Team ap-team at PAN: LLM Adapters for Various Datasets

Notebook for the PAN Lab at CLEF 2024

Galina Boeva[1,†], German Gritsai[1,2,†] and Andrey Grabovoy[1]

[1]Advacheck OÜ, Tallinn, Estonia
[2]Université Grenoble Alpes (UGA), Grenoble, France

Abstract

The recent breakthrough in text generation ensures that the quality level of generation increases with each new model. On the other hand, the task associated with the use of generated text is relevant. Spreading false information, spamming, generating scientific articles and texts are all problems that have arisen from this outburst. Binary text classification methods have been proposed to control the situation. This research provides an approach based on aggregating QLoRA adapters which are trained for multiple distributions of generative model families. Our method LAVA (LLM Adapters for Various dAtasets) demonstrates comparable results with the primary baseline provided by the PAN organizers. The proposed method provides an efficient and fast detector with high performance of the target metrics, in view of the possibility of parallel training of adapters for the language models. It makes detecting process straightforward and flexible to tailor the adapter to appearing distributions and add it to an existing approach. Furthermore, each learns dependencies separately from the others, after which the outputs are aggregated.

Keywords

natural language processing, large language models, machine-generated text, lora adapter

1. Introduction

Novelty advances in text generation include the development of machine and deep learning approaches and models. The main direction of growth is the modernization of existing techniques based on Transformers [1], since they are able to identify dependencies within a sequence. To build a meaningful model, not only the architecture itself is important, but many other factors are also crucial. The approaches well-known nowadays, such as ChatGPT [2], Google Bard [3], Jasper [4], YaGPT [5], GigaChat [6], have been trained on vast amounts of data to study the dependencies of tokens from related fields. These approaches are used widely and assist people in writing code, generating text, answering questions, and a host of other functions.

Furthermore, due to training on large numbers of data, significant generalizing ability and identify various patterns in the data. Such knowledge could greatly help in solving the problem of classifying generated texts [7, 8, 9]. However, training large models each time on huge datasets is not necessarily feasible due to resource and time constraints. Therefore, in this study, we propose an approach with training several QLoRA adapters [10, 11] for multiple data types thereby submitting our solution to the Voight-Kampff Generative AI Authorship Verification competition by PAN [12, 13, 14]. Most of the previously described methods mainly use a classifier that is trained on a single dataset. However, if we desire to capture as many features of different language models as possible, we need to extend the dataset with examples of each, but at some point this becomes inefficient. Having numerous examples to train a single model will trigger the “forgetting" effect inherent in Transformer architecture models due to the limited number of parameters. To solve this problem, we divided the training examples into families and collected separate datasets for them. One QLoRA adapter will be trained on each such dataset, which will then be combined with the others when aggregating the results. Therefore, by

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

† Authors contributed equally

boyeva@advacheck.com (G. Boeva); gritsai@advacheck.com (G. Gritsai); grabovoy@advacheck.com (A. Grabovoy)
0000-0002-4031-0025 (A. Grabovoy)

teaching adapters these steps, we want to learn the distributions of the model family and the relevant topics.

Contribution

- The evolution of artificial text detection methods is analyzed;
- Multiple datasets from different families of generation models are collected, each of which contains texts written by machine and human on the topics specified in the competition description;
- Implemented a basic approach that is trained on the dataset we gathered;
- An approach is proposed that includes the use of QLoRA adapters to pre-trained model on newly collected data that correspond to different families of generated texts.

2. Background

The field based on the detection of artificial texts is developing quite rapidly. This section presents the main approaches that are currently used for detection, as well as basic models with significant features.

Basic approaches for detection The paper [15] is based on the assessment of various syntactic, semantic and empirical characteristics in order to improve traditional language models. In this work, the substantial part was the compilation of artificial texts based on trigrams, and AdaBoost [16] was also used as a classification model. The authors of [17] consider the problem of recognizing fake news regarding linguistic features. Fake content has a number of characteristics related to syntactic, lexical and semantic levels, which in turn creates a number of features. For example, the calculation of punctuation types and the readability index were used. Moreover, more sophisticated methods are used to determine dynamic and local characteristics in a sequence of tokens. One such article [18] combines a convolutional neural network [19] and a bidirectional LSTM [20] to obtain representations to which MLP is applied for final classification. This vector representation becomes more meaningful by concatenating different pairs of speaker attributes (a dataset on political statements is viewed). Attributes are the features that have been considered in the dataset LIAR [21]. The merging took place after each individual attribute of the speaker's profile was passed through a different layer.

Attention-based approaches Recently, the main direction of generation development has been the use of attention-based models [1]. These approaches are decent at identifying dependencies within a sequence of tokens. When we classify texts, an essential component is the creation of vector representations that will fully describe the input sequence. To create vector representations, BERT-like [22] models are often used, which capture the context clearly. In the article [23], the authors mention the problem of collecting global information on the dictionary, so they propose combining BERT and the convolutional network of the dictionary graph, which helps to remove both local and global changes. What is more, there are also various approaches to decoding tokens and sampling methods. A comparison of three popular sampling-based decoding strategies such as untruncated random sampling, nucleus sampling and topk was conducted by the authors of the article [24]. This study examines the ability of discriminators to correctly distinguish texts created by machines from human-written ones.

Limitations of attention-based models A well-known limitation of transformers is their inability to work with long sequences. This problem raises the topic of the dependence of the detection quality on the length of the input. In article [25], a comparative analysis of the quality of discriminators when training models with the same parameters but different context lengths is carried out in order to identify the importance of context length and its influence on the quality of detection. There is an optimal length for different languages, which may vary, but at the same time there is a certain level that allows to maintain the quality of classification. All the same classifiers based on Transformer architecture

become especially vulnerable to adversarial attacks [26]. That is, the input data of a machine learning model is maliciously manipulated to force it to make incorrect predictions.

3. System Overview

3.1. Problem statement

Let $D = \{(x_i, y_i)\}_{i=1}^{N}$, N is the number of elements in the dataset D. Each pair (x_i, y_i) consists of a text x_i, i.e. a sequence of tokens, and its corresponding label y_i. This label indicates whether the text has been artificially generated or not. Let $x_i = \{p_1, \ldots, p_{M_i}\}$ and $p_j \in L$, where L describes all possible tokens corresponding to the language, M_i — the number of tokens in one text, this parameter can be different for each text.

Let's set the model $g : D \to Y$, where $Y = \{0, 1\}$ then the task is to find the binary classifier that minimizes an empirical risk on the dataset:

$$f = \arg\min_{g} \sum_{(x_i, y_i) \in D} [g(x_i) \neq y_i].$$

3.2. Methods

3.2.1. Basic approach

For a more complete study of the problem of classifying machine-generated texts, we first considered several basic approaches. It was decided to give the result of one of them — large language model Miscrosoft Phi-2 [27] with OneClassSVM [28]. This method is based on the research of features describing the characteristics of human-written texts. These features are taken from LLM output, by splitting the text into tokens and getting logits from the model.

Here is a list of the significant features:

- median of the entropy vector;
- median of the surprise vector;
- average number of characters per word;
- standard deviation of the surprise vector;
- 5th percentile of the entropy vector;
- 5th percentile of the surprise vector;
- maximum of the surprise vector.

After collecting features with the assistance of a language model then they are fed to the OneClassSVM algorithm input, which learns the distribution of such data. Also, during the inference the model will better understand that machine-generated texts do not come from a similar to human distribution.

The main results for this approach are presented in Table 1. This algorithm training stage was conducted on our collected dataset, that will be described in Section 3.2.2. The test stage was considered on the dataset with news that was provided by the competition organizers for research. It can be seen that this method does not show the best result, but it allows us to evaluate the ability to extract useful features using large language models. We cannot compare the quality of the models presented by the organizers as a baseline solution and our baseline solution, due to different datasets for testing.

One important note, LLM+OneClassSVM method was not used as our final solution at the PAN competition. Here we have tested the hypothesis of quality of statistics from large language models for artificial text detection metrics.

The table also shows the model performance for a different number of features to reveal the change in the quality of classification of artificial texts. The number of signs varies in this table. For the 3-feature approach, we considered median of the entropy vector, 5th percentile of the entropy vector and average number of characters per word. When using 4 features (our best result), median of the surprise vector was added. For the third model with 5 features, standard deviation of the surprise vector was added.

Table 1
The dependence of quality for the basic approach based on LLM and OneClassSVM with different number of features. The best results are indicated in bold, the second-best ones are underlined.

LLM+OneClassSVM	ROC-AUC	Brier	F1
3 features	0.7707	0.3958	0.7300
4 features	**0.7971**	**0.3452**	**0.7725**
5 features	<u>0.7745</u>	<u>0.3903</u>	<u>0.7347</u>
7 features	0.7306	0.4712	0.6612

3.2.2. LAVA

In the context of the competition, the data provided is one of the main components. Only a limited news dataset was ensured by the PAN organizers. Given its small size, we decided to collect our own dataset for training. We did not generate anything ourselves, but only searched among open sources for datasets on relevant topics specified in the description of the competition. To apply the idea of training multiple adapters, we split this set into several:

- Dataset A with texts from the GPT family models (GPT-3, GPT-3.5, GPT-4);
- Dataset B with texts from the LLama and Mistral family models (Llama-2, Llama-2, Mistral-v0.1, Mistral-v0.2);
- Dataset C with a small number of texts from more different models (Vicuna, OPT, BLOOM, Alpaca, Gemini Pro);

For training, the idea of adaptation, i.e. reusing the same base model with different adapters for different problems, was utilized. We are working with an upgraded version of LoRA [29] (Figure 1) — QLoRA [30], which quantizes the precision of the weight parameters in the pre-trained LLM with a precision of up to 4 bits. So, our main approach is to train lightweight adapters for the Mistral-v0.2 language model, where each adapter is trained on a separate dataset containing examples from different distributions. Thus, we have three adapters for one language model.

The inference stage was conducted using a competition system. Having trained the adapters, we aggregate them to improve the performance of the resulting model. Combining, merging and measuring the weights of three adapters did not result in metrics increase, but the combination of answers from each of them contributed to the high performance of the target metrics. If all three adapters in the example predict class 0 (human-written), only then do we put class 0 in the final markup, otherwise - 1 (machine-generated). This way of aggregation is necessary to get the best accuracy in detecting human texts. If none of the trained adapters tends to select a generated label, it means that the text is more than likely written by a human.

By training multiple adapters, we reward the model with high generalization capability since each adapter captures the dependencies of each distribution. We consider the Mistral [31] approach, as it shows itself perfectly in solving different tasks [32]. The idea of lightweight adapters helps us avoid learning again while we maintain knowledge of the model. As new model families become available, we can simply add a new trained adapter to the current ones and take it into account when aggregating.

4. Results

Table 3 shows the results, initially pre-filled with the official baselines provided by the PAN organizers and summary statistics of all submissions to the task (i.e., the maximum, median, minimum, and 95-th, 75-th, and 25-th percentiles over all submissions to the task).

Table 2 shows the summarized results averaged (arithmetic mean) over 10 variants of the test dataset. Each dataset variant applies one potential technique to measure the robustness of authorship verification approaches, e.g., switching the text encoding, translating the text, switching the domain, manual obfuscation by humans, etc.

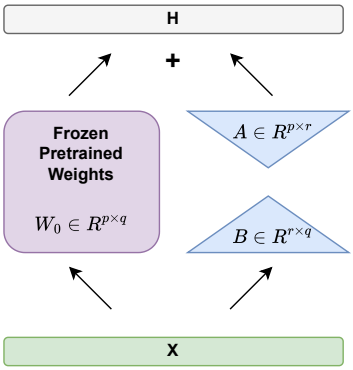

Figure 1: Description of the idea of LoRA Adapter. Let $W_{Updated} = W_0 + \Delta W$, where ΔW contains information about how much we want to update the original weights. For computational learning efficiency, the ΔW matrix is decomposed into two smaller matrices A and B. We have low-rank updates via AB, where the rank is denoted as r, which is a hyperparameter.

Table 2
Overview of the accuracy in detecting if a text is written by an human in task 4 on PAN 2024 (Voight-Kampff Generative AI Authorship Verification). We report ROC-AUC, Brier, C@1, F_1, $F_{0.5u}$ and their mean.

Approach	ROC-AUC	Brier	C@1	F_1	$F_{0.5u}$	Mean
ferocious-coot	0.937	0.91	0.906	0.894	0.894	0.908
marinated-pantone	0.967	0.955	0.965	0.939	0.943	0.954
Baseline Binoculars	0.972	0.957	0.966	0.964	0.965	0.965
Baseline Fast-DetectGPT (Mistral)	0.876	0.8	0.886	0.883	0.883	0.866
Baseline PPMd	0.795	0.798	0.754	0.753	0.749	0.77
Baseline Unmasking	0.697	0.774	0.691	0.658	0.666	0.697
Baseline Fast-DetectGPT	0.668	0.776	0.695	0.69	0.691	0.704
95-th quantile	0.994	0.987	0.989	0.989	0.989	0.990
75-th quantile	0.969	0.925	0.950	0.933	0.939	0.941
Median	0.909	0.890	0.887	0.871	0.867	0.889
25-th quantile	0.701	0.768	0.683	0.657	0.670	0.689
Min	0.131	0.265	0.005	0.006	0.007	0.224

The Table 3 shows that the proposed LAVA ("mariented-pantone") approach wins almost all baselines in terms of quality, except Binoculars baseline. It is also worth noting that our method is not limited by resources, since we can train any number of adapters for a different family of data generated by models. In addition, since we aggregate the outputs, our approach has more confidence in predicting the class, depending on the results of the adapters.

As for approach "ferocious-coot", behind it there is not aggregation of adapters, but the use of only one of all the adapters trained in the LAVA method. Its quality turned out to be lower, so aggregation was used as the final choice. Which in turn gives the insight that using different adapters on different family of data makes sense for quality gains.

5. Conclusion

In this paper, a study was conducted on the classification of machine-generated texts according to the PAN competition. The analysis of the area and the research were carried out in existing methods for searching machine-generated excerpts. A new approach is proposed based on training a group of

Table 3
Overview of the mean accuracy over 9 variants of the test set. We report the minumum, median, the maximum, the 25-th, and the 75-th quantile, of the mean per the 9 datasets.

Approach	Minimum	25-th Quantile	Median	75-th Quantile	Max
ferocious-coot	0.581	0.745	0.881	0.908	0.987
marinated-pantone	0.674	0.820	0.942	0.954	0.996
Baseline Binoculars	0.342	0.818	0.844	0.965	0.996
Baseline Fast-DetectGPT (Mistral)	0.095	0.793	0.842	0.931	0.958
Baseline PPMd	0.270	0.546	0.750	0.770	0.863
Baseline Unmasking	0.250	0.662	0.696	0.697	0.762
Baseline Fast-DetectGPT	0.159	0.579	0.704	0.719	0.982
95-th quantile	0.863	0.971	0.978	0.990	1.000
75-th quantile	0.758	0.865	0.933	0.959	0.991
Median	0.605	0.645	0.875	0.889	0.936
25-th quantile	0.353	0.496	0.658	0.675	0.711
Min	0.015	0.038	0.231	0.244	0.252

QLoRA adapters to identify patterns in the distribution of data, which helps to make a more accurate classification of texts. Adapters make it possible to easily obtain a high-quality model in different narrow domains without having to retrain the entire large language model. We have tried different approaches to aggregating predictions, but we used the OR operation. Our model outputs class 0 only if all trained adapters output such a class. From comparing the approaches, it can be seen that our model works above almost all the presented baselines, with mean value across competition's metrics equal to 0.954. This method is efficient and flexible in terms of running time and resource intensity.

In future work, we would like to investigate more the use of multiple adapters for the artificial text detection task. We aim to compare other aggregation methods, as well as to assign weights to each adapter depending on the importance and popularity of the family for the given task. In addition, it is necessary to try this approach on other language models, because this method is not tied to only one model.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[2] ChatGPT by OpenAI, https://chat.openai.com, 2022.

[3] Google Bard, https://google-bard-ai.com/try-bard/, 2019.

[4] Jasper, https://www.jasper.ai/, 2023.

[5] YaGPT by Yandex, https://yandex.ru/project/alice/yagpt, 2023.

[6] GigaChat by SberDevices, https://developers.sber.ru/portal/products/gigachat, 2023.

[7] Y. Liu, Z. Zhang, W. Zhang, S. Yue, X. Zhao, X. Cheng, Y. Zhang, H. Hu, Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models, arXiv preprint arXiv:2304.07666 (2023).

[8] Y. Ma, J. Liu, F. Yi, Q. Cheng, Y. Huang, W. Lu, X. Liu, Ai vs. human–differentiation analysis of scientific content generation, arXiv preprint arXiv:2301.10416 (2023).

[9] G. Gritsay, Y. V. Chekhovich, Artificially generated text fragments search in academic documents, in: Doklady Mathematics, volume 108, Springer, 2023, pp. S434–S442.

[10] R. He, L. Liu, H. Ye, Q. Tan, B. Ding, L. Cheng, J.-W. Low, L. Bing, L. Si, On the effectiveness of adapter-based tuning for pretrained language model adaptation, arXiv preprint arXiv:2106.03164 (2021).

[11] R. Wang, D. Tang, N. Duan, Z. Wei, X. Huang, G. Cao, D. Jiang, M. Zhou, et al., K-adapter: Infusing knowledge into pre-trained models with adapters, arXiv preprint arXiv:2002.01808 (2020).
[12] A. A. Ayele, N. Babakov, J. Bevendorff, X. B. Casals, B. Chulvi, D. Dementieva, A. Elnagar, D. Freitag, M. Fröbe, D. Korenčić, M. Mayerl, D. Moskovskiy, A. Mukherjee, A. Panchenko, M. Potthast, F. Rangel, N. Rizwan, P. Rosso, F. Schneider, A. Smirnova, E. Stamatatos, E. Stakovskii, B. Stein, M. Taulé, D. Ustalov, X. Wang, M. Wiegmann, S. M. Yimam, E. Zangerle, Overview of PAN 2024: Multi-Author Writing Style Analysis, Multilingual Text Detoxification, Oppositional Thinking Analysis, and Generative AI Authorship Verification, in: L. Goeuriot, P. Mulhem, G. Quénot, D. Schwab, L. Soulier, G. M. D. Nunzio, P. Galuščáková, A. G. S. de Herrera, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2024.
[13] J. Bevendorff, M. Wiegmann, J. Karlgren, L. Dürlich, E. Gogoulou, A. Talman, E. Stamatatos, M. Potthast, B. Stein, Overview of the "Voight-Kampff" Generative AI Authorship Verification Task at PAN and ELOQUENT 2024, in: G. Faggioli, N. Ferro, P. Galuščáková, A. G. S. de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2024.
[14] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241. doi:10.1007/978-3-031-28241-6_20.
[15] S. Badaskar, S. Agarwal, S. Arora, Identifying real or fake articles: Towards better language modeling, in: Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II, 2008.
[16] Y. Freund, R. Schapire, N. Abe, A short introduction to boosting, Journal-Japanese Society For Artificial Intelligence 14 (1999) 1612.
[17] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, R. Mihalcea, Automatic detection of fake news, arXiv preprint arXiv:1708.07104 (2017).
[18] A. Roy, K. Basak, A. Ekbal, P. Bhattacharyya, A deep ensemble framework for fake news detection and classification, 2018. arXiv:1811.04670.
[19] L. O. Chua, CNN: A paradigm for complexity, volume 31, World Scientific, 1998.
[20] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, arXiv preprint arXiv:1508.01991 (2015).
[21] LIAR, https://datasets.activeloop.ai/docs/ml/datasets/liar-dataset/, 2017.
[22] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
[23] Z. Lu, P. Du, J.-Y. Nie, Vgcn-bert: augmenting bert with graph embedding for text classification, in: Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I 42, Springer, 2020, pp. 369–382.
[24] D. Ippolito, D. Duckworth, C. Callison-Burch, D. Eck, Automatic detection of generated text is easiest when humans are fooled, arXiv preprint arXiv:1911.00650 (2019).
[25] G. Gritsay, A. Grabovoy, Y. Chekhovich, Automatic detection of machine generated texts: Need more tokens, in: 2022 Ivannikov Memorial Workshop (IVMEM), IEEE, 2022, pp. 20–26.
[26] M. Behjati, S.-M. Moosavi-Dezfooli, M. S. Baghshah, P. Frossard, Universal adversarial attacks on text classifiers, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 7345–7349.
[27] Microsoft LLM, https://huggingface.co/microsoft/phi-2, 2023.
[28] OneClassSVM, https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html, 2001.
[29] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank

adaptation of large language models, 2021. arXiv:2106.09685.
[30] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, Qlora: Efficient finetuning of quantized llms, 2023. arXiv:2305.14314.
[31] Mistral AI, https://huggingface.co/docs/transformers/main/model_doc/mistral, 2023.
[32] LMSYS Chatbot Arena Leaderboard, https://chat.lmsys.org/?leaderboard, 2024.
[33] Aaditya Bhat, Gpt-wiki-intro (revision 0e458f5), 2023. URL: https://huggingface.co/datasets/aadityaubhat/GPT-wiki-intro. doi:10.57967/hf/0326.
[34] R. Eldan, Y. Li, Tinystories: How small can language models be and still speak coherent english?, 2023. URL: https://arxiv.org/abs/2305.07759. arXiv:2305.07759.
[35] V. Verma, E. Fleisig, N. Tomlin, D. Klein, Ghostbuster: Detecting text ghostwritten by large language models, 2024. URL: https://arxiv.org/abs/2305.15047. arXiv:2305.15047.
[36] H. Xu, J. Ren, P. He, S. Zeng, Y. Cui, A. Liu, H. Liu, J. Tang, On the generalization of training-based chatgpt detection methods, 2023. arXiv:2310.01307.
[37] L. Yang, F. Jiang, H. Li, Is chatgpt involved in texts? measure the polish ratio to detect chatgpt-generated text, APSIPA Transactions on Signal and Information Processing 13 (????).
[38] D. Macko, R. Moro, A. Uchendu, J. Lucas, M. Yamashita, M. Pikuliak, I. Srba, T. Le, D. Lee, J. Simko, M. Bielikova, Multitude: Large-scale multilingual machine-generated text detection benchmark, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2023. URL: http://dx.doi.org/10.18653/v1/2023.emnlp-main.616. doi:10.18653/v1/2023.emnlp-main.616.
[39] L. Ben Allal, A. Lozhkov, G. Penedo, T. Wolf, L. von Werra, Cosmopedia, 2024. URL: https://huggingface.co/datasets/HuggingFaceTB/cosmopedia.
[40] Y. Li, Q. Li, L. Cui, W. Bi, Z. Wang, L. Wang, L. Yang, S. Shi, Y. Zhang, Mage: Machine-generated text detection in the wild, 2024. arXiv:2305.13242.
[41] Y. Wang, J. Mansurov, P. Ivanov, J. Su, A. Shelmanov, A. Tsvigun, C. Whitehouse, O. Mohammed Afzal, T. Mahmoud, T. Sasaki, T. Arnold, A. Aji, N. Habash, I. Gurevych, P. Nakov, M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection, in: Y. Graham, M. Purver (Eds.), Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, St. Julian's, Malta, 2024, pp. 1369–1407. URL: https://aclanthology.org/2024.eacl-long.83.
[42] N. I. Tripto, A. Uchendu, T. Le, M. Setzu, F. Giannotti, D. Lee, Hansen: Human and ai spoken text benchmark for authorship analysis, arXiv preprint arXiv:2310.16746 (2023).

A. Obtained collection sources

Table 4 depicts the datasets that were used to form the merged family-based collection. The distinctive feature of this collection is its richness in a large number of topics ranging from news to transcripts from podcasts.

Table 4
Overview of part of the datasets that are contained in the large collection we have gathered on model families. This includes text from the GPT family, the Llama and Mistral families, and other models.

Dataset	Source
GPT-wiki-intro-extension [33]	GPT-3
TinyStories v.1 [34]	GPT-3.5
ChatGPT-Research-Abstracts	GPT-3.5
Ghostbuster [35]	GPT-3.5
HC-Var [36]	GPT-3.5
TinyStories v.2	GPT-4
ShareGPT	GPT-4
ChatGPT-Detection-PR-HPPT [37]	Llama
SnifferBench	Llama, Alpaca
MULTITuDE [38]	Llama, Alpaca, OPT, Vicuna
Cosmopedia v0.1 [39]	Mistral
MAGE [40]	BLOOM
M4 [41]	BLOOM, Cohere
HANSEN [42]	Vicuna, GPT-3.5