PetKaz at SemEval-2024 Task 8: Can Linguistics Capture the Specifics of LLM-generated Text?

Kseniia Petukhova, Roman Kazakov, Ekaterina Kochmar
Mohamed bin Zayed University of Artificial Intelligence
{kseniia.petukhova, roman.kazakov, ekaterina.kochmar}@mbzuai.ac.ae

Abstract

In this paper, we present our submission to the SemEval-2024 Task 8 "Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection", focusing on the detection of machine-generated texts (MGTs) in English. Specifically, our approach relies on combining embeddings from the RoBERTa-base with diversity features and uses a resampled training set. We score 12th from 124 in the ranking for Subtask A (monolingual track), and our results show that our approach is generalizable across unseen models and domains, achieving an accuracy of 0.91. Our code is available at https://github.com/sachertort/petkaz-semeval-m4.

1 Introduction

SemEval-2024 Task 8 "Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection" (Wang et al., 2024) has focused on the detection of machine-generated texts (MGTs). In recent years, large language models (LLMs) have achieved human-level performance across multiple tasks, showing impressive capabilities in natural language understanding and generation (Minaee et al., 2024), including their abilities to generate high-quality content in such areas as news, social media, question-answering forums, educational, and even academic contexts. Often, text generated by LLMs is almost indistinguishable from that written by humans, especially along such dimensions as text fluency (Mitchell et al., 2023). Therefore, methods of automated MGT detection, intending to mitigate potential misuse of LLMs, are quickly gaining popularity. Automated MGT detection methods can be roughly split into black-box and white-box types, with the former being restricted to API-level access to LLMs and reliant on features extracted from machine-generated and human-written text samples for classification model training, and the latter focusing on zero-shot AI text detection without any additional training (see Section 2).

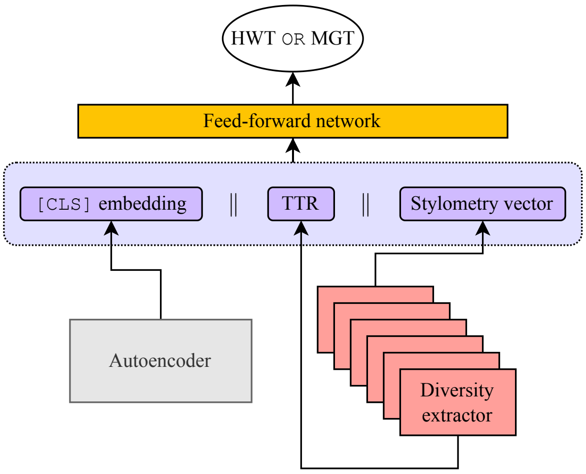

Figure 1: For each text, we get a [CLS] token embedding from an autoencoder model and extract vectors of linguistic features (e.g., lexical diversity, stylometry, etc.). Then, we pass the concatenated vector to a feed-forward network, whose output layer performs binary classification – HWT vs. MGT. The configurations of embeddings/features may vary between experiments.

For our submission to SemEval-2024 Task 8, the monolingual track of Subtask A, which focuses on MGT detection in English across a variety of domains and generative models, we have developed a system that can be categorized as a black-box detector and is based on a combination of embeddings, measures of lexical diversity, and careful selection of the training data (see Figure 1). We also present and discuss an extended set of linguistic features, including discourse and stylistic features, that we have experimented with during the development phase of the competition. The main motivation for using such a feature-based approach is that it helps us to focus on the fundamental differences between MGTs and human-written texts (HWTs) rather than capture the specifics of particular models.

Our results suggest that our best model, which uses diversity features and embeddings, outper-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1140–1147
June 20-21, 2024 ©2024 Association for Computational Linguistics

forms a very competitive baseline introduced in this task (Wang et al., 2024), yielding an accuracy of 0.95 on the development and 0.91 on the test set. It brought us 12th place out of 124 teams participating in the shared task. Furthermore, our investigation shows that a model using no embeddings but relying on such linguistic features as entity grid and stylometry yields results that are on par with the baseline model.

The main contributions of our work are as follows: **(1)** we investigate the impact on the detection task of a variety of linguistically motivated features, ranging from widely used stylometric features to novel ones, including those based on high-level discourse analysis; and **(2)** we show how training data can be selected in an informative way to help models better distinguish between MGTs and HWTs.

2 Related Work

A comprehensive survey by Yang et al. (2023) categorizes detection methods into training-based classifiers, zero-shot detectors, and watermarking techniques, covering both black-box and white-box detection scenarios. This survey discusses a range of strategies, including mixed training, proxy models, and semantic embeddings, indicating ongoing challenges in scalability and robustness. Given the fast development of LLMs and their capabilities, of particular interest are innovations in zero-shot detection methods highlighted by Mitchell et al. (2023) and Su et al. (2023). In addition, Mitchell et al. (2023) present DetectGPT, utilizing perturbation discrepancies to discern MGTs, while Su et al. (2023) propose DetectLLM-LRR and DetectLLM-NPR, which advance zero-shot detection by harnessing log rank information.

Another relevant line of research investigates the use of linguistic and stylometric features, such as the ones overviewed in Bergsma et al. (2012), for MGT detection. For instance, Wang et al. (2023) explore the use of logistic regression with GLTR features (analyzing the distribution of token probabilities and their relative frequencies within specific probability ranges from a language model's output), stylistic characteristics, and NELA news verification features (style, complexity, bias, affect, morality, and event specifics) on the M4 dataset, and Liu et al. (2022) introduce a model exploiting text coherence, named entities and relation-aware graph convolutional networks under a low-resource setting for MGT detection.

3 Methodology

Our general pipeline, visualized in Figure 1, consists of the following components: (1) an autoencoder model fine-tuned on HWT vs. MGT classification task; (2) linguistic features extraction pipeline; and (3) embeddings and features combination passed through a feed-forward neural network. Below we describe some of these components in more detail.

3.1 Embeddings

We employ an autoencoder model. First, we fine-tune it on the HWT vs. MGT classification task, and then we use its `[CLS]` tokens' embeddings in a feed-forward model.

3.2 Features

We study the impact on the classification accuracy of several types of linguistically motivated features extracted from texts, including those based on: 1) text statistics; 2) readability; 3) stylometry; 4) lexical diversity; 5) rhetorical structure theory (RST); and 6) entity grid. Below we provide a description of the features and their relevance to the task.

Text statistics We compute the following:[1] 1) the number of difficult words (words that have more than two syllables and are not in the list of easy words[2] from Dale and Chall, 1948); 2) raw lexicon count (unique words in text); 3) raw sentence count. In Appendix A.1, we provide the values for HWTs and across models.

Readability We assess the readability of MGTs and HWTs guided by the hypothesis that HWTs are easier to read than MGTs. We calculate a range of common readability scores for both types of texts to assess their readability, including 1) Flesch Reading Ease Test (Flesch, 1979); 2) Flesch-Kincaid Grade Level Test (Kincaid et al., 1975); and 3) Linsear Write Metric (O'Hayre, 1966).

Stylometry For stylometric features, we use the approach proposed in Bergsma et al. (2012). Specifically, we collect all unigrams and bigrams from the texts and keep punctuation, stopwords, and Latin abbreviations (e.g., *i.e.*) unchanged. Then, we build two types of representations where

[1] We use Python's `textstat` library: `https://pypi.org/project/textstat/`.

[2] `https://github.com/textstat/textstat/blob/main/textstat/resources/en/easy_words.txt`

other words are replaced by their PoS tags and "spelling signatures" (forms of words; e.g., xXX-dd for *iOS-17*).[3] Then, log token frequencies (TFs) are computed for each text and passed to the maximum absolute scaler, and these sparse representations are used as features. For further processing, sparse matrices with stylometry features are reduced by truncated singular value decomposition to a dimensionality of 768. See Appendix A.2 for the analysis of stylometry features importance.

Lexical diversity Lexical diversity tells us how "rich" texts are in terms of vocabulary, i.e., whether they use rare words, or include a wide range of synonyms, epithets, terms, etc. There are a few measures widely used to measure lexical diversity, mostly based on the variants of the type-token ratio (TTR). We extract 10 features, such as TTR, Maas TTR, Hypergeometric distribution d (HDD; McCarthy and Jarvis, 2007), etc.[4] For an in-depth overview, see McCarthy and Jarvis's (2010) study on lexical diversity assessment.

RST features In rhetorical structure theory (RST), proposed in Mann and Thompson (1988), texts are analyzed in terms of hierarchical structures, which represent the organization of information and text flow. These structures are made up of elementary discourse units (EDUs) connected through rhetorical relations, which include "elaboration", "contrast", "cause", "result", etc. Using an open-source sentence-level RST parser (Lin et al., 2019), we count the occurrences of various relations in each text and divide them by the total number of sentences in the text.

Entity grid Finally, we use the entity grid algorithm to analyze the coherence of text by capturing patterns of entity distribution (Barzilay and Lapata, 2005). This method transforms a text into sequences of entity transitions, documenting the distribution, syntax, and reference information of discourse entities. Entities from texts are first tagged with their syntactic roles[5] and categorized into three types: subject (s), object (o), and other (x). The next step involves examining the transition of entities' roles across consecutive sentence pairs. This includes transitions like subject-to-object, object-to-other, subject-to-none, among others. Finally, we calculate the frequency of each transition type for all entities by dividing the total count of each transition type by the number of sentence pairs.

3.3 Feed-forward neural network

Finally, we use a concatenation of embeddings and vectors representing combinations of various features described above and pass them as input to a feed-forward neural network. Then, the output layer performs binary classification.

4 Data

Shared task organizers have used an extension of the M4 dataset (Wang et al., 2023),[6] which covers a range of domains (including *WikiHow*, *Wikipedia*, *Reddit*, *arXiv*, *PeerRead*, and *Outfox*) and texts generated by a number of LLMs (including ChatGPT, Cohere, Davinci003, Dolly-v2, BLOOMZ, and GPT-4) as well as written by humans. Overall, the training set is roughly balanced between HWTs and MGTs, with 53% being HWTs and with the number of HWTs being around 5 times higher than that of texts generated by any single LLM for each of the domains. The only exception is *PeerRead*, where the distribution of texts generated by each LLM and written by humans is about the same. At the same time, the distribution is exactly 50%:50% for HWTs:MGTs in the development set, and 47.5%:52.5% for HWTs:MGTs in the test set. In addition, while both training and development sets cover a range of domains, the test set is limited to *Outfox* only.

A curious case of WikiHow Before running the experiments, we further investigate how the training data is composed. According to Wang et al. (2023), LLMs were provided with relatively short inputs to generate texts across various domains: for example, with titles for *Wikipedia* articles and *arXiv* papers, titles and abstracts for *PeerRead* articles, etc. On the one hand, we observe a high level of parallelism in the training data across HWTs and texts generated by various models, and on the other, we note that there is little consistency in what models generate in certain domains: for example, provided with a name of a personality they generate quite different *Wikipedia* entries, which do

[3] The pre-processing was done using spaCy: https://spacy.io.

[4] Using Python's lexical_diversity library: https://github.com/kristopherkyle/lexical_diversity.

[5] Noun coreference is resolved using spaCy (https://spacy.io) and neuralcoref (https://spacy.io/universe/project/neuralcoref).

[6] https://github.com/mbzuai-nlp/M4

not only differ from the correspondent HWTs but also vary from one LLM to another (see examples in Appendix B, Table 5). In contrast, texts in the *WikiHow* domain appear to be more similar to each other across LLMs, which can be explained either by the way the data was generated (using titles and headlines as prompts to produce MGTs) or by the fact that there are fewer ways to explain *How to do X?* compared to the tasks in other domains. Moreover, our experiments with in-domain training of the MGT detection classifier suggest that the best results can be obtained when it is trained on the *WikiHow* domain. We follow up on these observations and create a customized training subset by using all MGTs from the original data and limiting HWTs to the texts from the *WikiHow* domain only. This results in a training set of 56,406 MGTs and 15,499 HWTs, with the distribution between each LLM and humans being roughly 1:1.

5 Experiments

5.1 Experimental setup

As the source of embeddings, we use `roberta-base`[7] (Liu et al., 2019) fine-tuned within the baseline framework[8] over 3 epochs with the learning rate of 2e-5 and L_2 norm of the weights being 0.01. The feed-forward neural network with two hidden layers accompanied by a ReLU activation function is then trained with the learning rate 5e-5, L_2 norm of the weights 0.01, and early stopping after 25 epochs. Each hidden layer has batch normalization and a dropout of 0.5. We use `PyTorch`[9] (Paszke et al., 2019) for all training and evaluation steps.

Following up on our observations on the *WikiHow* subset described in Section 4, we conduct two series of experiments and train the feed-forward network on: 1) the *full* training set; and 2) the *reduced* training set where we use MGTs from all domains and HWTs from *WikiHow* only.

5.2 Experiments on the development set

The evaluation results of our model with different feature configurations applied to the development set are presented in Table 1. Several observations are due at this point.

Configuration	Full train	Reduced train
`feat`	0.60	0.60
`sty`	0.68	0.57
`sty` ‖ `feat`	0.69	0.60
`sty` ‖ `div`	0.65	0.72
`sty` ‖ `read`	0.67	0.61
`sty` ‖ `rst`	0.64	0.57
`sty` ‖ `ent`	0.73	0.56
`emb`	0.74	0.83
`emb` ‖ `sty`	0.73	0.82
`emb` ‖ `feat`	0.76	0.90
`emb` ‖ `div`	0.73	**0.95**
`emb` ‖ `read`	0.72	0.81
`emb` ‖ `rst`	0.73	0.81
`emb` ‖ `ent`	0.73	0.82
Baseline	0.74	–

Table 1: Accuracy of different configurations and the baseline on the development set. `feat` stands for all features except stylometry, `sty` – stylometry, `div` – lexical diversity, `read` – text statistics and readability, `rst` – RST, `ent` – entity grid, `emb` – embeddings (see Section 3.2).

First of all, we note that the highest accuracy of 0.95 is achieved with the model trained on the *reduced* training set using a combination of embeddings and diversity features. This does not mean that lexical diversity is necessarily the most powerful among linguistic features, but it suggests that it complements embedding representations better than other linguistic features. Moreover, it is the only feature type that increases the accuracy obtained with embeddings only. Finally, we also note that with the linguistic features, our model can outperform a competitive baseline used by the task organizers, which sets the accuracy at 0.74.

Secondly, stylometry features turn out to be the best linguistic feature type when used on their own: the accuracy with `sty` is 0.68 vs. 0.6 with `feat`. These representations reflect some general patterns of word types used in texts. However, it seems like they alone are not enough for effective classification, at least when applied to texts generated by modern LLMs. Notably, the configuration that combines stylometry with entity grid features (`sty + ent`) demonstrates performance that is nearly identical to the baseline employing a pre-trained language model (0.73 vs. 0.74), suggesting that entity grid adds further information about text coherence. Other features like RST do not seem to help distinguish MGTs from HWTs. This finding

[7] `https://huggingface.co/FacebookAI/roberta-base`

[8] `https://github.com/mbzuai-nlp/SemEval2024-task8/tree/main/subtaskA/baseline`

[9] `https://pytorch.org`

Configuration	Train	Accuracy	F$_1$
`emb` ‖ `div`	reduced	**0.91**	**0.92**
`sty` ‖ `ent`	full	0.84	0.85
Baseline	full	0.88	–

Table 2: Metrics on the test set. The first row is our main submitted configuration. The organizers do not report only the baseline's F_1 score.

suggests that the frequency or efficacy with which humans and models employ rhetorical structures is comparable.

Finally, we observe that the performance of the model using `emb` features always increases if it is trained on the *reduced* set. This determines the model configuration for our final submission.

6 Results

Table 2 presents accuracy on the test set obtained with two configurations: a model using embeddings and lexical diversity features trained on the reduced training set, and a model using stylometry and entity grid features trained on the full training set, which showed promising results on the development set. **The former one is our main configuration: our team has submitted its predictions for the test set and scored 12th in the shared task (out of 124 teams).** This model outperforms the organizers' baseline, which sets the accuracy at 0.88. However, we note that the latter model, which relies on linguistic features only and does not employ any pre-trained language model, also shows promising results, further strengthening our hypothesis that linguistic features are able to capture important properties of LLM-generated texts.

6.1 Analysis

We further analyze the performance of our best model across different LLMs on the test set, as illustrated in Figure 2. The results show that our model accurately identifies texts from `Dolly-v2`, `Cohere`, and `ChatGPT` as machine-generated, and achieves near-perfect classification precision on texts from `GPT-4` and `Davinci003`. `BLOOMZ` is the only model that presents a problem for our classifier, with an 8% misclassification rate. Additionally, we observe that 18% of HWTs are incorrectly classified as being generated by machines. This shows the remarkable generalizability of our approach compared to Wang et al. (2023), who reported that “it is challenging for detectors to generalize well on unseen examples if they are either from different domains or are generated by different large language models. In such cases, detectors tend to misclassify machine-generated text as human-written”.

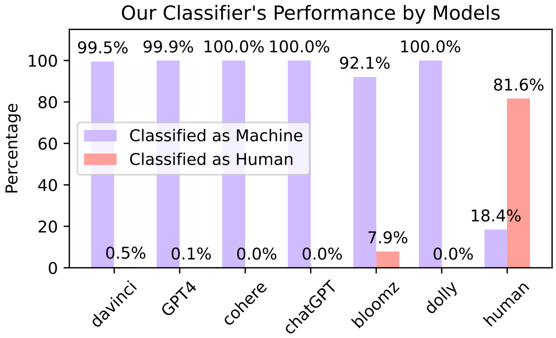

Figure 2: Performance of our classifier across models.

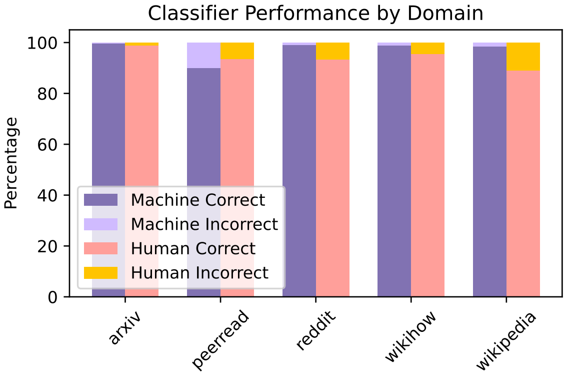

Figure 3: Performance of our classifier across domains (on the development set).

Furthermore, we evaluate our model's performance across domains (Figure 3). Our analysis reveals that we can accurately identify all MGTs and nearly perfectly recognize HWTs from *arXiv*. Our classifiers face the biggest difficulties when classifying MGTs from *PeerRead* and HWTs from *Wikipedia*. These results are aligned with those reported in Wang et al. (2023), who also found that training on *Wikipedia* leads to the worst out-of-domain accuracy.

In summary, our classifier demonstrates generalizability, performing well on both previously unseen models (`GPT-4` and `BLOOMZ`) and domains (with all texts in the test set being from *Outfox*).

7 Conclusions

When developing the models for our submission to the SemEval-2024 Task 8, we have primarily focused on: (1) the contribution of linguistic features to the task, and (2) the selection of the informative training data. Our results suggest that models using only linguistic features (specifically, those based on stylometry and entity grid) can perform competitively on this task, while careful selection of the training data helps improve the performance of the models that rely on embeddings. This shared task demonstrates that it is possible to distinguish between HWTs and MGTs, but the results also suggest promising avenues for future research, including in-depth analysis of the training data selection techniques and expansion of the linguistic features.

Limitations

Our work is limited to the English language only as we opted to participate in a single Subtask of SemEval-2024 Task 8. In addition, this work is only limited to the domains and LLMs included in the shared task data, therefore, the generalizability of our approach beyond these domains and LLMs will need to be verified in future experiments.

Acknowledgements

We would like to express our gratitude to Ted Briscoe for inspiring us with the idea that linguistics could be of help and for engaging in discussions with us. We are grateful to Mohamed bin Zayed University of Artificial Intelligence (MBZUAI) for supporting this work. We also thank the anonymous reviewers for their valuable feedback.

References

Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: An entity-based approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 141–148, Ann Arbor, Michigan. Association for Computational Linguistics.

Shane Bergsma, Matt Post, and David Yarowsky. 2012. Stylometric analysis of scientific articles. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 327–337, Montréal, Canada. Association for Computational Linguistics.

Edgar Dale and Jeanne S. Chall. 1948. A formula for predicting readability. *Educational Research Bulletin*, 27(1):11–28.

Rudolf Franz Flesch. 1979. *How to write plain English: a book for lawyers and consumers*. University of Canterbury.

Peter Kincaid, Robert P. Fishburne, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Xiang Lin, Shafiq Joty, Prathyusha Jwalapuram, and M Saiful Bari. 2019. A unified linear-time framework for sentence-level discourse parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4200, Florence, Italy. Association for Computational Linguistics.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Yu Lan, and Chao Shen. 2022. CoCo: Coherence-enhanced machine-generated text detection under data limitation with contrastive learning.

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.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Philip McCarthy and Scott Jarvis. 2010. MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42:381–92.

Philip M. McCarthy and Scott Jarvis. 2007. vocd: A theoretical and empirical evaluation. *Language Testing*, 24(4):459–488.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. DetectGPT: Zero-shot machine-generated text detection using probability curvature.

John O'Hayre. 1966. *Gobbledygook Has Gotta Go*. U.S. Department of the Interior, Bureau of Land Management.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

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

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. *ArXiv preprint*, abs/2306.05540.

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. 2024. 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, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. *ArXiv preprint*, abs/2305.14902.

Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda Petzold, William Yang Wang, and Wei Cheng. 2023. A survey on detection of LLMs-generated content. *ArXiv preprint*, abs/2310.15654.

A Features Analysis

A.1 Text statistics across models

Table 3 shows various text statistics calculated on the training set. It can be seen that HWTs have higher values than all MGTs across all these metrics.

Model	**DW**	**LC**	**SC**
ChatGPT	64	350	19
Cohere	37	256	13
Davinci003	58	315	16
Dolly-v2	54	342	18
Human	91	582	30

Table 3: Text statistics on the training set. DW = difficult words (mean), LC = lexicon count (mean), SC = sentence count (mean).

A.2 Stylometry features importance

Stylometry features are passed to a linear SVM classifier[10] to extract coefficients that may be interpreted as feature importances. Table 4 presents the most important features for MGTs and HWTs in the case of binary classification: for example, we can see that proper nouns are mostly associated with HWTs. It also makes it clear how the features are ordered by importance.

MGT feature	**Wt.**	**HWT feature**	**Wt.**
How to	3.28	NOUN SPACE	-4.12
SPACE *How*	2.34	SPACE	-4.12
NUM VERB	2.07	XXXX	-3.93
Xxxxx *the*	2.00	SPACE ADJ	-3.10
How	1.78	SPACE PROPN	-3.10
SPACE NUM	1.77	*the* SPACE	-2.67
Well	1.57	NOUN	-2.57
Xxx *the*	1.38	NUM SPACE	-2.21
dd Xxxxx	1.37	PROPN SPACE	-2.15
NOUN *you*	1.37	_XXX_d	-2.14

Table 4: Stylometric features highly weighted by the binary SVM classifier.

B Data Statistics

Table 5 shows some examples of parallel texts extracted from three domains represented in the training set (*WikiHow*, *Wikipedia*, and *PeerRead*). As explained in Wang et al. (2023), the data for each

[10] From scikit-learn (Pedregosa et al., 2011): https://scikit-learn.org.

	WikiHow
ChatGPT	Buying Virtual Console games for your Nintendo Wii is a fun and easy process that can net you some classic games to play on your console. [...]
Cohere	How to Buy Virtual Console Games for Nintendo Wii The Nintendo Wii has a feature called the Virtual Console that allows you to download and play games from past Nintendo consoles, such as the Nintendo Entertainment System. [...]
Davinci003	How to Buy Virtual Console Games for Nintendo Wii Most people know that Nintendo's library of classic titles is available on the Wii platform through the Virtual Console. [...]
Dolly-v2	Find a few Wii Points cards from game retailers like GameStop., Make sure your Wii is online and on a secure connection if possible. [...]
Human	They are about $20 a card. Or, if you want to just buy points with your credit card, Skip down to the section, With a Credit Card. [...]
	Wikipedia
ChatGPT	William Whitehouse was a 19th-century British engineer and inventor who made significant contributions to the field of hydraulics. [...]
Cohere	William Whitehouse (1567-1648) was an English scholar, schoolmaster, and Anglican clergyman. [...]
Davinci003	William Whitehouse (August 6, 1590 - May 18, 1676) was an English priest, scholar and biblical commentator. [...]
Dolly-v2	William Whitehouse (born William John Whitehouse; 15 July 1944) is an English musician, singer and songwriter. [...]
Human	William Edward Whitehouse (20 May 1859 - 12 January 1935) was an English cellist. [...]
	PeerRead
ChatGPT	The paper "End-to-End Learnable Histogram Filters" aims to introduce a novel approach that enables histogram filters to be learnable end-to-end. [...]
Cohere	This paper addresses the problem of designing end-to-end learnable histogram filters. [...]
Davinci003	This paper presents an interesting approach to combining problem-specific algorithms with machine learning techniques to find a balance between data efficiency and generality. [...]
Dolly-v2	The paper End-to-End Learnable Histogram Filters demonstrates an interesting technique for reducing photo noise without blurring the image. [...]
Human	We are retracting our paper "End-to-End Learnable Histogram Filters" from ICLR to submit a revised version to another venue. [...]

Table 5: Parallel HWTs and texts generated by different LLMs in the training set extracted from selected domains.

domain was generated in a different way (for instance, using an article title only in some cases, and more extended inputs in others). We observe that there is much less consistency between the outputs generated by different LLMs in such domains as *Wikipedia* and *PeerRead* than in *WikiHow*. For instance, in the case of generated *Wikipedia* articles, the models cannot even agree on what personality they are describing (which is obvious from the very first sentences of such generated articles), while in the case of generated reviews from *PeerRead*, article descriptions also exhibit high diversity in the way they are presented in the review. At the same time, we hypothesize that generating texts for the *WikiHow* domain, describing *How to do X?*, results in higher consistency in the models' outputs, which is exemplified in Table 5.