 2. Here we observe how LLMixtic using `LLaMA-2` features outperforms every baseline by large margins, improving upon the best baseline's score by 5 points in accuracy, while having only 0.07% relative training parameters. Notably, all the neural models outperform classical baselines, which suggests that grammatical features, especially those based on readability measures, are not enough to properly discriminate between human-written and generated text. Also, the usage of probabilistic features from `GPT-2` models does not yield good results in comparison to neural baselines and LLMixtic with `LLaMA-2` LLMs. This suggests that the scale of the LLM used to extract features could have a large impact on the results. Considering that the `LLaMA-2` family is more similar than `GPT-2` models to the LLMs that generated the text of the dataset, we also hypothesize that using feature extraction LLMs that more closely resemble the LLMs in the dataset can yield better results.

4.2 LLM and Feature Ablations

We study the impact the number of LLMs and probabilistic features have on LLMixtic's performance by means of two ablation studies: at LLM and at feature level. These experiments are performed with the same experimental setup: first training with a single LLM or feature, and continually adding the other LLMs or features.

Ablation results are presented in table 3. In LLM ablation we observe improvements as more LLMs are included. Notably, the inclusion of `chat` models provides the largest improvements of up to ten points. Building upon our hypothesis about similarities in architecture, training strategies, and datasets

[2] https://github.com/andreasvc/readability/

[3] Chosen for its success in previous shared tasks (Przybyła et al., 2023) and to test for more efficient feature extractors.

Ablation	Configuration	Accuracy (%)
LLMs	`LLaMA-v2-7b`	74.90
	+ `LLaMA-v2-13b`	75.86
	+ `LLaMA-v2-7b-chat`	78.48
	+ `LLaMA-v2-13b-chat`	**85.98**
Features	Predicted	79.40
	+ Entropy	83.26
	+ Observed	**85.98**

Table 3: Ablation study over LLMs and features.

of instruction-tuned LLMs, it is expected that most of them, especially the `chat` models we used, have learned close distributions. Therefore, we consider that this improvement can be explained by the nature of the dataset, where all the generators were instruction tuned. We also note that LLMIXTIC with only non-instructed LLMs achieves similar results to one of the neural baselines, outperforming LLMIXTIC with `GPT-2` by a large margin.

Similar to the LLM ablation, feature ablation results improve as more features are included, achieving an increment of more than six points when all the features are used. We observe that LLMIXTIC obtains similar performance to the best neural baseline just using the log probability of the predicted token and outperforms it after adding the entropy of the distribution. Besides, only with one feature, the performance is ten points higher than LLMIXTIC with `GPT-2` using all the features.

5 Results

Our official submission is LLMIXTIC with `LLaMA-2`, trained on the training and validation sets, using the previously described experimental setting. Table 4 presents the results obtained by our system, where it reaches an accuracy of 96.88%, surpassing the other participants' approaches and ranking first. Due to time constraints, we focused our participation on the monolingual track. However, having seen the performance of LLMIXTIC on the test set of the monolingual track, we trained LLMIXTIC under the same setting for the multilingual track in a post-deadline stage (denoted in tables with *). Here, we obtained an accuracy of 89.97%, which would have placed us at 14th position.

6 Analysis

We further analyze the behavior of LLMIXTIC in the test set by examining the probabilistic features extracted from `LLaMa-2`, the learned attention heads, and patterns in misclassified samples.

Track	Rank	Name	Accuracy (%)
Monolingual	**1**	**Genaios**	**96.88**
	2	USTC-BUPT	96.09
	20	*baseline*	88.46
		(119 more)	
Multilingual	1	USTC-BUPT	95.98
	14*	**Genaios**	**89.97**
	25	*baseline*	80.88
		(44 more)	

Table 4: Final results on the official ranking. Bold denotes our team's placement.

LLMIXTIC fails when human text probabilities become larger than for generated texts. In contrast, LLMIXTIC works better when the generated text probabilities are consistently larger than those from human texts. To illustrate this behavior, Figure 2 shows each LLM's feature averaged both for correct and erroneous predicted samples. Errors occur with unusually high values of α and γ features in the human class, and unusually low values for the generated class. The effect of feature β is also notable, with the margin between human and generated curves being smaller in misclassifications. Additionally, for each class, chat and base models reveal different curves for all three features.

LLMIXTIC pays more attention to the last positions. Figure 3 shows the average of the attention heads across all the samples to illustrate it. This behavior could be the main cause of errors when human text probabilities become consistently larger than those for generated texts in the last positions, as shown in Figure 2. A diagonal pattern with high probability is also noticeable until approximately position 150, after which it disappears.

Human text is more often confused with generated text than vice versa. There are twice as many false positives as there are false negatives (714 vs. 355). This translates into a false positive rate of 4.38% and a false negative rate of 1.97%.

Newlines are predominant in false negatives. We manually analyze the errors with higher confidence, finding that most of LLMIXTIC's false negatives include `\n` to separate sentences or paragraphs, while false positives do not, to the same extent. Specifically, `\n` is present in 75.49% of false negatives, whereas it is only present in 34.59% of false positives. This difference could suggest (i) a potential bias in the training data, with human texts containing more `\n` than the generated texts,

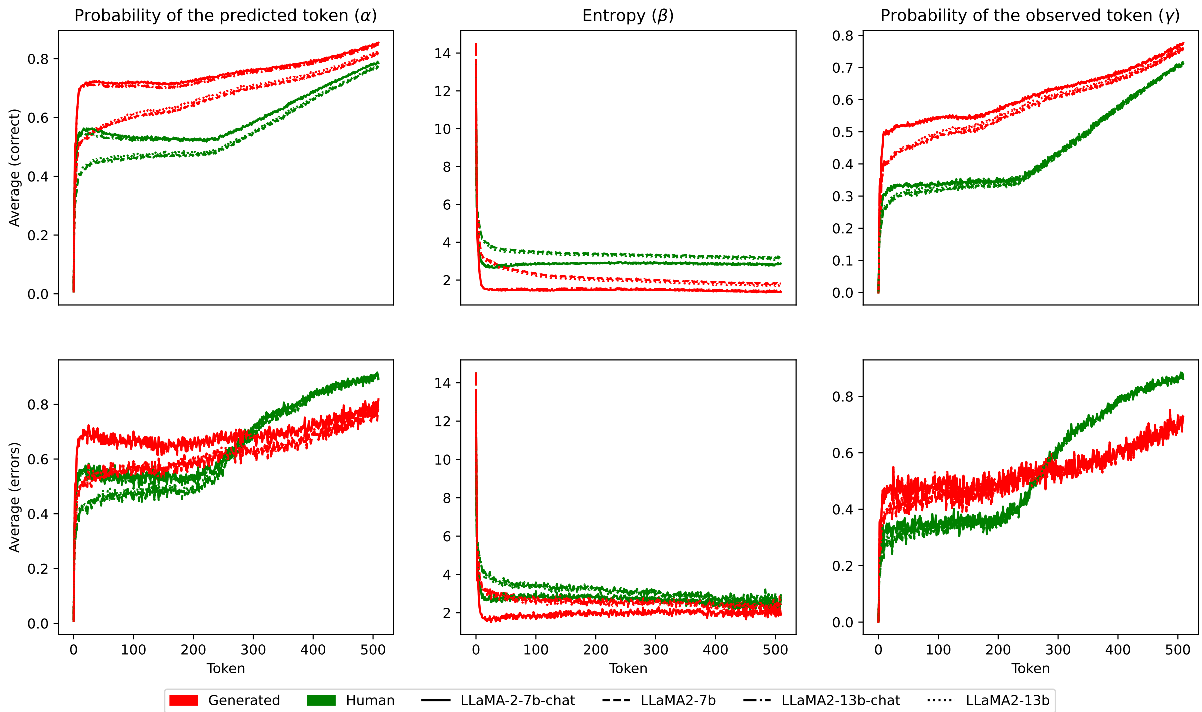

Figure 2: Sample-averaged probabilistic features of the four `LLaMa-2` models, for the two classes (generated and human). Both for correct predictions (top row) and errors (bottom row). The y axis denotes the average of the probabilistic feature (α, β, or γ) across all samples of a label in the test set, at a given position marked on the x axis. Throughout all positions, the probabilities of generated text for correct predictions consistently exceed those of humans. However, for errors, human probabilities surpass those of generated text from the middle of the sequences.

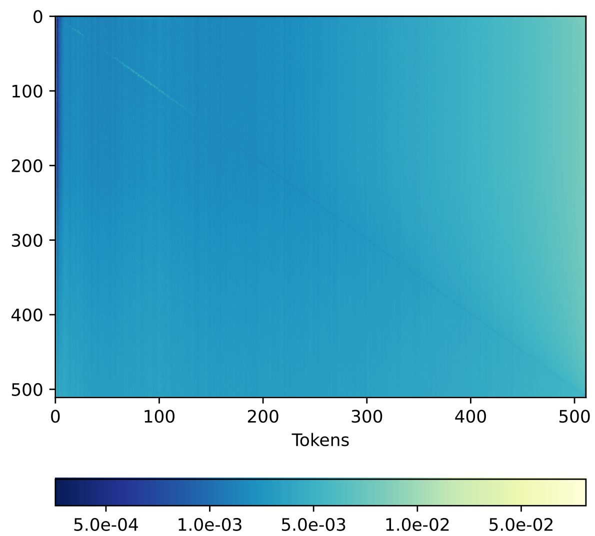

Figure 3: Sample-averaged and head-averaged attention scores from LLMIXTIC's Transformer encoder. LLMIXTIC pays more attention to the last positions.

or (ii) our system is learning a spurious correlation between \n and the human class.

7 Conclusion

We described the participation of the Genaios team in the monolingual track of Subtask A at SemEval-2024 Task 8. We proposed LLMIXTIC, a Transformer Encoder that mixes token-level probabilistic features extracted from four base and instructed `LLaMA-2` models, namely `LLaMA-2-7b`, `LLaMA-2-7b-chat`, `LLaMA-2-13b`, and `LLaMA-2-13b-chat`. Our system obtained the best results in the official ranking, with small false positive and false negative ratios.

Our ablation analyses showed that LLMIXTIC's performance improves as more LLMs and probabilistic features are used. We compared these features across correctly predicted and misclassified samples, finding that LLMIXTIC works better when MGT probabilities are consistently higher than for human text. In addition, attentions are mostly focused on the last tokens, which could be one of the causes of the errors made by LLMIXTIC. Finally, the newline character seems predominant in false negatives but not in false positives, which suggests biases either in the data or in our model.

Aiming to foster R&D in this area, future works will focus on TextMachina,[1] a framework to generate MGT datasets for tasks such the ones addressed in this SemEval shared task: detection, attribution, boundary, and mixcase detection.

References

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations*.

Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. 2023. Gpts are gpts: An early look at the labor market impact potential of large language models. *arXiv preprint arXiv:2303.10130*.

Peter Henderson, Xuechen Li, Dan Jurafsky, Tatsunori Hashimoto, Mark A Lemley, and Percy Liang. 2023. Foundation models and fair use. *arXiv preprint arXiv:2303.15715*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, et al. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, page 102274.

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*.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

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

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Piotr Przybyła, Nicolau Duran-Silva, and Santiago Egea-Gómez. 2023. I've seen things you machines wouldn't believe: Measuring content predictability to identify automatically-generated text. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2023). CEUR Workshop Proceedings, CEUR-WS, Jaén, Spain*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC^2 Workshop*.

Areg Mikael Sarvazyan, José Ángel González, and Marc Franco-Salvador. 2024. TextMachina: Seamless Generation of Machine-Generated Text Datasets. *arXiv preprint arXiv:2401.03946*.

Areg Mikael Sarvazyan, José Ángel González, Marc Franco-Salvador, Francisco Rangel, Berta Chulvi, and Paolo Rosso. 2023a. Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains. *Sociedad Española de Procesamiento del Languaje Natural (SEPLN)*, 71:275–288.

Areg Mikael Sarvazyan, José Ángel González, Marc Franco-Salvador, and Paolo Rosso. 2023b. Supervised machine-generated text detectors: Family and scale matters. In *Information Access Evaluation meets Multilinguality, Multimodality, and Visualization*. Springer International Publishing.

Tatiana Shamardina, Vladislav Mikhailov, Daniil Cherniavskii, Alena Fenogenova, Marat Saidov, Anastasiya Valeeva, Tatiana Shavrina, Ivan Smurov, Elena Tutubalina, and Ekaterina Artemova. 2022. Findings of the the ruatd shared task 2022 on artificial text detection in russian. *arXiv preprint arXiv:2206.01583*.

Lara Alonso Simón, José Antonio Gonzalo Gimeno, Ana María Fernández-Pampillón Cesteros, Marianela Fernández Trinidad, and María Victoria Escandell Vidal. 2023. Using linguistic knowledge for automated text identification. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2023). CEUR Workshop Proceedings, CEUR-WS, Jaén, Spain*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023. SeqXGPT: Sentence-level AI-generated text detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1144–1156, Singapore. 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. 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 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, Malta.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.