TrustAI at SemEval-2024 Task 8: A Comprehensive Analysis of Multi-domain Machine Generated Text Detection Techniques

Ashok Urlana Aditya Saibewar Bala Mallikarjunarao Garlapati
Charaka Vinayak Kumar Ajeet Kumar Singh Srinivasa Rao Chalamala
TCS Research, Hyderabad, India
ashok.urlana@tcs.com, aditya.saibewar@tcs.com, balamallikarjuna.g@tcs.com
charaka.v@tcs.com, ajeetk.singh1@tcs.com, chalamala.srao@tcs.com

Abstract

The Large Language Models (LLMs) exhibit remarkable ability to generate fluent content across a wide spectrum of user queries. However, this capability has raised concerns regarding misinformation and personal information leakage. In this paper, we present our methods for the SemEval2024 Task8, aiming to detect machine-generated text across various domains in both mono-lingual and multi-lingual contexts. Our study comprehensively analyzes various methods to detect machine-generated text, including statistical, neural, and pre-trained model approaches. We also detail our experimental setup and perform a in-depth error analysis to evaluate the effectiveness of these methods. Our methods obtain an accuracy of 86.9% on the test set of subtask-A mono and 83.7% for subtask-B. Furthermore, we also highlight the challenges and essential factors for consideration in future studies.

1 Introduction

Recent advancements in Large Language Models (LLMs) have facilitated a wide range of applications, notably in content generation (Chung et al., 2023). While LLMs offer creative and informative content generation capabilities, concerns such as misinformation, fake news, personal information leakage, legal and ethical issues have emerged (Chen and Shu, 2023; Li, 2023; Kim et al., 2023). Consequently, detecting machine-generated text has become a crucial task to address these aforementioned challenges.

The identification of machine-generated text is still an open challenge because of its overlapping similarities with human-written text. The current text generation models produce text that is strikingly similar to human language in terms of grammaticality, coherency, fluency, and utilization of real-world knowledge (Radford et al., 2019; Zellers et al., 2019; Brown et al., 2020). However, variations in sentence length, the presence of noisy data, and the generation of incomplete sentences are common indicators of machine-generated text.

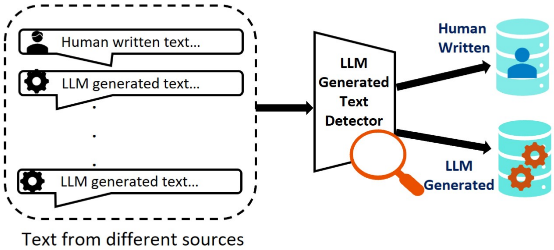

Figure 1: Block diagram for machine-generated text detection.

1.1 Essence of LLM generated text detection

LLMs' open-ended text generation techniques have sparked various concerns across domains (Jo et al., 2023). It has been demonstrated that LLMs have the potential to generate misinformation and fake news (Chen and Shu, 2023), which can be catastrophic in healthcare (Zhou et al., 2023), public safety, education, and finance. Moreover, LLMs can generate text without source attribution, raising the risk of plagiarism (Quidwai et al., 2023), and can include legal and ethical concerns too (Li, 2023).

Furthermore, when LLMs are used in enterprise applications there can be concerns of intellectual property rights infringement (Zhao et al., 2024) such as generated content might contain trademarks or branding elements (Ren et al., 2024). Lastly, LLMs can aggravate security concerns by generating phishing emails (Bethany et al., 2024), fake reviews (Adelani et al., 2020), hallucinations (Huang et al., 2023), biased content (Fang et al., 2023; Dai et al., 2024), and personal information leakage (Kim et al., 2023).

1.2 Tasks

The main objective of the competition is to differentiate text based on the source of its generation

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 927–934
June 20-21, 2024 ©2024 Association for Computational Linguistics

method (see Figure 1), with specific importance given to machine-generated text and human-written texts (Wang et al., 2024a). The competition consists of three tasks Subtask A, Subtask B, and Subtask C. Our study focuses on Subtasks A and B.

Subtask A. Binary Human-Written vs. Machine-Generated Text Classification: This task aims to distinguish between human-written or machine-generated text. This task acts as a binary classification. Subtask A is again subdivided into the following two categories. *Mono-lingual:* The text is in the English language. *Multi-lingual:* The text is in English, Chinese, Russian, Urdu, Indonesian, Arabic, and Bulgarian languages.

Subtask B. Multi-Way Machine-Generated Text Classification: This task aims to classify the given text into six distinct classes, which are 'human', 'chatGPT', 'cohere', 'davinci', 'bloomz', 'dolly' with each class representing the source of its generation. This task acts as a multi-class classification.

The key contributions of this work include, 1) We present a comprehensive analysis of various machine-generated text detection techniques for multi-domain mono and multi-lingual data, 2) We provide a detailed experimental setup for statistical, neural, and pre-trained models along with corresponding error analysis, 3) We emphasize the discussions and future perspectives derived from the findings of the study.

2 Related Work

Recent works on LLM-generated[1] text detection has shown promising results. Statistical methods are used to detect the LLM-generated text by utilizing the entropy (Shen et al., 2023), and N-gram frequency (Tassopoulou et al., 2021). Some other studies uses the fact that language models assign high probability for the repeated sentences which is often AI model generated and ranks the AI model generated sentence Krishna et al. (2022). In a study, OpenAI has trained a classifier to detect LLM-generated text using the RoBERTa-based model (Solaiman et al., 2019).

Some of the widely-used methods adopted the GPT detectors such as OpenAI detection classifier[2], GPTZero[3], and ZeroGPT[4]. Another variant is DetectGPT (Mitchell et al., 2023), which works on the assumption of LLM-generated text lies in the negative curvature region of the log-likelihood. Using this approach, DetectGPT perturbs the input text using masked language models, such as BERT (Devlin et al., 2018), BART (Lewis et al., 2019), T5 (Raffel et al., 2019) and compare the log probability of the text and masked filled variants. Similarly, few works utilized the different decoding strategies including top-k, nucleus, and temperature sampling to generate the text from GPT2 and BERT based models employed to perform binary classification to label text as human-written or machine generated (Ippolito et al., 2020).

Recently, watermarking methods have been used in enterprises to protect the intellectual properties and fair use of the generation models. However these techniques simplify the detection of the LLM-generated output text by synonym replacement over generated outputs and text level posthoc lexical substitutions (Li et al., 2023; Sadasivan et al., 2023), and soft watermarking was introduced in (Kirchenbauer et al., 2023) using green and red token lists. Hidden space operations were also introduced by injecting secret signals into the probability vector of each target token (Zhao et al., 2023).

Bhattacharjee and Liu (2023) proposed a method which triggers when the text has common words randomly assembled as it is easier to find than identifying unique and rare tokens. Sadasivan et al. (2023) focused on zero-shot AI text detection by using two clusters depending on watermarked or not. Another study (Wang et al., 2024c), proposed a benchmark framework consists of an input module, a detection module and an evaluation module for machine generated text detection against human-written text. In contrast to existing works, this study presents the multi-domain multi-lingual machine generated text detection techniques.

3 Datasets

This section given an overview of the dataset utilized and the corresponding analysis.

3.1 Source and acquisition

The task organizers provided the dataset[5] for all the tasks (§1.2). The dataset is an extension of the M4 dataset (Wang et al., 2024b). The dataset provided for this task consists of machine-generated text and human-written text. The human-written text is gathered from various sources such as

[1]We interchangeably use the terms 'LLM-generated' or 'machine-generated'

[2]https://platform.openai.com/ai-text-classifier

[3]https://gptzero.me/

[4]https://www.zerogpt.com/

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

	Subtask - A (Mono-lingual)			Subtask - A (Multi-lingual)			Subtask - B		
	Train	Development	Test	Train	Development	Test	Train	Development	Test
# Samples	**119757**	**5000**	**34272**	**172417**	**4000**	**42378**	**71027**	**3000**	**18000**
# Avg sentences	23	17	18	19	10	17	18	12	18
# Minimum sentences	1	1	1	0	1	1	1	1	1
# Maximum sentences	1583	699	882	1583	59	882	699	477	882
# Median sentences	14	9	18	12	10	17	12	10	17
# Avg words	530	394	437	445	222	396	398	267	414
# Minimum words	2	7	12	0	41	12	6	7	12
# Maximum words	38070	19115	2946	38070	2081	6308	19115	1484	2946
# Median words	319	213	424	296	218	379	290	217	413

Table 1: SemEval 2024 Task 8 data statistics.

Wikipedia, WikiHow (Koupaee and Wang, 2018), arXiv, and PeerRead (Kang et al., 2018), Reddit (Fan et al., 2019) for English, Baike and Web question answering (QA) for Chinese, news for Urdu, news for Indonesian and RuATD (Shamardina et al., 2022) for Russian. On the other hand, the machine-generated text is gathered by prompting different multi-lingual LLMs: ChatGPT (Achiam et al., 2023), BLOOMz (Muennighoff et al., 2023), textdavinci-003, FlanT5 (Chung et al., 2022), Cohere, Dolly-v2, and LLaMa (Touvron et al., 2023).

3.2 Exploratory data analysis

Preliminary analysis of data is a crucial step that is required to understand the dataset characteristics. We have observed that the number of sentences in each task data varies from 1 to a few hundred. Particularly, a few samples in the multi-lingual training data consist of empty samples as well. Another point to note is, that the number of sentences in the multi-lingual train and development varies a lot, which indicates the dataset obtained from different sources. There are a few cases, where some of the samples consist of more than 38k tokens in a single sample. With these observations, to experiment on cleaned data, we employ two types of pre-processing settings. The former (Version-1) applies heuristic-based pre-processing and sub-word removal, whereas the latter (Version-2) applies only heuristic-based pre-processing. We reported the detailed analysis of the dataset statistics in Table 1.

4 System Overview

This section offers various approaches employed to perform machine-generated text identification. Our approaches are categorized into 1) statistical, 2) neural, and 3) pre-trained models.

4.1 Methodology

4.1.1 Statistical methods

To understand the effectiveness of statistical models, we experimented with a wide range of statistical models and their variants including ensemble approaches. The statistical models including Logistic Regression (LR), SVM, MLP, LightGBM and some of the ensemble models detailed in Table 3.

4.1.2 Neural methods

Neural networks have demonstrated remarkable success in various domains, from image and speech recognition to natural language processing. We experiment with Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and their combinations. We utilize FastText [(Joulin et al., 2016), (Bojanowski et al., 2017)] embeddings to capture hierarchical patterns within the text data.

4.1.3 Pre-trained models

Self-supervised pre-trained models have been effective for the classification tasks. In this study, we experiment with a wide range of pre-trained models trained on either open-source or language model-generated data. The pretrained models including BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), DistilRoBERTa base (Sanh et al., 2019), RoBERTa Base OpenAI Detector (Solaiman et al., 2019), XLM RoBERTa (Conneau et al., 2019).

4.2 Experimental setup

For all the experiments, we have utilized the default data splits provided by the task organizers. For all the statistical models, four types of embeddings were employed namely counter vectors,

Models	Subtask - A (Monolingual)				Subtask - A (Multilingual)				Subtask - B			
	Count	Word	N-gram	Character	Count	Word	N-gram	Character	Count	Word	N-gram	Character
LR	0.544	0.566	**0.712**	0.615	0.511	0.516	0.498	0.561	0.544	0.514	0.519	0.558
Naive Bayes	0.506	0.520	0.568	0.599	0.510	0.515	0.489	0.509	0.463	0.533	0.495	0.354
SVM	0.534	0.573	0.708	0.634	0.344	0.494	0.512	0.571	0.569	0.550	0.518	0.573
Random Forest	0.576	0.614	0.619	0.682	0.465	0.517	0.504	0.559	0.579	0.462	0.429	0.408
XG Boost	0.584	0.623	0.639	-	0.499	0.507	0.558	-	0.605	**0.619**	0.591	-
MLP	0.594	0.604	0.683	0.647	0.544	0.528	0.485	**0.609**	0.529	0.506	0.493	0.583

Table 2: Accuracy of statistical models development set; LR refers to Logistic Regression, Subtask-B deals with multi-class classification task.

Model	Subtask-A (Monolingual)	Subtask-B
Naive Bayes + SGDClassifier + LightGBM	0.714	0.708

Table 3: Ensemble model Accuracy scores on development set.

Model	Subtask-A		Subtask-B
	Mono	Multi	
CNN + FastText	0.711	0.545	0.652
RNN + LSTM + FastText	0.682	0.615	0.549
Bidirectional RNN + FastText	0.689	0.579	0.582

Table 4: Accuracy of neural models on development set.

Task	Model	Accuracy
Subtask-A (Mono)	BERT Base	**0.825**
	BERT Base_v1	0.807
	BERT Base_v2	0.813
	BERT Base_v2	0.809
	RoBERTa Base OpenAI Detector	0.766
Subtask-A (Multi)	BERT Multilingual Base_v2	0.622
	XLM-RoBERTa	**0.766**
	BERT Multilingual Base	0.622
Subtask-B	RoBERTa Large	0.751
	RoBERTa Base OpenAI Detector	**0.753**
	DistilRoBERTa Base	0.733

Table 5: Pre-trained models Accuracy scores on development set; Where v1 and v2 indicates different preprocessing strategies.

word, n-gram, character-level TF-IDF vectors and spaCy embeddings. Moreover, we used the default configurations mentioned in the scikit-learn[6]. Whereas for pre-trained models the list of hyperparameters details are listed in Table 6. We have not performed any hyperparameter-tuning for our experiments. We conduct most of our experiments using four Nvidia GeForce RTX 2080 Ti (11GB) GPUs. To evaluate all the models, we reported the 'Accuracy' scores.

5 Results and Analysis

This section provides a detailed analysis of the models utilized for subtasks A and B. Our experiments aim to showcase the effectiveness of several machine-generated text detection techniques.

5.1 Subtask A Mono-lingual

We experiment with the statistical and neural models to perform subtasks A and B. All the statistical and ensemble models experimental results on development data are mentioned in Table 2 and Table 3. The results on test data mentioned in Table 7. In the case of statistical models, Logistic Regression obtains the superior performance of 71.2% accuracy using n-gram level TF-IDF embeddings compared to other methods on the development dataset. Whereas in the case of the performance of the test set, our ensemble surpass all the remaining models. We built the ensemble model by creating a custom tokenizer by combining spaCy embedding and TF-IDF with n-gram level range of (3-5) embedding. Moreover, we trained an ensemble model with Naive Bayes, SGDClassifier[7], and LightGBM models which gave 86.9% accuracy on the test set. We experiment with a few neural models with fastText embeddings and out of them CNN+fastText outperforms the other models. We have listed results in Table 4. Moving ahead, we fine-tuned transformer-based pre-trained language models like RoBERTa Base OpenAI detector (Solaiman et al., 2019), which gave 76.6% accuracy on the development set and 78.7% accuracy on test set, BERT base model which gave 82.5% accuracy on the development set and 71.7 % accuracy on test set. The results are detailed in Table 5. Furthermore, we use the pre-processing steps discussed in Section 3.2.

[6]https://scikit-learn.org/stable/supervised_learning.html

[7]https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html

Model	Batch size	Epochs	Vocab size
BERT Base	16	10	30522
OpenAI Detector	16	10	50265
BERT Multilingual Base	8	3	30522
XLM-RoBERTa	8	5	250002
RoBERTa Large	4	2	50265
DistilRoBERTa Base	16	10	29409

Table 6: Experimental setup for pre-trained models. For all the models max source length set to 512 and learning rate $5e^{-5}$.

Fine-tuned the BERT base model with version-1's pre-processed data gave 80.7% on the development dataset and 71.7% on the test set. Then we fine-tuned the BERT base model with version-2 pre-processed data gave 81.3% on the development dataset and 69.7% on the test set. We secured 24^{th} rank out of 137 participants.

We observed that statistical models that performed modestly on the development set generalized effectively to the test set, whereas some pre-trained language models, despite performing well on the development set, struggled to generalize on test set. This discrepancy may stem from the differing sources of the training and development sets ('arxiv', 'reddit', 'wikihow', 'wikipedia', 'peerread') compared to the test set, potentially causing over-fitting of the pre-trained models on the training data and hindering their performance on the test set.

5.2 Subtask A Multi-lingual

For subtask A multi-lingual, we fine-tuned BERT Multilingual Base and XLM RoBERTa base models. BERT Multilingual Base along with version-2 pre-processed data resulted in 62.2% accuracy on the development set and 73.8% accuracy on the test set. Moreover, despite the decent performance of XLM-RoBERTa on the development set with 76.6% accuracy, the performance of on test set is sub-par. Furthermore, the BERT Multilingual base gave 62.2% accuracy on the development set and 73.1% accuracy on the test set. As mentioned in Section 3.2, we observed that, the multi-lingual data consists of empty samples. Hence, we fine-tuned the BERT Multilingual Base model on the version-2 of the pre-processed data, which helped in improving the accuracy of the test set even if we had the same accuracy on development set.

5.3 Subtask B

Subtask B deals with multi-class classification task. For this task, we have conducted experiments using

Task	Model	Accuracy
Subtask - A (Mono)	Baseline	0.74
	Naive bayes + SGDClassifier + LightGBM*	**0.869**
	RoBERTa Base OpenAI Detector	0.787
	BERT Base_v1	0.717
	BERT Base	0.715
	BERT Base_v2	0.697
Subtask - A (Multi)	Baseline	0.72
	BERT Multilingual Base_v2	**0.738**
	BERT Multilingual Base	0.731
	XLM-RoBERTa *	0.50
Subtask - B	Baseline	0.75
	RoBERTa Base OpenAI Detector	**0.837**
	DistilRoBERTa Base*	0.791
	Naive bayes + SGDClassifier+ LightGBM	0.650

Table 7: Test set accuracy results; *entries are the official submission models of the competition.

the statistical models as well as the pre-trained language models. MLP model gave the best accuracy on the development set with 60.9% accuracy. Our ensemble approach obtains 70.8% accuracy on the development set and 65% accuracy on the test set. Moreover, we experimented with RoBERTa Base OpenAI Detector gave 75.3% on the development set and 83.7% accuracy on the test set. Whereas, the DistilRoBERTa base obtains 73.3% accuracy on the development set and 79.1% accuracy on the test set and secured 17^{th} rank out of 86 participants.

6 Conclusions

The study explores different methodologies for detecting machine-generation text, leveraging statistical, neural, and pre-trained models. We observe that the ensemble models are more effective in classifying the mono-lingual data (Subtask-A mono), while models trained on GPT2-text surpass other models in multi-class classification.

7 Limitations

In our study, due to computational constraints, we have not performed experiments with any large language models. Current evaluation has been limited to conventional ML and pre-tained language models. Some of our experimental methods perform better on development data, where as there is a significant drop on test data, this may result in lack of generalization.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. 2020. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection. In *Advanced Information Networking and Applications*, pages 1341–1354, Cham. Springer International Publishing.

Mazal Bethany, Athanasios Galiopoulos, Emet Bethany, Mohammad Bahrami Karkevandi, Nishant Vishwamitra, and Peyman Najafirad. 2024. Large language model lateral spear phishing: A comparative study in large-scale organizational settings.

Amrita Bhattacharjee and Huan Liu. 2023. Fighting fire with fire: Can chatgpt detect ai-generated text?

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

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.

Canyu Chen and Kai Shu. 2023. Combating misinformation in the age of llms: Opportunities and challenges.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

John Chung, Ece Kamar, and Saleema Amershi. 2023. Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

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. *CoRR*, abs/1911.02116.

Sunhao Dai, Yuqi Zhou, Liang Pang, Weihao Liu, Xiaolin Hu, Yong Liu, Xiao Zhang, Gang Wang, and Jun Xu. 2024. Llms may dominate information access: Neural retrievers are biased towards llm-generated texts.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Xiao Fang, Shangkun Che, Minjia Mao, Hongzhe Zhang, Ming Zhao, and Xiaohang Zhao. 2023. Bias of ai-generated content: An examination of news produced by large language models.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

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, Online. Association for Computational Linguistics.

Eunkyung Jo, Daniel A Epstein, Hyunhoon Jung, and Young-Ho Kim. 2023. Understanding the benefits and challenges of deploying conversational ai leveraging large language models for public health intervention. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–16.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. 2023. Propile: Probing privacy leakage in large language models.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *CoRR*, abs/1810.09305.

Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. 2022. RankGen: Improving text generation with large ranking models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 199–232, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Zihao Li. 2023. The dark side of chatgpt: Legal and ethical challenges from stochastic parrots and hallucination.

Zongjie Li, Chaozheng Wang, Shuai Wang, and Cuiyun Gao. 2023. Protecting intellectual property of large language model-based code generation apis via watermarks. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23, page 2336–2350, New York, NY, USA. Association for Computing Machinery.

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. *CoRR*, abs/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. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Ali Quidwai, Chunhui Li, and Parijat Dube. 2023. Beyond black box AI generated plagiarism detection: From sentence to document level. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 727–735, Toronto, Canada. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Jie Ren, Han Xu, Pengfei He, Yingqian Cui, Shenglai Zeng, Jiankun Zhang, Hongzhi Wen, Jiayuan Ding, Hui Liu, Yi Chang, and Jiliang Tang. 2024. Copyright protection in generative ai: A technical perspective.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected?

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.

Tatiana Shamardina, Vladislav Mikhailov, Daniil Cherniansk ii, 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. In *Computational Linguistics and Intellectual Technologies*. RSUH.

Lujia Shen, Xuhong Zhang, Shouling Ji, Yuwen Pu, Chunpeng Ge, Xing Yang, and Yanghe Feng. 2023. Textdefense: Adversarial text detection based on word importance entropy.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models.

V. Tassopoulou, G. Retsinas, and P. Maragos. 2021. Enhancing handwritten text recognition with n-gram sequence decomposition and multitask learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10555–10560, Los Alamitos, CA, USA. IEEE Computer Society.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

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.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Akim Tsvigun, Jinyan Su, Artem Shelmanov, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024c. MG-Bench: Evaluation benchmark for black-box machine-generated text detection.

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.

Jiawei Zhao, Kejiang Chen, Xiaojian Yuan, Yuang Qi, Weiming Zhang, and Nenghai Yu. 2024. Silent guardian: Protecting text from malicious exploitation by large language models.

Xuandong Zhao, Yu-Xiang Wang, and Lei Li. 2023. Protecting language generation models via invisible watermarking. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Jiawei Zhou, Yixuan Zhang, Qianni Luo, Andrea G Parker, and Munmun De Choudhury. 2023. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.