Proceedings of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 603–608
DOI: 10.15439/2024F8906
ISSN 2300-5963 ACSIS, Vol. 39

Impact of Spelling and Editing Correctness on Detection of LLM-Generated Emails

Paweł Gryka, Kacper Gradoń, Marek Kozłowski, Miłosz Kutyła, Artur Janicki
0009-0002-8505-2098
0000-0003-0750-8678
0000-0002-6313-8387
0009-0002-0947-8986
0000-0002-9937-4402
Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
Email: {Pawel.Gryka.stud, Kacper.Gradon, Marek.Kozlowski, Milosz.Kutyla.stud, Artur.Janicki}@pw.edu.pl

***Abstract*—In this paper, we investigated the impact of spelling and editing correctness on the accuracy of detection if an email was written by a human or if it was generated by a language model. As a dataset, we used a combination of publicly available email datasets with our in-house data, with over 10k emails in total. Then, we generated their "copies" using large language models (LLMs) with specific prompts. As a classifier, we used random forest, which yielded the best results in previous experiments. For English emails, we found a slight decrease in evaluation metrics if error-related features were excluded. However, for the Polish emails, the differences were more significant, indicating a decline in prediction quality by around 2% relative. The results suggest that the proposed detection method can be equally effective for English even if spelling- and grammar-checking tools are used. As for Polish, to compensate for error-related features, additional measures have to be undertaken.**

I. Introduction

ONE of the most serious problems associated with the developments in Information Technologies and their public availability today is the detection of content generated by Artificial Intelligence (AI). Apart from the substantial benefits introduced by public AI applications (especially in such fields as medical imaging diagnostics, data analysis, or automated translation), there is also a set of undeniable challenges caused by the constantly increasing difficulty in distinguishing between human and machine-generated text, images, video, and audio.

These problems are highlighted by transnational law-enforcement institutions, such as Europol [1], the intelligence and national security community [2] or public health policy-makers and researchers [3], who emphasize the threats related to the application of the Generative Artificial Intelligence (GAI) for the creation of disinformation, sophisticated scams, social engineering and political manipulation. The GAI-related challenges do not have to be linked to high-profile security domains only. An important and worrying abuse of technology can also be seen in the academic world, where GAI brings plagiarism and academic dishonesty to an entirely new level. Early detection and flagging of high-quality, language-agnostic content produced by AI tools require urgent research and development efforts and the creation of high-quality detection tools.

Research was funded by the Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) programme.

This paper is a continuation of our efforts to recognize LLM-generated texts, initially focused on email messages. The preliminary results of our experiments have been described in our previous work [4]. The detection results yielded F1-scores of almost 0.98 for English and over 0.92 for Polish. It turned out that the detection algorithm strongly relied on sentence statistics, such as the average word and sentence length, as well as on typographical and orthographic (spelling) imperfections. However, those experiments did not consider that the analyzed text might have undergone spelling and grammar checks. In this study, we would like to check what the actual impact of those errors on detection accuracy is.

Our paper is structured as follows: first, in Section II, we summarize related work in this field. In Section III, we describe our approach. Section IV presents the methodology of our experiments. Results are shown and discussed in Section V, followed by conclusions in Section VI.

II. Related work

Human communication is increasingly flooded by AI-generated texts. LLMs suggest words and paragraphs or produce entire essays across chat, email, and social media. Therefore, there is a huge need for an effective method of detecting LLM-generated texts (LLMGT).

Several approaches for LLMGT detection have been suggested and explored. Some researchers proposed watermarking or registering AI-generated content. The main idea of these approaches is that any organization developing a foundation model intended for public use must demonstrate a reliable detection mechanism for the content it generates as a condition of its public release. Knott et al. [5] proposed using watermarking as a solution for detecting LLMGT. The authors claim that searching for watermarks can be very effective. Another approach [6] relies on retrieving semantically-

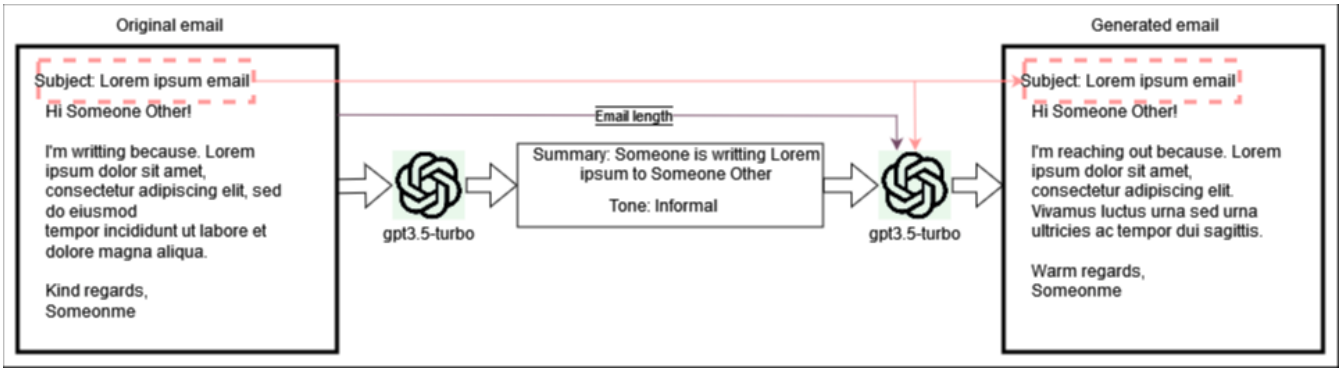


Fig. 1. Process of creating LLM-generated version of email.

similar generations from a huge database with language model historical outputs.

Many researchers use feature-based detection, which seems to be the most straightforward. It consists of extracting various text characteristics in hopes of finding differences between human-written and generated texts. Then, those characteristics are used to train a machine learning (ML) model that will be able to analyze them and produce classification results. Many possible characteristics can be measured; those include stylometry features like word frequency and sentence structure, LLM-specific features like perplexity [7] (measure representing how "surprised" the specific LLM model is when seeing the given text; human-written texts are usually characterized with higher perplexity scores than LLM generated ones) and burstiness [8] (metric based on words' distribution and the variance of sentence length in the text; humans tend to write with a less consistent style than an LLM might). Their effectiveness was demonstrated by Cingillioglu [9], who used linguistic, semantic, and stylistic features to train a support vector machine (SVM) classifier and got over 92% accuracy when detecting generated essays. This approach was also used by Fröhling and Zubiaga [10], who observed that AI-generated texts exhibit 1) lack of syntactic and lexical diversity, 2) repetitiveness (tendency to overuse frequent words), 3) lack of coherence, and 4) lack of clear purpose or focus.

Another group of methods uses LLMs themselves to detect LLM-generated texts. The first set of methods applies LLMs as they are, i.e., without any training steps. Shi et al. [11] introduced Proxy-Guided Efficient Re-sampling (POGER). It worked by selecting a subset of unusual keywords, i.e., characterizing low probabilities of appearing in their contexts according to the given LLM. Then, the text is re-sampled, namely, each identified unusual keyword is removed, and the LLM is prompted to fill in the gap. If the resulting text is similar enough, the original text was likely LLM-generated.

Mitchell et al. [12] defined a new curvature-based criterion for judging if a passage is generated from a given LLM. This approach, called DetectGPT, does not require training a separate classifier or collecting a dataset of real or generated passages. It uses only log probabilities computed by the LLM of interest and random perturbations of the passage from another generic pre-trained language model. To classify a candidate passage, DetectGPT first generates minor perturbations of the passage using a generic pre-trained model such as T5. DetectGPT compares the log probability under p of the original sample with each perturbed corresponding sample. If the average log ratio is high, the sample is likely AI-generated content. The main contribution of this work was to identify a property of the log probability function computed by a wide variety of large language models, showing that a tractable approximation to the trace of the Hessian of the model's log probability function provides a useful signal for detecting model samples.

There are also separate approaches based on fine-tuning the LLMs to classify whether the text is AI-generated. Harrag et al. [13] fine-tuned a BERT model, specifically AraBERT, to differentiate between human-written and AI-generated Arabic tweets, primarily produced by GPT-2. They achieved very promising results with an F1-score equal to 98.7%. Rodriguez et al. [14] also trained a BERT-based model to identify texts that were fully or partially created by AI. They showed that one can fine-tune a RoBERTa model with texts from one scientific domain, and it will still accurately detect AI-generated texts from another domain, provided that a few samples from the new field are used in the fine-tuning process.

III. Proposed Method

In the current study, we followed the procedure outlined in our previous paper [4]. We used the feature-based approach and employed a binary classifier, training it to detect LLM-generated emails. As the training data, we used original emails and their LLM-generated versions. From each email, we extracted various features, such as 1) token-level perplexity (1 feature), measuring how likely the chosen LLM is to generate the input text sequence [7], 2) burstiness (1 feature), accounting for words' distribution and occurrence patterns in a generated text [15], 3) distribution of sentence length (6 features: average, standard deviation and variance of sentence length in words and characters, 4) average word char length (1 feature), 5) punctuation metrics (2 features), counting the number of punctuation marks (.,;:!?) per number of sentences

TABLE I
TEN MOST DISCRIMINATIVE FEATURES FOR DETECTING LLM-GENERATED EMAILS, FOR ENGLISH AND POLISH LANGUAGE, SORTED BY THEIR IMPORTANCE.

Rank	Features	
	English	Polish
1	number_of_errors	punctuation_per_sentence
2	no_space_after_punctuation	number_of_errors
3	stdev_sentence_char_length	stdev_sentence_char_length
4	variance_sentence_char_length	variance_sentence_char_length
5	variance_sentence_word_length	variance_sentence_word_length
6	stdev_sentence_word_length	stdev_sentence_word_length
7	double_spaces	no_space_after_punctuation
8	text_errors_by_category.typos	number_of_sentences
9	punctuation_per_sentence	text_errors_by_category.typos
10	average_word_char_length	double_spaces

and per number of characters, 6) general statistics (3 features), such as the number of characters, words, and sentences, 7) *Stylometrix* features (172 features for Polish and 196 features for English), describing stylometric characteristics obtained using the `StyloMetrix` library [16]. 8) emotion-related features (5 features), such as the use of emojis, the number of question/exclamation marks, and occurrences of multiple question/exclamation marks (e.g., ??, !!,?!?).

We also extracted 26 features related to errors in text. Most of them were extracted using the Python library `language-tool-python` [17]:

- *Editing-related errors (17 features)*: features capturing a variety of typographical and stylistic errors. We counted mistakes like missing spaces after punctuation marks, double spaces between words, inconsistencies in the use of American and British English conventions, errors related to incorrect use of uppercase and lowercase letters, awkward word combinations (collocations), and incorrect word order. This category also encompasses issues such as unnecessary repetition of words, improper punctuation, and errors in forming compound words.
- *Spelling-related errors (4 features)*: we counted general spelling mistakes, probable typos, and errors involving the incorrect spelling of multi-word phrases.
- *Grammar-related errors (2 features)*: number of mistakes related to the rules of grammar, such as subject-verb agreement and sentence structure.
- *Other (3 features)*: general count of errors, miscellaneous errors, and semantic errors.

In total, 241 features for English and 217 for Polish were extracted for each email text. In this study, we aimed to assess the impact of spelling, grammar, and editing-related features on the detection of LLM-generated emails.

IV. EXPERIMENTS

In this work, we analyzed detection performance for various feature groups to find out what impact the features related to spelling, grammar, and editing have on LLMGT detection accuracy. The study setup is similar to the one used in our previous work [4]. Since our previous experiments revealed that a random forest classifier with 100 trees yielded the best results, we used it exclusively here. All experiments were conducted using `scikit-learn` [18] library version `1.4.2`, following a 10-fold cross-validation scheme.

As for the email data, we used three publicly available email datasets: "Spam email dataset" [19], containing email subjects and their content in plain text, "Email classification dataset" [20], and "The Spam Assassin Email Classification Dataset" [21]. Out of them, we obtained a set of 20156 emails.

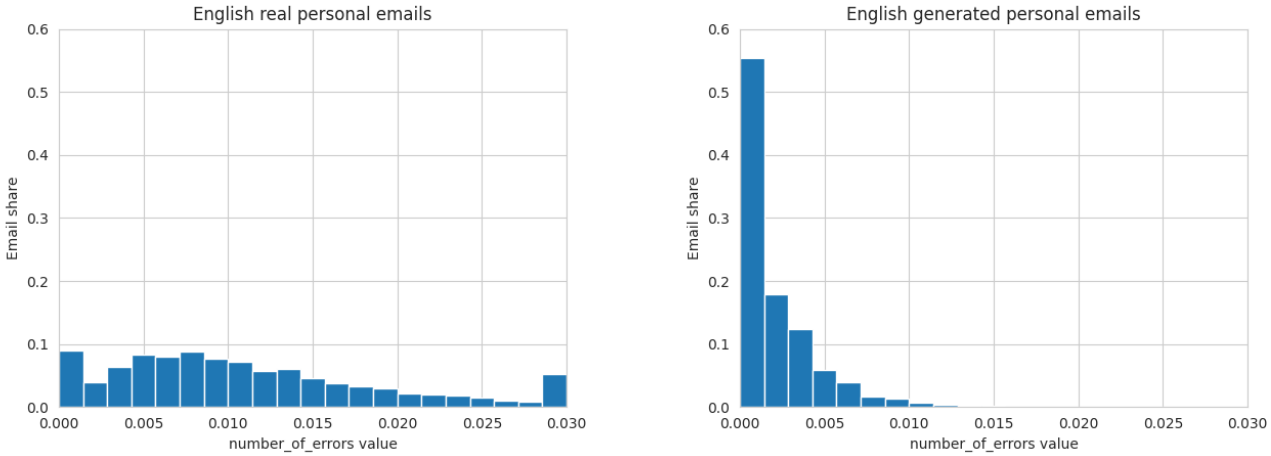
Since we also wanted to detect email messages in Polish, we had to add our in-house data. These data contained 38776 emails, both in Polish and English. Next, we filtered out emails with less than ten characters of content and those that were created later than 2022 to be sure that none of the widely used LLMs (such as GPT-3.5) generated them. We also filtered out spam and advertisement emails to focus just on emails that can be considered as human conversations. Eventually, we obtained a dataset with 9885 original (i.e., human-written) emails in English and 471 emails in Polish.

Next, we created a "mirror" dataset with LLM-generated email texts that closely resembled the content (both in terms of the topic and the sentiment) of human-written emails. Every generated email was based on a single real email (see the generation scheme shown in Fig. 1). Through OpenAI API, we provided the email's subject and body in plaintext and prompted `gpt-3.5-turbo-0125` model to shortly summarize the email and classify the email's tone as either `formal`, `neutral`, `informal`. Next, we took the summary and the tone of the email, and we prompted the same model to generate a complete email based on that information. This way, we created a dataset with generated emails. Noteworthy, they were not simple paraphrases of original emails, but new emails of roughly similar size, generated based on a short summary of original emails.

V. RESULTS

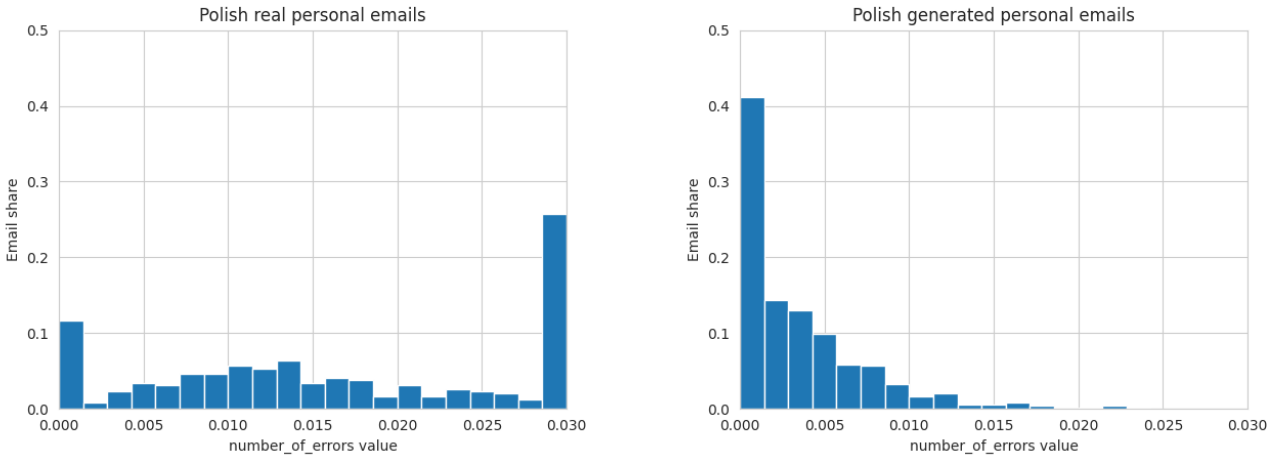
We evaluated the detection ability for various feature groups using standard metrics, such as accuracy (ACC), precision, recall, F1-Score, and the area under the curve (AUC).

Table I presents the most discriminative features, identified according to the mutual information (MI) value. It confirms what was initially stated, following our previous paper [4], that the detection relied strongly on the features related to various types of mistakes: the total number of mistakes is ranked #1



(a) Original emails (b) Generated emails

Fig. 2. Histograms for `number_of_errors` feature, for a) original and b) generated emails in English.



(a) Original emails (b) Generated emails

Fig. 3. Histograms for `number_of_errors` feature, for a) original and b) generated emails in Polish.

for English and #2 for Polish. It is also very well visible in histograms displayed in Figures 2 and 3, for emails in English and Polish, respectively. They show that real (original) emails exhibit a rather uniform distribution of errors, i.e., there is a remarkable number of emails with multiple errors. In contrast to that, most of the generated emails are either error-free or exhibit a low number of mistakes. The number of errors seems to decrease exponentially.

Since users often use tools to check spelling and grammar, we wanted to see if detecting LLM-generated text on "dirty" and "clean" texts makes a remarkable difference. Table II shows the detection metrics for various groups of features used by the classifier. For English and Polish, detection accuracy and other metrics decreased when error-related text features were removed, yet the decrease was not uniform. The F1-score for English, after removing 26 error-related features, decreased from 0.9885 to 0.9833, so the drop was only minor (around 0.5% relative). Similar minor deterioration was observed for other metrics. However, the drop for Polish was more visible: the F1-score decreased from 0.9484 to 0.9235, i.e., by 2.5% relative.

Table II displays that removing editing-related parameters (such as missing spaces, double spaces, or incorrect casing) contributed most to this drop in detection performance for Polish. As for English, all three groups had a similar, minor impact on the evaluation metrics.

We also made an interesting observation when we selected the 10 best features and trained the detection classifier in a 10-dimensional space. When using the 10 best features for English (the list is shown in Table I), we were able to create

TABLE II
RESULTS OF DETECTION OF LLM-GENERATED EMAILS FOR VARIOUS GROUPS OF FEATURES

Language	Features	# Features	Accuracy	Precision	Recall	F1 Score	ROC AUC
English	All	244	0.9882	0.9923	0.9848	0.9885	0.9995
	All but editing-related	224	0.9859	0.9901	0.9825	0.9863	0.9991
	All but spelling-related	237	0.9861	0.9905	0.9827	0.9866	0.9992
	All but grammar-related	239	0.9878	0.9916	0.9847	0.9881	0.9994
	All but error-related	218	0.9828	0.9878	0.9790	0.9833	0.9987
	10 best of all	10	0.9884	0.9931	0.9845	0.9888	0.9995
	10 best, no error-related	10	0.9700	0.9798	0.9617	0.9707	0.9948
Polish	All	220	0.9461	0.9375	0.9607	0.9484	0.9868
	All but editing-related	200	0.9192	0.8942	0.9549	0.9227	0.9773
	All but spelling-related	213	0.9389	0.9278	0.9560	0.9413	0.9857
	All but grammar-related	215	0.9410	0.9355	0.9524	0.9432	0.9876
	All but error-related	194	0.9202	0.8935	0.9575	0.9235	0.9747
	10 best of all	10	0.9202	0.9185	0.9248	0.9211	0.9774
	10 best, no error-related	10	0.8953	0.8792	0.9226	0.8989	0.9624

TABLE III
TEN MOST DISCRIMINATIVE FEATURES FOR DETECTING LLM-GENERATED EMAILS, FOR ENGLISH AND POLISH LANGUAGE, SORTED BY THEIR IMPORTANCE, ASSUMING EDITING, SPELLING, AND GRAMMAR CORRECTNESS.

Rank	Features	
	English	Polish
1	variance_word_char_length	punctuation_per_sentence
2	stdev_word_char_length	stdev_sentence_word_length
3	stdev_sentence_char_length	stdev_sentence_char_length
4	variance_sentence_char_length	variance_sentence_char_length
5	stdev_sentence_word_length	variance_sentence_word_length
6	variance_sentence_word_length	number_of_sentences
7	stylometrix_statistics_ST_SENT_D_NP	variance_word_char_length
8	stylometrix_statistics_ST_SENT_D_PP	stdev_word_char_length
9	punctuation_per_sentence	average_sentence_char_length
10	average_word_char_length	average_sentence_word_length

a detection model of the same detection efficacy as for the full 244-feature space. However, if we removed error-related features, a classifier working in such a feature space would have accuracy and the F1-score lower by 1.8% and 1.5% relative, respectively. To compensate for the loss of error-related features, at least 20-feature space would be needed (see the ACC and AUC values against the number of features shown in Fig. 4).

As for Polish, using only 10-best features would yield results lower than for the full feature space by more than 2.5%. At least 30 features would be required to achieve the accuracy results as for the full feature set (see Fig. 5). After removing error-related features, the 10 best feature space allows the detection with the metrics by around 5% relative lower than for the full set. However, using the 20 best non-error-related features seems optimal for Polish and yields better accuracy even than for the complete feature set. Yet, the results are clearly inferior to those for English, which partially can be a consequence of a much smaller size of the Polish email dataset and partially of different characteristics of the tools used for Polish.

Table III displays the 10 most discriminative non-error-related features. One can see that statistical parameters related to word and sentence length, as well as the number of punctuation marks per sentence (which is correlated with sentence length, especially for Polish), exhibited the highest discriminative power when detecting LLM-generated emails in the absence of error-related features. As for English, also the usage statistics of noun phrases (NP) and prepositional phrases (PP) turned out to be important.

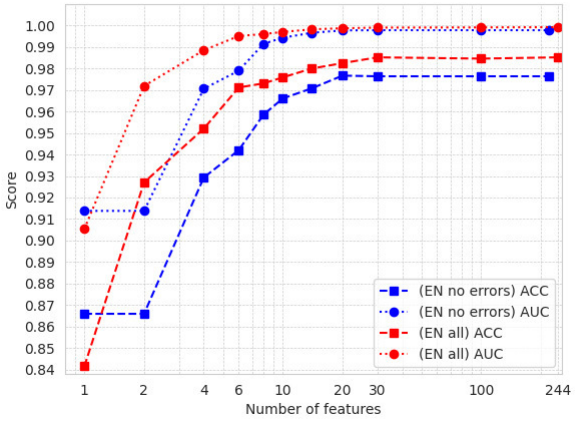


Fig. 4. Comparison of ACC and AUC for detection with and without error-related features for English

VI. CONCLUSIONS

In our paper, we have expanded upon the foundational work presented in [4], and we investigated the impact of spelling and

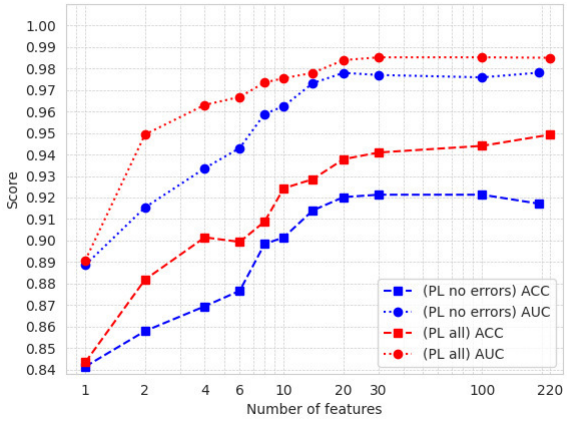


Fig. 5. Comparison of ACC and AUC for detection with and without error-related features for Polish

editing correctness on detection accuracy, motivated by the observed significance of these factors. Our findings reinforce the observations and provide further insights into this dependency. The code used in the experiments and the complete feature list have been made public[1].

For English texts, when using a limited number of features, we noted a slight decrease in ACC and AUC for the feature set, excluding error-related features; however, this difference became less relevant as more features were used. In contrast, for the Polish emails, the differences were more significant even when full feature lists were used, indicating a decline in prediction quality by around 2% relative for ACC, AUC, and F1-scores. The results indicate that the proposed detection method can be equally effective for English even if spelling- and grammar-checking tools are used. As for Polish, to compensate for the "loss" of text errors, we need to use more features and, potentially, also to seek new ones.

REFERENCES

[1] Europol, "Chatgpt: the impact of large language models on law enforcement," 2023. doi: 10.2813/255453

[2] H. Williams and C. McCulloch, "Truth decay and national security: Intersections, insights, and questions for future research," Santa Monica, CA, USA, 2023. [Online]. Available: https://www.rand.org/pubs/perspectives/PEA112-2.html

[3] K. T. Gradoń, "Generative artificial intelligence and medical disinformation," *British Medical Journal*, no. 384, 2024. doi: 10.1136/bmj.q579

[4] P. Gryka, K. Gradoń, M. Kozłowski, M. Kutyła, and A. Janicki, "Detection of AI-generated emails – a case study," in *Proc. 13th International Workshop on Cyber Crime (IWCC 2024)*, Vienna, Austria, 2024, (accepted for publication).

[5] A. Knott, D. Pedreschi, R. Chatila, T. Chakraborti, S. Leavy, R. Baeza-Yates, D. Eyers, A. Trotman, P. D. Teal, P. Biecek, S. Russell, and Y. Bengio, "Generative AI models should include detection mechanisms as a condition for public release," *Ethics and Information Technology*, vol. 25, no. 4, p. 55, 12 2023. doi: 10.1007/s10676-023-09728-4

[6] K. Krishna, Y. Song, M. Karpinska, J. Wieting, and M. Iyyer, "Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense," *Advances in Neural Information Processing Systems*, vol. 36, 3 2024.

[1]https://github.com/mksochota16/anti-gpt-checker/

[7] F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker, "Perplexity—a measure of the difficulty of speech recognition tasks," *The Journal of the Acoustical Society of America*, vol. 62, no. S1, pp. S63–S63, 1977. doi: 10.1121/1.2016299

[8] M. Chakraborty, S. T. I. Tonmoy, S. M. M. Zaman, S. Gautam, T. Kumar, K. Sharma, N. Barman, C. Gupta, V. Jain, A. Chadha, A. Sheth, and A. Das, "Counter Turing test (CT2): AI-generated text detection is not as easy as you may think - introducing AI detectability index (ADI)," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023. doi: 10.18653/v1/2023.emnlp-main.136 pp. 2206–2239. [Online]. Available: https://aclanthology.org/2023.emnlp-main.136

[9] I. Cingillioglu, "Detecting AI-generated essays: the ChatGPT challenge," *International Journal of Information and Learning Technology*, vol. 40, pp. 259–268, 5 2023. doi: 10.1108/IJILT-03-2023-0043

[10] L. Fröhling and A. Zubiaga, "Feature-based detection of automated language models: tackling GPT-2, GPT-3 and Grover," *PeerJ Computer Science*, vol. 7, p. e443, 4 2021. doi: 10.7717/peerj-cs.443

[11] Y. Shi, Q. Sheng, J. Cao, H. Mi, B. Hu, and D. Wang, "Ten words only still help: Improving black-box AI-generated text detection via proxy-guided efficient re-sampling," *arXiv preprint*, vol. arXiv:2402.09199, 2024. [Online]. Available: http://arxiv.org/abs/2402.09199

[12] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn, "DetectGPT: Zero-shot machine-generated text detection using probability curvature," in *Proc. International Conference on Machine Learning*. Online: PMLR, 2023, pp. 24 950–24 962.

[13] F. Harrag, M. Dabbah, K. Darwish, and A. Abdelali, "Bert transformer model for detecting Arabic GPT2 auto-generated tweets," in *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, I. Zitouni, M. Abdul-Mageed, H. Bouamor, F. Bougares, M. El-Haj, N. Tomeh, and W. Zaghouani, Eds. Barcelona, Spain (Online): Association for Computational Linguistics, Dec. 2020, pp. 207–214. [Online]. Available: https://aclanthology.org/2020.wanlp-1.19

[14] J. D. Rodriguez, T. Hay, D. Gros, Z. Shamsi, and R. Srinivasan, "Cross-domain detection of GPT-2-generated technical text," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 2022.naacl-main.88. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.naacl-main.88 pp. 1213–1233.

[15] S. Mukherjee, "Exploring burstiness: Evaluating language dynamics in LLM-generated texts," 2023, [Online]. Available: https://ramblersm.medium.com/exploring-burstiness-evaluating-language-dynamics-in-llm-generated-texts-8439204c75c1 (Accessed on Apr 30, 2024).

[16] I. Okulska, D. Stetsenko, A. Kołos, A. Karlińska, K. Głąbińska, and A. Nowakowski, "Stylometrix: An open-source multilingual tool for representing stylometric vectors," *arXiv preprint arXiv:2309.12810*, vol. 2309.12810, 9 2023.

[17] J. Morris, "LanguageTool Python library," 2024, https://pypi.org/project/language-tool-python/ (Accessed on May 10, 2024). [Online]. Available: https://pypi.org/project/language-tool-python/

[18] 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, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[19] _w1998, "Spam email dataset," 2023, (Accessed on Jan 14, 2024). [Online]. Available: https://www.kaggle.com/datasets/jackksoncsie/spam-email-dataset/data

[20] R. Modi, "Email classification dataset," 2023, (Accessed on Jan 14, 2024). [Online]. Available: https://github.com/rmodi6/Email-Classification/tree/master

[21] Apache Public Datasets, "The Spam Assassin Email Classification Dataset," 2023, (Accessed on Jan 14, 2024). [Online]. Available: https://www.kaggle.com/datasets/ganiyuolalekan/spam-assassin-email-classification-dataset/data