ISSN 2007-9737

PoSLemma: How Traditional Machine Learning and Linguistics Preprocessing Aid in Machine Generated Text Detection

Diana Jiménez, Marco A. Cardoso-Moreno, Fernando Aguilar-Canto,
Omar Juárez-Gambino, Hiram Calvo

Centro de Investigación en Computación,
Instituto Politécnico Nacional,
Mexico

{djimenezl2022, mcardosom2021, faguilarc2021, hcalvo}@cic.ipn.mx, jjuarezg@ipn.mx

Abstract. With the release of several Large Language Models (LLMs) to the public, concerns have emerged regarding their ethical implications and potential misuse. This paper proposes an approach to address the need for technologies that can distinguish between text sequences generated by humans and those produced by LLMs. The proposed method leverages traditional Natural Language Processing (NLP) feature extraction techniques focusing on linguistic properties, and traditional Machine Learning (ML) methods like Logistic Regression and Support Vector Machines (SVMs). We also compare this approach with an ensemble of Long-Short Term Memory (LSTM) networks, each analyzing different paradigms of Part of Speech (PoS) taggings. Our traditional ML models achieved F1 scores of 0.80 and 0.72 in the respective analyzed tasks.

Keywords. Generative text detection, text generation, AuTexTification, logistic regression, support vector machine (SVM), classification.

1 Introduction

There has been an increase—in the last few years—in the number of LLMs available to the public, among such models we find: Pathways Language Model (PaLM) [4], BLOOM [28], Bidirectional Encoder Representations from Transformers (BERT) [6], BART [14], Robustly Optimized BERT Pretraining Approach (RoBERTa) [15], Generative Pre-trained Transformer (GPT) [23], GPT-2 [24], GPT-3 [3], and, more recently, Chat-GPT and GPT-4 [22].

Although great performance has been achieved in terms of text generation, there are some ethical issues that need to be addressed, for instance: the lack of validation for the data retrieved from these models, since they suffer from hallucinations—the information provided might be incorrect— [13]; the creation of fake news [34, 3] and academic cheating [1].

Since these models are publicly available, there exists high risk of misuse of LLMs models; therefore, there is a requirement for the creation of algorithms and systems capable of differentiating human generated text from that generated by LLMs. To this purpose, both, LLMs, as well as traditional Machine Learning (ML) have been tested on this classification task. When it comes to LLMs, several models have been used to leverage their capabilities (including BERT, RoBERTa, and GPT-2), while traditional ML have been less used (see Related work).

In this paper, we present our approach for the human vs. machine generated text classification problem, to this purpose, we selected the AuTexTification dataset [26] to work with. This dataset is divided, mainly, into two subtasks: 1) human vs. machine generated text and 2) model profiling, i.e., for a given machine generated sequence, determine which model created it.

Our proposal includes: a preprocessing stage, common in traditional NLP tasks, conformed by Tokenization, Multi-Word Token Expansion, Part of Speech and Lemmatization.

This stage is followed by Logistic Regression and Support Vector Machines (SVMs) models with 10-fold Cross Validation. Due to the particularities of both subtasks, special attention was taken into the feature extraction stages (previous to providing inputs to the models), creating combinations of unigrams, bigrams and trigrams of different nature, from PoS Taggins to Lemmas.

The rest of the paper is structured as follows: Section 2 provides a literature review on the present task, highlighting the most important research. Section 3 introduces the dataset, preprocessing and models employed for our study.

In Section 4 we present our empirical results from the experiments, as well as an overview of the meaning of such values. Lastly, Section 5 offers our conclusions drawn based on our results and observations.

2 Related Work

The task of detecting machine-generated text is closely related to and a direct consequence of the task of text generation, which was first implemented by authors such as Mann and McKeown [17, 18].

However, only recently has the problem of detecting whether a given text was generated or identifying the model that produced the text become challenging, even for humans [5]. Most efforts to detect or classify generated content apply Large Language Models such as BERT [10, 32], DistilBERT [21], RoBERTa [30, 7, 31, 12, 25, 9, 16, 19, 35], and others [2].

Classical approaches to detect generated text have also been attempted. Solaiman et al. [30] utilized classical machine learning classifiers such as logistic regression to identify content generated by GPT-2, finding that they do not perform worse than language models.

Unfortunately, this approach has not been completely followed by subsequent literature. In addition, statistical criteria have also been considered to address the task of detecting generated text [8, 20], as well as feature extraction [33, 29].

In the AuTexTification task [27], a preliminary evaluation of the task of detecting generated

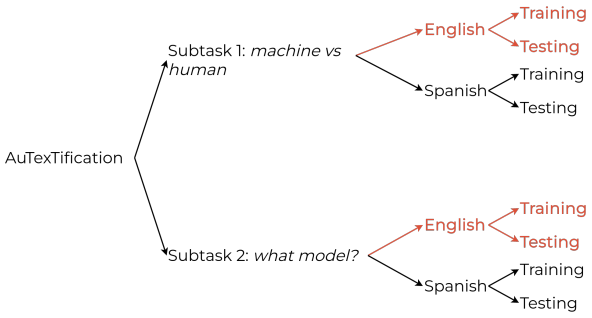

Fig. 1. The AuTexTification subtasks and their corresponding datasets for each language

English text, logistic regression achieved a score of 0.6578 in F1-macro, surpassing the language model DeBERTa V3 (0.571), whereas in the model attribution task logistic regression reached a score of 0.3998, while DeBERTa V3 reached 0.6042.

3 Methodology

3.1 Dataset Description

The AuTexTification dataset [26] consists on data for two different subtasks: human vs machine generated text detection and model attribution. Therefore, this dataset contains text produced either by humans or by any of the following models: BLOOM-1b7, BLOOM-3b, BLOOM-7b1, GPT-3 Babbage, GPT-3 Curie and GPT-3 DaVinci-003.

Consequently, the training dataset for the first subtask consists on 33,845 text samples labeled as either machine-generated or human; on the other hand, for the second subtask, its training set includes 22,416 machine-generated text samples, each labeled with one of the letters A to F, corresponding to the modelo which generated the text.

The maximum text length is 98 and 97 words for subtasks one and two, respectively. Subtask 1 has a fixed testing set of 21,832 samples, whereas subtask 2 has 5,605 samples in its testing set. AuTexTification provides separate datasets for English and Spanish Languages, but our study only focuses on English. Figure 1 shows the structure of the dataset; it highlights in red the data used in this paper.

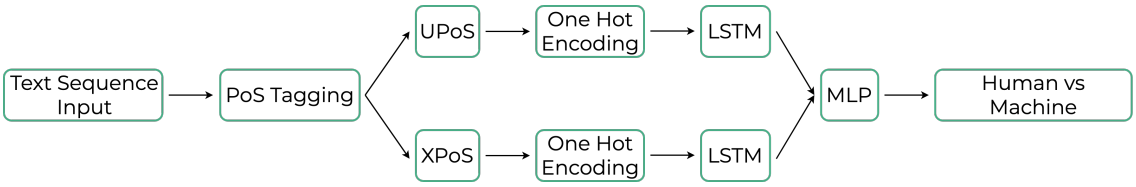

Fig. 2. LSTM ensemble proposal pipeline

3.2 Preprocessing and Feature Extraction

3.2.1 Machine Learning Algorithms

We utilized a preprocessing pipeline for both tasks with the aim of extracting more significant syntactic and lexical features. This preprocessing sequence encompassed the following stages:

1. Tokenization: The raw text was broken down into individual tokens (words in this case).
2. Multi-Word Token (MWT) Expansion: This involved identifying combinations of words that function as a single unit and treating them as cohesive entities.
3. Part of Speech Tagging (PoS): To understand the grammatical roles and relationships within the text, we applied Part of Speech tagging. This involved assigning a specific tag, such as noun, verb, adjective, etc., to each token. Part of Speech tagging assists in capturing syntactic structures and provides valuable context for subsequent steps.
4. Lemmatization: Is a text normalization technique that involves reducing words to their base or root form.

The preprocessing steps applied to the raw text contribute significantly to its refinement, making it more suitable for analysis and improving classification accuracy. For our experiments, we included the sequence of lemmas and Part of Speech (PoS) tags as features.

The incorporation of PoS sequences was particularly important for both tasks, as it provided valuable insights into the writing style and enabled us to identify patterns and combinations of PoS tags that contributed to the text's structure. Overall, utilizing PoS sequences enhanced our ability to extract syntactic information.

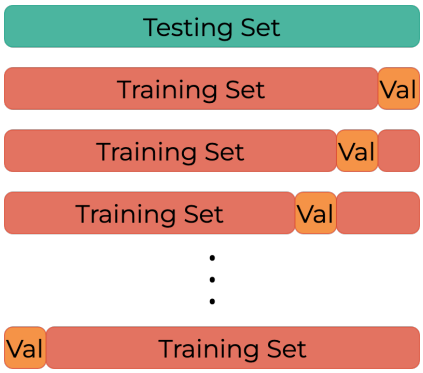

Fig. 3. 10-fold cross validation procedure

3.2.2 LSTM Ensemble

For this approach, we decided to work with two different PoS tags configurations, i.e., we used UPoS and XPoS, since the first provides a PoS analysis universal to most natural languages and the latter provides a specific PoS analysis for the given language which, in this particular case, is English.

The proposal was tested only on subtask 1 of the dataset (See Section 3.3), and involves the training of two LSTM cells, each one of them analysing—simultaneously— either UPoS or XPoS, where each tag was converted to a one-hot representation; once both LSTMs give their outputs, these are concatenated to become the input of a Multi-Layer Perceptron (MLP) of the following characteristics:

— A linear layer with a number of neurons equal to the sum of the hidden states of both LSTMs, for a total of 64 units.

— A dropout layer—for regularization purposes—, with a probability of 0.5 for every neuron to be zeroed out.

Table 1. Results of experiments of the subtask 1

Model	Feat.	Recall	Precision	Accuracy	F1 score
LR	a	0.7636	0.763	0.7633	0.0763
	b	0.774	0.7735	0.7732	0.7731
	c	0.6215	0.5793	0.5811	0.53332
	d	0.7831	0.7807	0.7801	0.7797
	e	0.7405	0.7405	0.7405	0.7404
	f	0.761	0.7546	0.7538	0.7538
	g	0.8057	0.8055	0.8055	0.8054
SVM	a	0.754	0.7552	0.7516	0.7512
	b	0.7664	0.7616	0.7625	0.7587
	c	0.6425	0.6076	0.6063	0.573
	d	0.774	0.7616	0.7625	0.7587
	e	0.7301	0.7301	0.73	0.7299
	f	0.7558	0.7516	0.7518	0.7506
	g	0.8016	0.8014	0.8014	0.8013

— A ReLU activation funciton layer, and,

— A single neuron as output, with a Sigmoid activation function for classification purposes.

Figure 2 shows the pipeline used for this ensemble proposal.

3.3 Subtask 1: Automatically Generated Text Identification

We deemed feature extraction crucial for this task, prompting us to experiment with various combinations of features:

a) Unigrams of lemmas with frequency counts.

b) Unigrams of lemmas with binary counts.

c) Unigrams of PoS with frecuency counts.

d) Unigrams of lemmas with binary counts along with unigrams of PoS and their frequency counts.

e) Unigrams, bigrams and trigrams of lemmas with binary counts.

f) Unigrams, bigrams and trigrams of PoS with frecuency counts.

g) Unigrams, bigrams and trigrams of lemmas with binary counts along with unigrams, bigrams and trigrams with their frecuency counts.

We employed Logistic Regression and Support Vector Machines (SVMs) for the classification, utilizing Stochastic Gradient Descent for both algorithms. In the case of the Support Vector Machine, a linear kernel was applied.

3.4 Task 2: Model Attribution

For this task, we experimented with the following features:

a) Unigrams of lemmas with binary counts.

b) Unigrams of PoS with frecuency counts.

c) Unigrams, bigrams and trigrams of lemmas with binary counts.

d) Unigrams, bigrams and trigrams of PoS with frecuency counts.

e) Unigrams, bigrams and trigrams of lemmas with binary counts along with unigrams, bigrams and trigrams with their frecuency counts.

Continuing with the previous proposal, we employed Logistic Regression and Support Vector Machines using Stochastic Gradient Descent.

4 Results

4.1 Validation Methods and Evaluation

As stated in Section 3.1, the AuTexTification tasks data is available for two subtasks, each one of them with four datasets: a training and a testing datasets, for English and Spanish Languages (see Figure 1).

For the purpose of this study, we exclusively focused on utilizing the training and testing data associated with subtasks one and two, for the English language.

We employed the training set as a development set. This involved partitioning the training data further to create a validation subset.

For the conventional machine learning techniques, we adopted a cross-validation approach as our chosen validation method.

Table 2. Results in the testing set with the selected features (subtask 1)

	Recall	Precision	Accuracy	F1 score
LR	0.7296	0.6942	0.699	0.6852
SVM	0.7319	0.7258	0.7277	0.7252
Baseline				0.6578
SotA				0.8091

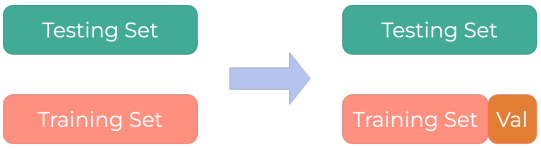

Fig. 4. Hold out validation procedure

The validation results presented in this study represent the average outcomes obtained from applying a 10-fold cross-validation process. On the other hand for our LSTM ensemble proposal, we used a hold out validation procedure on the training set, where 80% was still considered as training data, while the resting 20% became the validation set.

Figures 3 and 4 show a graphical representation of both 10-fold cross validation and hold out validation, respectively.

Subsequently, we proceeded to evaluate our models on the designated test set. The outcomes generated from this testing phase serve as the basis for our comparative analysis against both the state-of-the-art methods and the established baselines set forth in the contest.

4.2 Classification results. Subtask 1

4.2.1 LSTM ensemble

For the LSTM ensemble, the selected loss function was Binary Cross Entropy, with a learning rate of 3×10^{-4}; the number of epochs proposed was 80; and the optimizer used was Adam [11]; additionally, a hold out validation procedure was performed on the provided training set, as described in Section 4.1.

To test the model's performance, the corresponding testing set was used. This model only achieved an F1 average macro value of **0.62**.

4.2.2 Machine Learning Algorithms

Initially, we tested the various features suggested in the preceding section using the development set. The results are presented in Table 1.

The Table 1 highlights that the optimal feature combination for both classification algorithms is the combination "g," encompassing unigrams, bigrams, and trigrams of lemmas, as well as unigrams, bigrams, and trigrams for the PoS sequence.

As a result, we have chosen to employ these features for the test set.The results are presented in Table 2.

4.3 Classification results. Subtask 2

For subtask 2, we adhered to a similar approach, commencing with experimentation on the development set. The results are presented in Table 3.

Table 3 demonstrates that the most effective feature blend for both classification algorithms is the "e" combination.

This amalgamation comprises unigrams, bigrams, and trigrams of lemmas, along with unigrams, bigrams, and trigrams for the PoS sequence. Subsequently, we opted to utilize these features for the test set. The outcomes are outlined in Table 4.

5 Conclusions

We introduce a proposal addressing two distinct tasks that share similarities with the Natural Language Processing task of "author attribution."

Our approach places a stronger emphasis on feature extraction rather than on the model itself, employing traditional machine learning algorithms. Notably, both tasks surpass the baseline set by the AuTexTification contest organizers, utilizing Logistic Regression.

While our approach may not attain the pinnacle of state-of-the-art performance, it highlights the enduring importance of preprocessing and feature selection.

Tables 1 and 3 show us the significant impact of these factors, as they reveal substantial variations in outcomes even with the exact same model.

Table 3. Results of experiments mentioned in the previous section

Model	Feat.	Recall	Precision	Accuracy	F1 score
LR	a	0.2897	0.2777	0.2793	0.2277
	b	0.3844	0.3916	0.3924	0.3847
	c	0.4274	0.4399	0.4399	0.4302
	d	0.3332	0.3416	0.3415	0.3444
	e	0.459	0.4649	0.4646	0.4594
SVM	a	0.2814	0.2513	0.2537	0.1945
	b	0.3643	0.3701	0.3798	0.3647
	c	0.4182	0.4284	0.4284	0.4176
	d	0.3321	0.3208	0.3219	0.3088
	e	0.4585	0.4514	0.4517	0.4443

Table 4. Results for the test set with the selected features

	Recall	Precision	Accuracy	F score
LR	0.4785	0.4854	0.4869	0.4797
SVM	0.462	0.4844	0.4639	0.4602
Baseline				0.3998
SotA				0.625

In the first task, our results with Support Vector Machine approach the state-of-the-art performance without resorting to large models, indicating that notable progress can be achieved. Furthermore, our findings underscore the pivotal role of syntactic information as a crucial feature when discerning authorship.

It is important to note that, although a Deep Learning (DL) architecture was proposed, we consider that the lower performance achieved by the model is inherent to the lack of a detailed text preprocessing, based strictly on the tasks to solve.

Therefore, special care must to be taken when proposing DL models, since the preprocessing must consider both, the characteristics of the text generation and model profiling tasks, as well as the type of preprocessing that works well with a given DL architecture.

We aspire to build upon our current approach by integrating these features into more sophisticated machine learning algorithms, such as Neural Networks. This progressive step aims to further enhance the quality of our outcomes.

Acknowledgments

The authors gratefully acknowledge the Instituto Politécnico Nacional (Secretaría Académica, Secretaría de Investigación y Posgrado, Centro de Investigación en Computación) and the Consejo Nacional de Humanidades Ciencias y Tecnologías (CONAHCYT) for their economic support to develop this work.

References

1. **Alexander, K., Savvidou, C., Alexander, C. (2023).** Who wrote this essay? Detecting AI-generated writing in second language education in higher education. Teaching English with Technology, Vol. 23, No. 2, pp. 25–43. DOI: 10.56297/BUKA4060/XHLD5365.
2. **Antoun, W., Baly, F., Hajj, H. (2020).** AraBERT: Transformer-based model for arabic language understanding. Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, pp. 9–15.
3. **Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., et al. (2020).** Language models are few-shot learners. 34th Conference on Neural Information Processing Systems, Vol. 33, No. 159, pp. 1877–1901.
4. **Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., et al. (2022).** PaLM: Scaling language modeling with pathways. DOI: 10.48550/arXiv.2204.02311.

5. **Clark, E., August, T., Serrano, S., Haduong, N., Gururangan, S., Smith, N. A. (2021).** All that's 'human' is not gold: Evaluating human evaluation of generated text. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Vol. 1, pp. 7282–7296. DOI: 10.18653/v1/2021.acl-long.565.

6. **Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019).** BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 4171–4186.

7. **Fagni, T., Falchi, F., Gambini, M., Martella, A., Tesconi, M. (2021).** TweepFake: About detecting deepfake tweets. PLoS One, Vol. 16, No. 5. DOI: 10.1371/journal.pone.0251415.

8. **Gehrmann, S., Harvard, S., Strobelt, H., Rush, A. M. (2019).** GLTR: Statistical detection and visualization of generated text. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 111–116. DOI: 10.18653/v1/P19-3019.

9. **Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., Yue, J., Wu, Y. (2023).** How close is ChatGPT to human experts? Comparison corpus, evaluation, and detection. DOI: 10.48550/arXiv.2301.07597.

10. **Ippolito, D., Duckworth, D., Callison-Burch, C., Eck, D. (2019).** Automatic detection of generated text is easiest when humans are fooled. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1808–1822. DOI: 10.18653/v1/2020.acl-main.164.

11. **Kingma, D. P., Ba, J. (2014).** ADAM: A method for stochastic optimization. International Conference on Learning Representations. DOI: 10.48550/arXiv.1412.6980.

12. **Kumarage, T., Garland, J., Bhattacharjee, A., Trapeznikov, K., Ruston, S., Liu, H. (2023).** Stylometric detection of AI-generated text in twitter timelines. DOI: 10.48550/arXiv.2303.03697.

13. **Leiser, F., Eckhardt, S., Knaeble, M., Mädche, A., Schwabe, G., Sunyaev, A. (2023).** From ChatGPT to FactGPT: A participatory design study to mitigate the effects of large language model hallucinations on users. Proceedings of Mensch und Computer 2023, pp. 81–90. DOI: 10.1145/3603555.3603565.

14. **Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L. (2019).** BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703.

15. **Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019).** RoBERTa: A robustly optimized BERT pretraining approach. DOI: 10.48550/arXiv.1907.11692.

16. **Ma, Y., Liu, J., Yi, F., Cheng, Q., Huang, Y., Lu, W., Liu, X. (2023).** AI vs. human–differentiation analysis of scientific content generation. DOI: 10.48550/arXiv.2301.10416.

17. **Mann, W. C., Matthiessen, C. M. (1983).** Nigel: A systemic grammar for text generation. Technical report, University of Southern California, Marina del Rey, Information Sciences Institute.

18. **McKeown, K. R. (1992).** Text generation (Studies in natural language processing). Cambridge University Press.

19. **Mireshghallah, F., Mattern, J., Gao, S., Shokri, R., Berg-Kirkpatrick, T. (2023).** Smaller language models are better black-box machine-generated text detectors. DOI: 10.48550/arXiv.2305.09859.

20. **Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., Finn, C. (2023).** DetectGPT: Zero-shot machine-generated text detection using probability curvature. DOI: 10.48550/arXiv.2301.11305.

21. **Mitrović, S., Andreoletti, D., Ayoub, O. (2023).** ChatGPT or human? Detect and explain. Explaining decisions of machine learning model for detecting short ChatGPT-generated text. DOI: 10.48550/arXiv.2301.13852.

22. **OpenAI (2023).** GPT-4 technical report. DOI: 10.48550/arXiv.2303.08774.

23. **Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. (2018).** Improving language understanding by generative pre-training. OpenAI.

24. **Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2019).** Language models are unsupervised multitask learners. OpenAI, Vol. 1, No. 8, pp. 9.

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

26. **Sarvazyan, A., González, J. Á., Franco, M., Rangel, F. M., Chulvi, M. A., Rosso, P. (2023).** AuTexTification dataset (full data). Zenodo. DOI: 10.5281/zenodo.7956207.

27. **Sarvazyan, A. M., González, J. Á., Franco-Salvador, M., Rangel, F., Chulvi, B., Rosso, P. (2023).** Overview of AuTexTification at IberLEF 2023: Detection and attribution of machine-generated text in multiple domains. Procesamiento del Lenguaje Natural, , No. 71, pp. 275–288.

28. **Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., Tow, J., Rush, A. M., Biderman, S., Webson, A., Ammanamanchi, P. S., Wang, T., Sagot, B., Muennighoff, N., Villanova-Moral, A., Ruwase, O., et al. (2022).** BLOOM: A 176B-parameter open-access multilingual language model. BigScience Workshop. DOI: 10.48550/arXiv.2211.05100.

29. **Shijaku, R., Canhasi, E. (2023).** ChatGPT generated text detection. Unpublished.

30. **Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., Radford, A., Krueger, G., Kim, J. W., Kreps, S., McCain, M., Newhouse, A., Blazakis, J., McGuffie, K., Wang, J. (2019).** Release strategies and the social impacts of language models. OpenIA Report.

31. **Tourille, J., Sow, B., Popescu, A. (2022).** Automatic detection of bot-generated tweets. Proceedings of the 1st International Workshop on Multimedia AI against Disinformation, pp. 44–51. DOI: 10.1145/3512732.3533584.

32. **Uchendu, A., Le, T., Shu, K., Lee, D. (2020).** Authorship attribution for neural text generation. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 8384–8395. DOI: 10.18653/v1/2020.emnlp-main.673.

33. **Zaitsu, W., Jin, M. (2023).** Distinguishing ChatGPT (-3.5,-4)-generated and human-written papers through Japanese stylometric analysis. PLoS One, Vol. 18, No. 8. DOI: 10.1371/journal.pone.0288453.

34. **Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., Choi, Y. (2019).** Defending against neural fake news. Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vol. 32, No. 812, pp. 9054–9065.

35. **Zhan, H., He, X., Xu, Q., Wu, Y., Stenetorp, P. (2023).** G3Detector: General GPT-generated text detector.

Article received on 14/06/2023; accepted on 20/09/2023.
Corresponding author is Hiram Calvo.