

Advacheck at GenAI Detection Task 1: AI Detection Powered by Domain-Aware Multi-Tasking

German Gritsai^{1,2}, Anastasia Voznyuk¹, Ildar Khabutdinov^{1,2}, Andrey Grabovoy¹

¹Advacheck OÜ, Estonia

²Université Grenoble Alpes, France

{gritsai, voznyuk, khabutdinov, grabovoy}@advacheck.com

Abstract

The paper describes a system designed by Advacheck team to recognise machine-generated and human-written texts in the monolingual subtask of GenAI Detection Task 1 competition. Our developed system is a multi-task architecture with shared Transformer Encoder between several classification heads. One head is responsible for binary classification between human-written and machine-generated texts, while the other heads are auxiliary multiclass classifiers for texts of different domains from particular datasets. As multiclass heads were trained to distinguish the domains presented in the data, they provide a better understanding of the samples. This approach led us to achieve the first place in the official ranking with 83.07% macro F_1 -score on the test set and bypass the baseline by 10%. We further study obtained system through ablation, error and representation analyses, finding that multi-task learning outperforms single-task mode and simultaneous tasks form a cluster structure in embeddings space. We release our code and model¹.

1 Introduction

With the continuous improvement of Large Language Models (LLMs), the task of detection machine-generated texts demands more and more attention from the community. The potential cases of misuse include malicious usage by students (Zeng et al., 2023; Koike et al., 2023) and scientists (Ma et al., 2023; Gritsai et al., 2023a). Furthermore, this is often the cause of plagiarism (Avetisyan et al., 2023) and spam (Labonne and Moran, 2023). The mentioned things are encouraging researchers to improve methods for detecting artificial text simultaneously with enhancing generation methods.

¹<https://github.com/Advacheck-OU/ai-detector-coling2025>

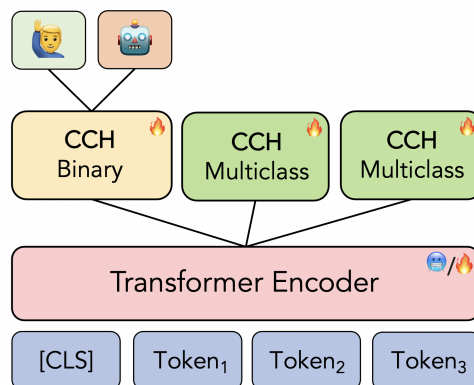





Figure 1: Overview of the proposed multi-task architecture. Modules marked only with  are trainable at all stages. The weights of the Transformer Encoder are frozen  at the first stage of training and trainable  at the second one. The Custom Classification Head (CCH) described in Appendix A is used for predictions.

The task of detecting machine-generated texts is usually formulated as a binary text classification task (Jawahar et al., 2020). The most common solutions are to fine-tune the Transformer-based model (Gritsai et al., 2023b) or to use zero-shot approaches with intrinsic statistics of the text (Mitchell et al., 2023; Hans et al., 2024). While these methods perform well on in-domain tasks (Uchendu et al., 2021), they are not robust to change of the domain, generator model, or language of the texts (Wang et al., 2024a; Tulchinskii et al., 2023; Kuznetsov et al., 2024). Meanwhile, for the detection of AI-content in the wild such a change is, on the contrary, a more realistic setup (Dugan et al., 2024). Beyond this, the data presented for the detection task may be of poor quality, which also complicates the challenge of the detection task (Gritsai et al., 2024). Therefore, the goal is to obtain a model that is robust to the presence of data of poor quality and with a lot of noise, and, in addition, to make this model capable to adapt to new domains.

The high coherence and quality of writing achieved by modern LLMs makes it difficult to find a specific sample-counted qualitative feature by which it would be possible to create a hyper-plane in the space of texts and separate generated from human-generated ones. One of the possible refinements of the representations from the single-task learning architecture is the multi-task learning (MTL) (Crawshaw, 2020; Gritsay et al., 2022). It was also noted that systems with MTL architecture had achieved high results in the previous competitions (Guo et al., 2024), therefore we decided to utilize this approach in our work. In this paper we discuss our solution as the Advcheck team at GenAI Content Detection Task 1: English and Multilingual Machine-generated Text Detection: AI vs. Human (Wang et al., 2025). Our method shows that with additional internal data analysis and embedding alignment using MTL, it is still possible to achieve high performance in detecting fragments in cross-domain and cross-generator setups on texts from the advanced LLMs. As we forced model to focus on various domains, it allowed us to form a cluster domain-wise structure for the text representations in the vector space. In our research, we show that (1) multi-task learning outperforms the default single-task, (2) cluster structure is formed at the shared encoder (3) compare different configurations of the system and (4) analyse the errors of the system.

2 Task Definition

The monolingual subtask of *Task 1: Binary Multilingual Machine-Generated Text Detection* focuses on identifying whether the English text was entirely authored by a human or generated by a language model. The competition is the continuation and improvement of the *SemEval Shared Task 8 (subtask A)* (Wang et al., 2024b) and combines refreshed training and testing samples from different domains and novel LLMs. The statistics of the dataset are summarised in Appendix C. The official evaluation metric for the monolingual subtask is Macro F_1 -score and the additional metric is Micro F_1 -score.

3 System Overview

Why multi-task learning? In the current task formulation we have more than 600k texts in the training set with dozens of domains and generation models. Such an amount is very noisy for a binary

classification task, because it can be challenging for a model to distinguish between relevant and irrelevant features. Multi-task learning may help the model focus on those features that actually matter as other tasks will update representation of samples with inner information. Our aim is to obtain fine-grained representations of the data that ideally ignores data-dependent noise and generalises well. Since different tasks involve distinct noises, a model trained on multiple tasks simultaneously is able to learn a more general representation. Furthermore, it reduces the risk of overfitting.

Model. We propose an MTL architecture with hard parameter sharing (HPS), it is depicted in Figure 1. In HPS, a common Transformer-based encoder is used for multiple tasks. After several variations of set of parallel heads, we focused on three custom classification heads (CCH) for simultaneous training:

- Binary CCH head for solving the initial monolingual subtask [2 classes]
- Multiclass CCH to define a sub-source within the HC3 (Su et al., 2024) source [5 classes]
- Multiclass CCH for sub-source detection within the M4GT (Wang et al., 2024a) source [6 classes]

The model was trained in two phases: fine-tuning chosen classifiers with frozen 🧊 shared encoder weights and fine-tuning the complete model with all weights unfrozen 🔥. These learning stages help to shift the distribution of the encoder weights in the right direction and avoid overfitting (Xie et al., 2021). At the inference stage, only binary CCH predictions used for final classification.

4 Experiments

We focused on the monolingual subtask, carrying out comparisons among models and ablations of the best system. For these we employed the original training and development splits provided by the organizers. Our objective here was to reveal the quality improvement in multi-task training compared to single-task training.

4.1 Model Comparison

Baselines. As models for comparison we chose Logistic Regression classifier (Chao-Ying Joanne Peng and Ingersoll, 2002) with TF-IDF

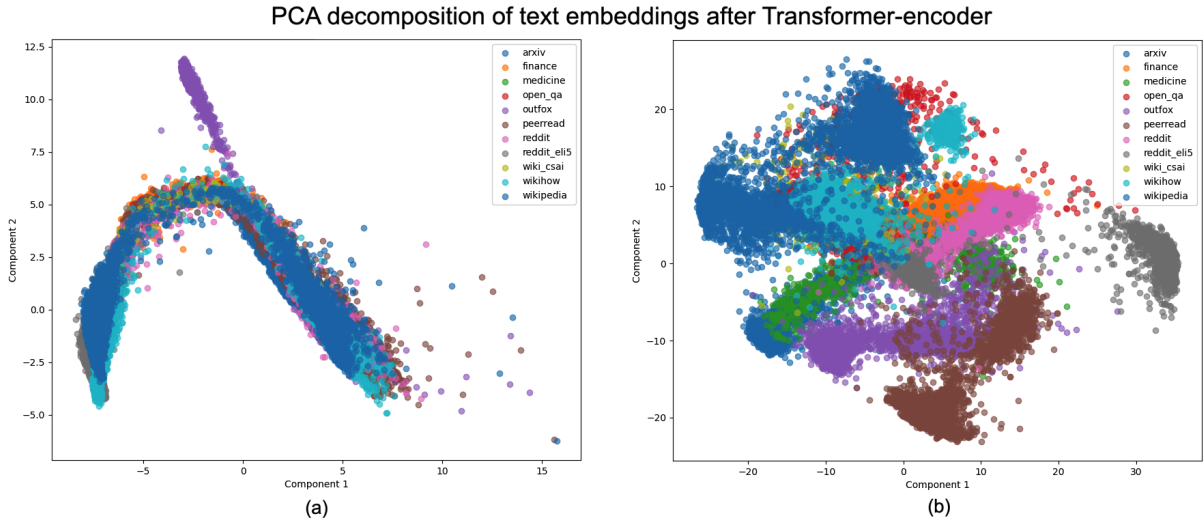


Figure 2: Two principal component decomposition of PCA for texts from the development subsample. In (a) the vector space structure for the `deberta-v3-base` fine-tuned in single-task mode is shown, while figure (b) shows the same model but fine-tuned in MTL mode with two additional custom classification heads.

Model	Development	Test	Rank	System	F_1 -score (%)
TF-IDF with LogReg	63.53	60.93	1	Advacheck (germgr)	83.07
DeBERTaV3 base	82.56	78.52	2	tmatchitan	83.01
MTL: 1 stage	80.51	78.67	3	karla	82.80
MTL: 2 stage	87.33	81.55	15	<i>baseline</i>	73.42
MTL: 2 stage + threshold	87.96	83.07	36	nitstejasrikar	44.89

Table 1: Results of model comparison on the test and development set. The highlighted metric is macro F_1 -score (%).

Table 2: Final results on the official ranking. Bold denotes our team’s placement.

features on word n-grams, and DeBERTa-v3 fine-tuned in two-stage mode described earlier, but in single-task setting. In the MTL approach, we compared checkpoints from different stages, and also explored the effect of adding thresholds on the output of the final classifier. We chose DeBERTa-v3 base for the baseline and the backbone in our system, as it is currently state-of-the-art model for supervised fine-tuning for binary classification (Macko et al., 2023).

The results are presented in Table 1. It can be seen that there is a weak correlation between the gap within the predictions on the dev and test subsamples. For example, the presence of a threshold after the final layer affected the dev result only slightly, but at the same time allowed us to achieve a winning result on the test set. The hyperparameters of the final model are given in Appendix B.

5 Results

Table 2 reports the leaderboard results on the test set, where our system, Advacheck, achieves a macro F_1 -score of 83.07%, outperforming approaches of the other participants and ranking first. Our solution surpassed the claimed baseline by 10%.

6 Analysis

6.1 Embeddings after MTL

We made a comparison of text embeddings after fine-tuning stages. Samples from development part of data were forwarded to the Transformer-encoder and [CLS] vectors were extracted as outputs. We visualised these vectors using PCA in Figure 2. We observe that the alignment of the representations, which was mentioned above, introduces a cluster domain-wise structure. Although the clusters are not perfectly separable, a meaningful difference between the standard BERT-like model and MTL

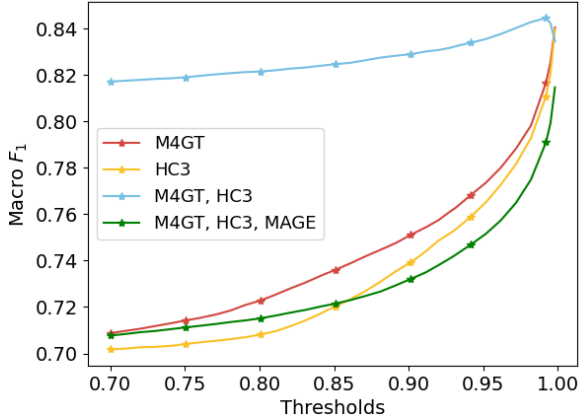


Figure 3: Macro F_1 -score on the test set of different configuration of the systems depending on the threshold.

Task Head	Development	Test
HC3	92.27	82.70
M4GT	91.70	81.07
MTL (HC3 + M4GT)	87.96	83.07
HC3 + M4GT + MAGE	91.43	79.23

Table 3: Comparison of different configurations of heads and tasks trained simultaneously in MTL architecture. The highlighted metric is macro F_1 -score (%).

fine-tuning pipelines can be seen. Additional decompositions are presented in Appendix E.

6.2 Ablations

To provide further analysis on the multi-task setup we experimented with configurations of our systems, changing the number of multiclass CCH. The system on the leaderboard has 2 Multiclass CCH, and we ran ablation experiments with 1 CCH and 3 CCH. The results are in the Table 3. Setups with 1 and 3 CCH showed better performance on the development set, but marginally dropped in performance on the test set compared to setup with 2 CCH. What is also interesting is that the results obtained on HC3-trained CCH are similar to the results obtained in M4GT-trained CCH, although M4GT has 10 times more training data than HC3. Additionally, we experimented with the threshold values on all our configurations. The figures are shown in the Figure 3 and reaffirm the choice of the final system and threshold for it.

6.3 Error Analysis

Answers on different datasets. The two datasets with the highest percentage of incorrect predictions are Mixset (Zhang et al., 2024) and CUDRT (Tao

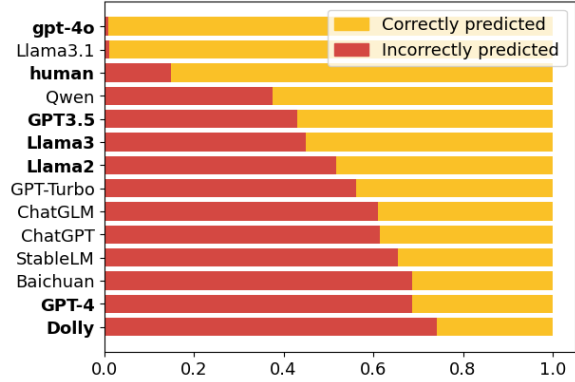


Figure 4: Proportion of predictions for different generators from test set. Labels in **bold** are generators texts from which are present in the train set.

et al., 2024), while the texts in the other datasets are detected with very high precision. We attribute this to the additional manipulations with these texts, such as rewriting, “humanizing” and other editing, done by the authors of the initial datasets. Other texts in the test set and in train set, on the other hand, are the raw output of generation models, without post-processing them, or otherwise not stated. See more details in Appendix D. In future iterations, it may be worth trying to address these challenges with the post-processing addition for some samples in the training data.

Answers on different generators. The proportions of correct and incorrect predictions on test set with respect to different generators are shown in Figure 4. The majority of texts in the test set are either human-written or generated with gpt-4o and our system predicted labels for them very accurately, therefore compensating the poorer performance on other generators. Still, our system is not yet robust enough to the change of generators, such as texts from some of the unseen detectors, and we will focus future research on addressing generalisation gaps and refining the approach for unseen generators.

7 Conclusion

In this paper we described the system by the Advacheck team in the monolingual subtask at GenAI Detection Task 1 competition. We proposed solution with multi-task learning architecture that consists of shared Transformer Encoder and composition of one binary and two multiclass Custom Classification Heads. Our system obtained the best results in the official ranking bypassing the baseline by 10%. Adding tasks for training in parallel

reveal the formation of a cluster structure in the space of embeddings, helping to achieve high results despite the presence of a large amount of noisy data. Also, it has been demonstrated that training a similar model but in single-task mode loses to the proposed approach, and configurations with one or three multiclass heads also perform worse than our final system.

References

- K. Avetisyan, G. Gritsay, and A. Grabovoy. 2023. [Cross-lingual plagiarism detection: Two are better than one](#). *Program. Comput. Softw.*, 49(4):346–354.
- Kuk Lida Lee Chao-Ying Joanne Peng and Gary M. Ingersoll. 2002. [An introduction to logistic regression analysis and reporting](#). *The Journal of Educational Research*, 96(1):3–14.
- Michael Crawshaw. 2020. [Multi-task learning with deep neural networks: A survey](#). *Preprint*, arXiv:2009.09796.
- Liam Dugan, Alyssa Hwang, Filip Trhlik, Josh Magnus Ludan, Andrew Zhu, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. [Raid: A shared benchmark for robust evaluation of machine-generated text detectors](#). *Preprint*, arXiv:2405.07940.
- German Gritsai, Anastasia Voznyuk, Andrey Grabovoy, and Yury Chekhovich. 2024. [Are ai detectors good enough? a survey on quality of datasets with machine-generated texts](#). *Preprint*, arXiv:2410.14677.
- German Gritsai, Andrey Grabovoy, and Yury Chekhovich. 2022. [Automatic detection of machine generated texts: Need more tokens](#). In *2022 Ivan-nikov Memorial Workshop (IVMEM)*, pages 20–26.
- German Gritsai, Andrey Grabovoy, Aleksandr Kildyakov, and Yury Chekhovich. 2023a. [Artificially generated text fragments search in academic documents](#). In *Doklady Mathematics*, volume 108, pages S434–S442. Springer.
- German Gritsai, Andrey Grabovoy, Aleksandr Kildyakov, and Yury Chekhovich. 2023b. [Automated text identification: Multilingual transformer-based models approach](#). In *IberLEF@ SEPLN*.
- Zikang Guo, Kaijie Jiao, Xingyu Yao, Yuning Wan, Haoran Li, Benfeng Xu, Licheng Zhang, Quan Wang, Yongdong Zhang, and Zhendong Mao. 2024. [USTC-BUPT at SemEval-2024 task 8: Enhancing machine-generated text detection via domain adversarial neural networks and LLM embeddings](#). In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1511–1522, Mexico City, Mexico. Association for Computational Linguistics.
- Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. [Spotting llms with binoculars: Zero-shot detection of machine-generated text](#). *Preprint*, arXiv:2401.12070.
- Dan Hendrycks and Kevin Gimpel. 2023. [Gaussian error linear units \(gelus\)](#). *Preprint*, arXiv:1606.08415.
- Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. 2020. [Automatic detection of machine generated text: A critical survey](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2296–2309, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2023. [Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples](#). In *AAAI Conference on Artificial Intelligence*.
- Kristian Kuznetsov, Eduard Tulchinskii, Laida Kushnareva, German Magai, Serguei Barannikov, Sergey Nikolenko, and Irina Piontkovskaya. 2024. [Robust ai-generated text detection by restricted embeddings](#). *Preprint*, arXiv:2410.08113.
- Maxime Labonne and Sean Moran. 2023. [Spam-t5: Benchmarking large language models for few-shot email spam detection](#). *Preprint*, arXiv:2304.01238.
- Yongqiang Ma, Jiawei Liu, Fan Yi, Qikai Cheng, Yong Huang, Wei Lu, and Xiaozhong Liu. 2023. [Ai vs. human – differentiation analysis of scientific content generation](#). *Preprint*, arXiv:2301.10416.
- Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. [Multitude: Large-scale multilingual machine-generated text detection benchmark](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, page 9960–9987. Association for Computational Linguistics.
- Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. [Detectgpt: Zero-shot machine-generated text detection using probability curvature](#). *Preprint*, arXiv:2301.11305.
- Zhenpeng Su, Xing Wu, Wei Zhou, Guangyuan Ma, and Songlin Hu. 2024. [Hc3 plus: A semantic-invariant human chatgpt comparison corpus](#). *Preprint*, arXiv:2309.02731.
- Zhen Tao, Yanfang Chen, Dinghao Xi, Zhiyu Li, and Wei Xu. 2024. [Cudrt: Benchmarking the detection models of human vs. large language models generated texts](#). *Preprint*, arXiv:2406.09056.

Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Sergey Nikolenko, Evgeny Burnaev, Serguei Barannikov, and Irina Piontkovskaya. 2023. Intrinsic dimension estimation for robust detection of ai-generated texts. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA. Curran Associates Inc.

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, Punta Cana, Dominican Republic. 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, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024a. **M4GT-bench: Evaluation benchmark for black-box machine-generated text detection**. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3964–3992, Bangkok, Thailand. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation, SemEval 2024*, Mexico, Mexico.

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 1st Workshop on GenAI Content Detection (GenAIDetect)*, Abu Dhabi, UAE. International Conference on Computational Linguistics.

Sang Michael Xie, Tengyu Ma, and Percy Liang. 2021. **Composed fine-tuning: Freezing pre-trained denoising autoencoders for improved generalization**. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11424–11435. PMLR.

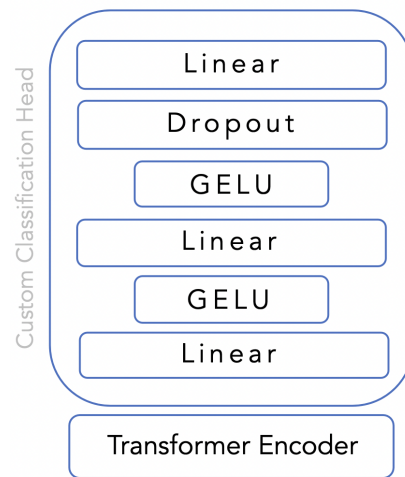


Figure 5: The architecture of the custom classification head used in the described approach. Value of dropout is equal to 0.5.

Zijie Zeng, Lele Sha, Yuheng Li, Kaixun Yang, Dragan Gašević, and Guanliang Chen. 2023. **Towards automatic boundary detection for human-ai collaborative hybrid essay in education**. *Preprint*, arXiv:2307.12267.

Qihui Zhang, Chujie Gao, Dongping Chen, Yue Huang, Yixin Huang, Zhenyang Sun, Shilin Zhang, Weiye Li, Zhengyan Fu, Yao Wan, and Lichao Sun. 2024. **Llm-as-a-coauthor: Can mixed human-written and machine-generated text be detected?** *Preprint*, arXiv:2401.05952.

A Custom Classification Head

In our approach, we replaced the default one-layer linear classifier with a more extended version by adding multiple layers, the final structure of Custom Classification Head (CCH) is shown in Figure 5. We chose GELU (Hendrycks and Gimpel, 2023) as the activation feature and added dropout. In earlier experiments, when compared with the base head, this adaptation gives a higher quality therefore we used it in all subsequent experiments.

B Final System Hyperparameters

Our final system is MTL architecture with shared deberta-v3-base encoder and 3 CCH. The training of main system was conducted on NVIDIA GeForce RTX 3090 and the training of other configurations on NVIDIA A100. See hyperparameters in Table 4.

C Provided Data

The organizers of the competition provided data for the train and development stages of the evolving

Hyperparameters	1st stage	2nd stage
Epochs	1	1
Learning rate (LR)	3e-4	3e-6
Warmup steps	50	75
Weight decay	0.01	0.01
Batch size	32	16
Classifier threshold	-	0.92

Table 4: Hyperparameters for fine-tuning MTL architecture. We trained for 1 epoch in both stages with possibility of early exit.

solutions. This is the continuation and improvement of the SemEval Shared Task 8 samples. New domains and generation models were added to the data; details of train and dev sets are shown in Table 7. In addition, a separate development dataset was available on the CodaBench² platform where the competition was held; its statistics are shown in Table 5.

Source	Development Set	
	Human	Machine
RAID	13371	0
LLM-DetectAIve	0	19186
Total		32557

Table 5: Statistics on development data from CodaBench platform for monolingual subtask of the GenAI Detection Task 1.

Source	Test Set	
	Human	Machine
CUDRT	12287	10691
IELTS Duck	9747	12418
PeerSum	5080	6995
LLM-DetectAIve	1635	900
Mixset	0	1086
NLPeer	5326	5376
Total	34075	37466

Table 6: Statistics on test data for monolingual subtask of the GenAI Detection Task 1.

D Percentage of Failures

See Figure 6 for detailed proportions of incorrectly predicted texts from CUDRT and Mixset.

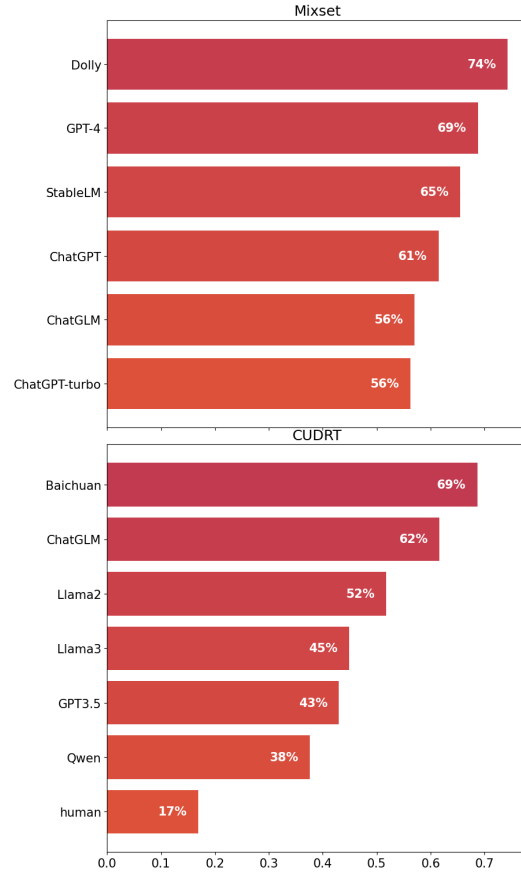


Figure 6: The percentage of falsely labelled texts from two subdatasets in test set.

E Decomposition Study

In addition to the PCA decomposition of text embeddings after passes on our system, we mapped the logit decomposition of two multiclass heads – CCH on HC3 and CCH on M4GT. We sampled the texts from the dev set, passed them through the encoder and the corresponding classifiers, and then decomposed the logits. From Figure 7, we can observe that the data after the classifiers passes remain in the expected cluster structure intended by shared encoder. On the dataset HC3 this can be seen more clearly. With these plots we can also understand how well the multiclass classification heads were trained directly, as they were not used for the inference.

²<https://www.codabench.org/competitions/3734/>

Source	Sub-sources	Training Set		Dev Set	
		Human	Machine	Human	Machine
HC3	Finance, Medicine, OpenQA, Reddit_ELI5, Wiki_CSAI	39140	17671	16501	7917
M4GT	Arxiv, Outfox, PeerRead, Reddit, WikiHow, Wikipedia	86682	180381	36420	74167
MAGE	CMV, CNN, DialogSum, ELI5, HellaSwag, IMDB, PubMed, Roct, SciGen, SQUAD, TLDR, WP, XSum, Yelp	103100	183793	45407	81462
Total		228922	381845	98328	163430

Table 7: Statistics on training and development data from monolingual subtask of the GenAI Detection Task 1.

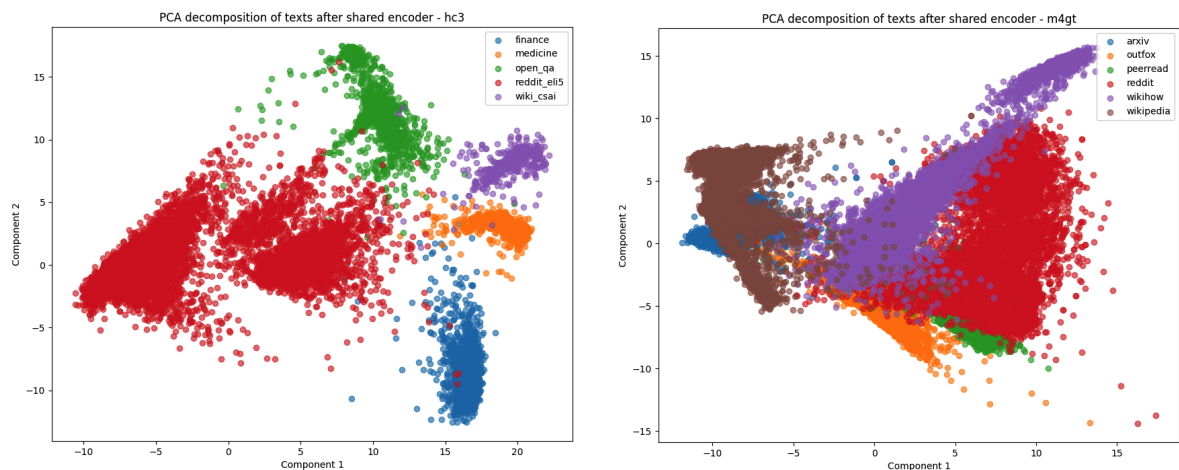


Figure 7: Two principal component decomposition of PCA for texts from development sub-sample. We decomposed here by PCA the logits of the texts after the corresponding multiclass classifiers, namely HC3 and M4GT.