



Papilusion at DAGPap24: Paper or Illusion? Detecting AI-generated Scientific Papers

Nikita Andreev[‡], Alexander Shirnin[‡]

Vladislav Mikhailov^{*}, Ekaterina Artemova[‡]

[‡]CAIT and Applied AI Institute, [‡]HSE University

^{*}University of Oslo, [‡]Toloka AI

Correspondence: ashirnin@hse.ru

Abstract

This paper presents Papilusion, an AI-generated scientific text detector developed within the DAGPap24 shared task on detecting automatically generated scientific papers. We propose an ensemble-based approach and conduct ablation studies to analyze the effect of the detector configurations on the performance. Papilusion is ranked 6th on the leaderboard, and we improve our performance after the competition ended, achieving 99.46 (+9.63) of the F1-score on the official test set.

1 Introduction

A series of the DAGPap shared tasks addresses mitigation of the risks associated with misusing language technologies in the scientific domain, such as fabricating research publications (Else and Van Noorden, 2021; Van Noorden, 2021) and affecting the peer review ecosystem at the leading AI conferences (Liang et al., 2024). DAGPap22 (Kashnitsky et al., 2022) focuses on a well-established task formulation in the artificial text detection field (Uchendu, 2023), which requires to determine if a given text is human-written or AI-generated. DAGPap24¹ offers a less explored task formulation, which is framed as a token-level classification problem. Developing robust and reliable AI-generated scientific text detectors promotes analyzing scientific text collections at scale and assisting non-expert and expert users in identifying the fabricated content (Cabanac et al., 2021; Else, 2023).

This paper proposes Papilusion², an AI-generated text detection system for tagging text segments, which are produced by a generative language model (LM) or modified through synonym replacement and summarization. The token-level AI-generated text detection accounts for a more

practical scenario, where a generative LM is used as a writing assistant. Papilusion is an ensemble of encoder-only sequence taggers that leverages the strengths of multiple independently finetuned LMs to enhance the accuracy of detecting AI-generated text. We conduct ablation studies to analyze the impact of various hyperparameter and LM configurations on the target performance.

Papilusion is ranked 6th among 30 participating teams at DAGPap24, with 89.83 of the F1-score on the leaderboard. After the competition ends, we improve our solution achieving 99.46 (+9.63) of the F1-score on the official evaluation set.

2 Background

The DAGPap24 dataset is a collection of human-machine mixed scientific texts, where the text segments can be (i) human-written, (ii) modified through an NLTK-based synonym replacement (Bird and Loper, 2004), (iii) produced by ChatGPT, and (iv) summarized. The organizers provide 5000, 5000, and 20000 training, development, and test examples, respectively. The average number of tokens³ is 5591.

Task Formulation The task is to assign one of the four corresponding labels to each token in a given text, as it was described above. The example for a text with a synonym replacement is shown below. Here, “1” stands for a synonym replacement, and “0” refers to the human-written segment:

- text: “this was continued until successful intravascular positioning. An observer, blinded to the lens allocation, <...>”
- tokens: [“this”, “was”, “continued”, “until”, “successful”, “intravascular”, “positioning.”,

¹github.com/ChamezopoulosSavvas/DAGPap24

²Papilusion is named after a pokémon *Papilusion*, which is the French name for *Butterfree*.

³The tokens are provided by the organizers and are obtained by splitting a text based on whitespace characters.

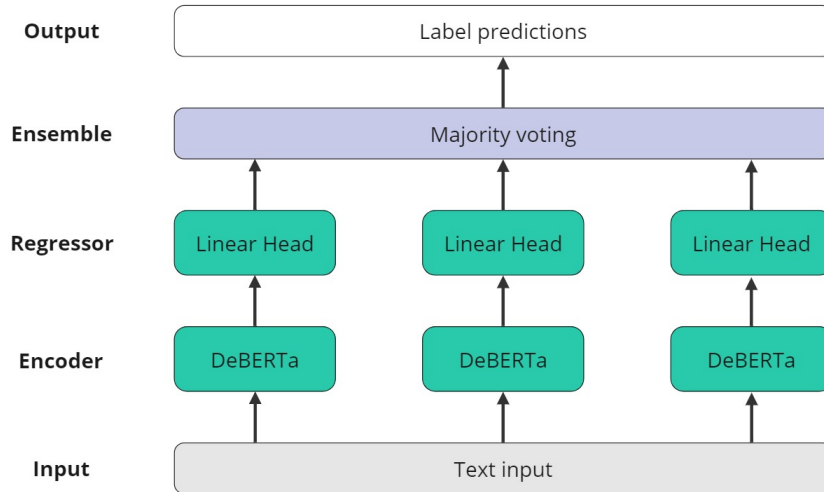


Figure 1: The PapiLusion pipeline involves fine-tuning three distinct encoder models, which are based on the same architecture but trained independently with different hyperparameters. These models use linear heads to predict labels that differentiate between human-written and machine-generated text. Finally, a majority vote is applied to aggregate the predicted labels.

“An”, “observer,” , “blinded”, “to”,
 “the”, “lens”, “allocation,” <...>]

- labels: [1, 1, 1, 1, 1, 1, 1, 0, 0,
 0, 0, 0, 0, 0 <...>]

Performance Metric The performance is evaluated using the macro-averaged F1-score.

3 PapiLusion

The PapiLusion pipeline involves several key steps (see Figure 1):

1. **Model Fine-Tuning:** We fine-tuned encoder models to predict the corresponding labels. Although based on the same architecture, the models were fine-tuned with different hyperparameters to create a more diverse ensemble. Each model was trained on a sequence labeling task, predicting labels for every token in the input sequence.
2. **Majority Voting:** After fine-tuning multiple models, we used a majority voting strategy to aggregate their predictions. This approach involved combining the predictions from each model and selecting for each token the label that received the most votes across the ensemble of models. It is a popular ensembling method, which helps to gain better performance by using several models (Wani et al., 2018; K. et al., 2020). For the ensemble we chose three best models, based on the performance metric on the development dataset.

By using encoder models fine-tuned for text classification and applying a majority voting mechanism, we aimed to improve the accuracy and robustness of our classification system.

4 Experiments

Overview We conducted a wide series of experiments with a family of DeBERTaV3 models (He et al., 2023). We investigate experiments with bottom layers freezing (Ingle et al., 2022) and the size of the input sequence as the most important hyperparameters for the efficient model selection. We used longer sequences to fit in more context for a model. Additionally, we believe that there are some dependencies between the size of the model and effectiveness of a layer freezing and using various input sequences lengths. In our experiments we conducted a detailed research and comparison of various setups.

Model	Params	Hidden size	Layers
Xsmall	22M	384	12
Small	44M	768	6
Base	86M	768	12
Large	304M	1024	24
DistillBERT	66M	768	6

Table 1: Model architectures comparison. The DistillBERT model was used as a baseline by the competition organizers.

Model size	Xsmall			Small		Base			Large			
	0	6	12	0	6	0	6	12	0	6	12	18
Frozen layers												
256 input	97.32	97.00	79.20	95.88	82.74	96.99	96.22	89.68	-	-	98.12	98.17
512 input	98.33	97.71	78.49	98.81	83.31	98.55	98.72	90.66	-	98.83	98.82	99.07
1024 input	86.38	85.82	55.65	88.13	63.14	88.12	80.97	71.24	96.29	-	98.84	99.21
2048 input	67.87	62.95	28.99	65.30	45.47	66.21	56.40	50.25	92.87	92.87	-	-

Table 2: Test results obtained after the end of competition. The symbol '-' is used to denote experiment results where with the given configuration it was not possible to achieve satisfactory or adequate outcomes.

Competition-included experiments During the competition we utilized competition organizers' baseline code and did several approaches of fine-tuning DeBERTa models. During the competition timeline we tried several variations of different hyperparameters such as input sequence length, model size, freezing several bottom layers of the model, learning rate scheduler. Our final submission brought us to 6th place with **89.83** F1 metric which was obtained with the ensemble of 3 DeBERTa-base models with different number of frozen layers and the default input sequence length (512 tokens).

Post-competition study After the end of competition we identified a tokenization issue in the code: the chunk function truncated sequence by the number of words, not tokens which resulted in information loss. We fixed our training scripts with proper tokenization in chunk function, along with some minor adjustments in the prediction function. These modifications significantly improved the models' performance, resulting in a score increase of +9.63, and enabled us to conduct a detailed study of various models and hyperparameter variations. The results are shown in the Table 2

We used the following hyperparameters for our experiments:

- Model size - {Xsmall, Small, Base, Large}
- Input sequence length - {256, 512, 1024, 2048}
- Frozen layers - {0, 6, 12, 18}

We fixed other hyperparameters (learning rate, scheduler, seed, validation dataset size) to perform fair comparison. In this case, we can see how changing previously mentioned hyperparameters affects the result. Additionally, we did a series of ensemble mixtures based on majority voting technique for our best models to improve our final score.

Hardware specification We run experiments on a single GPU TESLA V100 32GB. Model fine-tuning is conducted using the transformers library (Wolf et al., 2020). The fine-tuning for DeBERTa requires approximately 4-12 hours to complete depending on the size of the model and the length of the sequence. The inference on the official test set runs within 1 hour. For DeBERTa-large model we lower the batch size down to 4, for other model versions we set the batch size to 16.

5 Results

Competition phase During the competition we fine-tuned base DeBERTa models with different number of frozen bottom layers and then applied majority voting for 3 models. This brought us to the 6th place (see Table 3) and the best score among the teams below 90.0 F1 score.

Place	Test score
1st place	99.54
...	
4th place	99.24
5th place	95.54
6th place (Our system)	89.83
7th place	89.82
8th place	89.67
Baseline	86.36

Table 3: Competition results.

Post-competition phase Table 4 provides results of our best setups, ensembles based on those setups and comparison to the test scores on the leaderboard. More detailed results in various setups provided in the table 2, and our main outcomes are:

- 1024-token sequence inputs are typically better for DeBERTa-large model; however, we found that using a 512-token sequence results in comparable quality across almost all model types.

Setup	Model	Frozen layers	input length	Test score
a)	Xsmall	0	512	98.33
b)	Small	0	512	98.81
c)	Base	6	512	98.72
d)	Large	18	256	98.17
e)	Large	18	512	99.07
f)	Large	12	1024	98.84
g)	Large	18	1024	99.21
b) & c) & g)				99.28
d) & e) & g)				99.45
e) & f) & g)				99.46
1st place solution				99.54
2nd place solution				99.44
Baseline*	DistilBERT	0	512	93.64

Table 4: Comparison of best results across various experimental configurations, denoted by letters (a, b, c, etc.). Our ensemble achieves the 2nd place compared to competition scores. '&' sign stands for ensembling via majority voting. With '*' we mark improved baseline results obtained by ourselves after the competition.

- Freezing more of the bottom layers generally leads to better results for the DeBERTa-large model, while for other models it is better to not freeze the layers.
- Despite freezing all 12 encoder layers, we achieve a 90.66 F1 score with the Base DeBERTa. This result indicates that the task is relatively simple for these models.
- Ensembling the best single models leads to the score improvement.

The usage of DeBERTa models with corrected input sequence chunking allowed us to achieve significant improvements over the baseline. We got **98.33** F1 score with only Xsmall model version. It is **3 times smaller** than DistillBERT, that was used as the baseline model by competition organizers. Also we figured out that freezing of some layers could be helpful, especially for bigger models when model is already trained and have sufficient knowledge, thus we achieve our best single-model score of **99.21** with DeBERTa-large and 18/24 layers frozen. With the majority voting ensemble of our best Large models we achieve **99.46** F1 score which is compared to the second place based on the competition results.

As it was mentioned, the smallest DeBERTa model already gets F1 score above 98, while larger ones improve just moderately depending on their settings. This suggests that the problem we are addressing might be not particularly complex, reducing the need for large models. Upon a close examination of the dataset’s origin, we hypothesize that the construction process may have relied

heavily on basic generation rules and may not have included sufficient filtering of simple samples. This could have resulted in notable differences among various classes within the dataset. Additionally, we believe that synonym replacement is a simple and difficult-to-control process, likely causing a significant shift in the data distribution and making the task easier. Furthermore, the task of designing and creating text that combines inputs from both humans and LMs presents a separate significant and unique challenge.

6 Conclusion

In this paper, we proposed Papilusion, an AI-generated text detection system designed for the DAGPap24 shared task. Our system ranked 6th in the competition, achieving an F1 score of 89.83. Post-competition enhancements, including fixing tokenization errors and optimizing model parameters, increased F1 score to 99.46.

Through comprehensive ablation studies, we identified the most impactful hyperparameters and model configurations, leading to substantial performance improvements.

While our experiments demonstrate that larger models achieve the best results when resources are not limited, we also found that even the small models (DeBERTa-Xsmall) exhibit promising performance metrics with minimal computational resources.

Acknowledgements

AS’s work results from a research project implemented in the Basic Research Program at the Na-

tional Research University Higher School of Economics (HSE University). We acknowledge the computational resources of HSE University’s HPC facilities.

References

- Steven Bird and Edward Loper. 2004. [NLTK: The natural language toolkit](#). In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.
- Guillaume Cabanac, Cyril Labbé, and Alexander Magazhinov. 2021. Tortured phrases: A dubious writing style emerging in science. evidence of critical issues affecting established journals. *arXiv preprint arXiv:2107.06751*.
- Holly Else. 2023. Abstracts written by chatgpt fool scientists. *Nature*, 613(7944):423.
- Holly Else and Richard Van Noorden. 2021. The fight against fake-paper factories that churn out sham science. *Nature*, 591(7851):516–520.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. [DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing](#). In *The Eleventh International Conference on Learning Representations*.
- Digvijay Ingle, Rishabh Tripathi, Ayush Kumar, Kevin Patel, and Jithendra Vepa. 2022. [Investigating the characteristics of a transformer in a few-shot setup: Does freezing layers in RoBERTa help?](#) In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 238–248, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Rishivardhan K., Kayalvizhi S, Thenmozhi D., Raghav R., and Kshitij Sharma. 2020. [SSN-NLP at SemEval-2020 task 4: Text classification and generation on common sense context using neural networks](#). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 580–584, Barcelona (online). International Committee for Computational Linguistics.
- Yury Kashnitsky, Drahomira Herrmannova, Anita de Waard, George Tsatsaronis, Catriona Catriona Fennell, and Cyril Labbe. 2022. [Overview of the DAGPap22 shared task on detecting automatically generated scientific papers](#). In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 210–213, Gyeongju, Republic of Korea. Association for Computational Linguistics.
- Weixin Liang, Zachary Izzo, Yaohui Zhang, Haley Lepp, Hancheng Cao, Xuandong Zhao, Lingjiao Chen, Hao-tian Ye, Sheng Liu, Zhi Huang, et al. 2024. Monitoring ai-modified content at scale: A case study on the impact of chatgpt on ai conference peer reviews. *arXiv preprint arXiv:2403.07183*.
- Adaku Uchendu. 2023. *Reverse Turing Test in the Age of Deepfake Texts*. Ph.D. thesis, The Pennsylvania State University.
- Richard Van Noorden. 2021. Hundreds of gibberish papers still lurk in the scientific literature. *Nature*, 594(7862):160–161.
- Nikhil Wani, Sandeep Mathias, Jayashree Aanand Gajjam, and Pushpak Bhattacharyya. 2018. [The whole is greater than the sum of its parts: Towards the effectiveness of voting ensemble classifiers for complex word identification](#). In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 200–205, New Orleans, Louisiana. Association for Computational Linguistics.
- 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.