ISSN 2007-9737

A Dimensionality Reduction Approach for Text Vectorization in Detecting Human and Machine-generated Texts

Jonathan Rojas-Simón, Yulia Ledeneva[*], René Arnulfo García-Hernández

Autonomous University of the State of Mexico,
Tianguistenco Professional Academic Unit, Toluca,
Mexico

{jrojass, ynledeneva, reagarciah}@uaemex.mx

Abstract. Distinguishing between human and machine-generated texts has been a task of recent interest in Natural Language Processing (NLP), especially in the face of the malicious use of Large-Language Models (LLMs). As the result of this, several state-of-the-art methods and approaches have been proposed, providing promising results. However, some of them are unreliable in explaining how features influence the detection of human and machine-generated texts. In this sense, previous studies have explored the effectiveness of traditional machine learning algorithms using lexical features based on ASCII code characters. Nevertheless, not all these features are used, which may difficult this task. Therefore, in this paper, we propose a dimensionality reduction of these features to improve the performance of this text vectorization using traditional machine learning algorithms. The proposed dimensionality reduction has been tested in the AuTexTification task in English and Spanish documents. According to the results, the dimensionality reduction of features improves the performance of machine-learning algorithms, serving this vectorization as inputs to more advanced machine-learning algorithms.

Keywords. Large-language models (LLMs), machine learning algorithms, ASCII-based text vectorization, dimensionality reduction, AuTexTification.

1 Introduction

The Natural Language Processing (NLP) has been an active subfield of research for Artificial Intelligence (AI) over the last three decades, proposing several applications such as Machine Translation (MT), Automatic Text Summarization (ATS) [12, 13], Sentiment Analysis (SA) [11], Chatbots [9], etc. However, significant advances have been achieved since the creation of Large-Language Models (LLMs). According to [9], the LLMs are learning models that can process, comprehend, and generate natural language text, whose training is done with massive datasets of language. Currently, the most popular LLMs are the Generative Pre-trained Transformer (also known as GPT) [14, 20], Pathways Language Model (PaLM) [5], BLOOM [24], and ChatGPT. Nevertheless, other LLMs have been released year by year, such as Falcon [17], LLaMA [29], and Gemini [8], among others.

Nowadays, LLMs are being extensively used for different purposes, such as searching relevant information from news, customer service, social media content creation, answering questions naturally, and others. In general, LLMs have demonstrated that they can produce high-quality texts regarding grammaticality, fluency, coherence [15], and usage of real-world knowledge [20]. However, LLMs can be misused to spread false information, reviews, and opinions to influence people's interests.

Recent studies have shown that content produced by LLMs is generally more accurate than human-generated information but also presents more persuasive disinformation [28]. As a result, it is becoming increasingly difficult to differentiate between LLM-generated content and human-generated content [32]. Moreover, the government, academia, and industry have proposed ethical regulations regarding using these models [30].

On the other hand, this situation motivates the research community to develop methods that can distinguish human-generated texts from machine-generated texts. Some of them are based on

doi: 10.13053/CyS-28-4-5214

traditional machine learning algorithms [10, 22, 27], well-known vector representations [23, 32], text representation models [27], and other LLMs [2, 7]. In addition, academia and industry have organized large-scale shared tasks for the same purpose, such as RuATD-2022 [25] and DagPap24 [4]. In particular, AuTexTification (**Au**tomatic **Tex**t Iden**Tification**) [23] has been a task of recent interest because several efforts have been made to develop methods to detect human and machine-generated texts in different domains.

In previous works [23], it has been observed that LLMs-based classifiers usually perform better than traditional machine learning algorithms. However, the LLMs are unreliable in explaining how features influence the detection of human and machine-generated texts. In this sense, Rojas et. al. [22] explored the effectiveness of supervised and unsupervised machine learning algorithms, using lexical features based on the characters of the ASCII code.

Although the results obtained from supervised machine learning algorithms perform better than the baseline methods, it is necessary to determine what features are more relevant to the task. Therefore, in this paper, we analyze what features from the ASCII code are helpful to detecting human and machine-generated texts.

From such analysis, we perform a dimensionality reduction of these features to train machine learning classifiers, such as the Multilayer Perceptron (MLP), Naive Bayes (NB), Logistic Regression (LR), and K-Nearest Neighbors (KNN). Thus, we assume that if the most frequent features of ASCII code are used, the classification will be improved.

The rest of the paper is organized as follows: Section 2 describes the background and previous studies. Section 3 briefly describes the supervised machine-learning methods used in this study. Additionally, we describe the lexical features to represent each document and select the most relevant.

Section 4 explains the experiments and obtained results of the proposed dimensionality reduction. In addition, we compare it to baseline and state-of-the-art methods. Finally, Section 5 describes the conclusions of this work and exposes future works.

2 Background and Previous Studies

The need to identify texts created by humans and machines first arose in relation to fake news. Previous studies [28] revealed that humans tend to rate disinformation produced by LLMs as more credible than disinformation written by humans. The first work that followed this idea was done in [32], whose idea was to propose a generative model called GROVER that creates and detects fake news.

This model is based on the transformer architecture (GPT2) and BERT to detect automatic fake news. It places a special [CLS] token at the end of each article and then extracting the final hidden state. This state is fed to a neural layer, which predicts who created each news article using Human and Machine tags.

Afterward, other studies have concentrated on developing baseline models to determine whether a text was written by a human or a machine. For instance, Solaiman et al. [27] applied a baseline model representing each text document through TF-IDF vectors obtained from the Bag-Of-Words (BOW) and Bigrams text representation models. These vectors were introduced into a Logistic Regression algorithm to distinguish WebText articles written by humans from texts generated by GPT-2.

Additionally, other studies have explored the differences between humans and automatic detectors (e.g., GROVER [32]) in identifying LLM-generated texts. As a result, the following observations were stated: (i) humans may better detect semantic errors of LLMs than automatic detectors, and (ii) automatic detectors can identify LLM-generated texts by spotting an excessive use of high-probability words.

Although the works mentioned before contributed to the task, they focused on specific domains, which poses a challenge when text documents come from different domains to news. In this regard, the RuATD-2022 [25], DagPap24 [4], and AuTexTification [23] were proposed as shared tasks to detect human and machine-generated texts from several domains (e.g., historical texts, Wikipedia pages, social media posts, scientific papers, etc.).

In particular, AuTexTification has attracted the research community's attention because it included several LLMs to detect (e.g., GPT, PaLM, BLOOM, and ChatGPT) according to the following subtasks:

1. **Human or Generated:** Develop methods to distinguish between machine or human-generated text written in English and Spanish in different domains.
2. **Model Attribution:** Unlike subtask 1, automatic methods must attribute the authorship of each text to one of six LLMs labeled as A, B, C, D, E, and F. Therefore, the names of the LLMs behind the classes are not provided.

In the context of Subtask 1, several methods have been proposed. For instance, the authors in [7] employed textual, readability, complexity, sentiment, emotion, and toxicity features as inputs to train traditional and deep learning algorithms. Other works employ individual and ensemble classifiers of LLMs based on the GPT2 and GPT3 architectures [1, 2].

Subsequently, Rojas et al. [22] proposed a text vectorization based on the ASCII code. This vectorization considers the frequency of characters to determine the probability that each character may appear from a given document. Formally, this vectorization is shown in Eq. (1):

$$\mathrm{ASCII(d)} = [\mathrm{p(c_1), p(c_2), \ldots, p(c_{255})}], \mathrm{p(c_i)} = \frac{\mathrm{f(c_i)}}{\mathrm{len(d)}}. \quad (1)$$

The function $\mathrm{ASCII(d)}$ receives the input document d. Next, it generates a vector representation of 255 values; each $\mathrm{p(c_i)}$ represents the probability that the character $\mathrm{c_i}$ appears in d. Such probabilities are calculated by dividing the frequency of character $\mathrm{c_i}$ ($\mathrm{f(c_i)}$) and $\mathrm{len(d)}$ (an essential function that counts the number of characters). Furthermore, we considered the value $\mathrm{p(c_{256})}$ for emojis or unknown characters that may appear in the same document.

With this representation in mind, we have sought a lower dimensionality vector representation able to capture only relevant features of a document. Moreover, it is language-independent because we do not require linguistic resources to represent the information provided in the document.

However, not all these features were used, so their selection would be a suitable alternative for improving human and machine-generated text detection. On the other hand, in this paper, we focused on experimenting with the dimensionality reduction of features, where selected features are introduced to supervised machine learning methods.

3 Proposed Dimensionality Reduction of Features

In this section, we briefly describe the dimensionality reduction of features considered for each algorithm. Subsequently, we provide a general description of the machine-learning algorithms used in this work.

3.1 Reduction of Dimensionality

As mentioned in Section 2, the function $\mathrm{ASCII(d)}$ creates a vector of 256 probabilities in which each $\mathrm{c_i}$ appears in d. Nevertheless, in the previous work [22], not all characters obtained a probability value because some are not commonly used in English text documents.

For this reason, it is necessary to determine how many features obtained probability values greater than 0. Therefore, we create a histogram (see Fig. 1), which displays the mean probability values for each feature.

According to the values shown in Fig. 1, we observe that the character number 33 (whitespace) is the most probable to appear for any input document. Moreover, we notice that punctuation marks (character numbers $39 - 47$), numbers (character numbers $48 - 57$), and letters in lowercase (character numbers $98 - 122$) are likely to appear in any document. We have estimated that only 91 features have probability values different from 0.

That is, only 35.54% of features are being used to represent each document. Thus, we considered these 91 features as inputs to machine learning algorithms, reducing 64.46% of the dimension from the original vector representation.

3.2 Machine-learning Algorithms

MLP. The Multilayer Perceptron (MLP) is a deep learning model composed of three types of fully connected layers: (i) the input layer, (ii) one or more hidden layers, and (iii) the output layer. Furthermore, this model can capture complex data relations and solve Pattern Recognition (PR) tasks, such as classification or regression.

LR. The Logistic Regression (LR) measures the relationship between a set of features (X) and a binary output ($\mathrm{y} \in \{0,1\}$). Mathematically, LR is expressed as $1/(1+\mathrm{e}^{-\mathrm{z}})$, where z is the linear combination between the input features ($\mathrm{x_i}$) and model parameters ($\mathrm{w_i}$). The output values range from 0 to 1, indicating the likelihood that the input belongs to the output 1.

NB. The Naive Bayes (NB) algorithm relies on the Bayes' theorem, which deals with the probability calculus. For classification tasks, the NB computes the probability that a pattern may belong to a specific class, considering its input features.

KNN. K-Nearest Neighbors (KNN) is a widely used machine learning algorithm that operates on the majority rule principle. It predicts the label of a test data point by assigning it to the class most common among its K nearest training data points in the feature space.

4 Experimental Results

This section is structured as follows: First, we portray the AuTexTification dataset and evaluation metrics used to measure the performance of the proposed feature dimensionality reduction. Next, we highlight the key experiments conducted with each algorithm discussed in Section 3.2 and their corresponding results. Finally, we compare the performance of these classifiers with state-of-the-art and baseline methods.

4.1 Document Collection

In relation to the subtasks of AuTexTification (see Section 2), we have centered on Subtask 1. Unlike the previous experimental approximation [22], we tested the proposed vector representation in English and Spanish text documents.

Thus, the dataset comprises 55677 English documents and 52191 Spanish documents. Humans and LLMs generated these documents in five domains: tweets (currently known as posts), reviews, how-to articles, news, and legal documents. Table 1 provides an overall description of this dataset.

For the English language, 33845 documents compose the training set, which is relatively balanced per class (Human or Generated).

In the test set, 21832 documents were used, and it is also balanced per class. On the other hand, for the Spanish language, 32062 documents compose the training set, which is also balanced per class.

In the test set, 20129 documents were used, but unlike the English language, this set shows an evident imbalance in classes. For the evaluation stage, the Macro F-measure was used.

4.2 Experimenting with the Parameters of Machine-learning Algorithms

To test the performance of the dimensionality reduction, we performed several experiments in the parameters of algorithms described in Section 3.2 to fine-tune generated models.

Each algorithm received as input the vector representation of features discussed in Section 2, along with the corresponding feature reduction detailed in Section 3.1.

Below, the best parameters of each algorithm are described.

- MLP: The best parameters of the MLP consist of five hidden layers, each containing 300, 250, 500, 100, and 10 neurons. To avoid overfitting, we included Dropout layers in the second (0.2) and third (0.5) hidden layers. The neurons of these layers employ the ReLu function. Finally, in the output layer neurons, we employ the sigmoid function. Moreover, we used the Adam algorithm for the learning process, iterating it in 50 epochs.
- LR: Like previous work [22], we employed the default parameters of the LR provided by the scikit-learn Python library [16].
- NB: We experimented with the same NB variants as in the previous work [22]. Nevertheless, we selected the Bernoulli NB in this experiment because it obtained the best results.

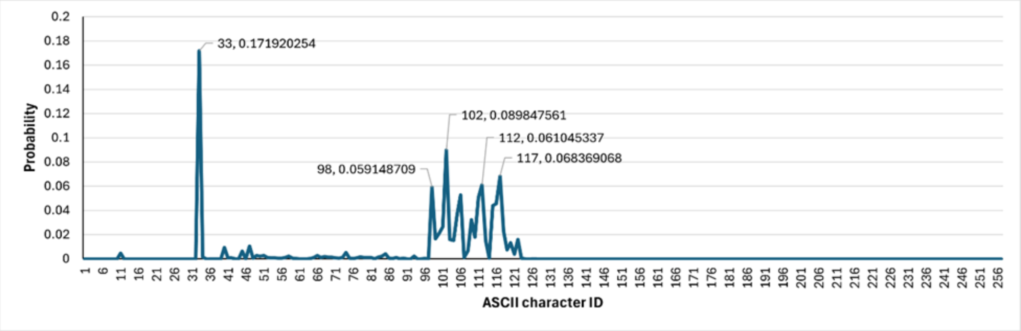


Fig. 1. Distribution of probabilities of features obtained from ASCII code

Table 1. Number of documents per class in English and Spanish (Subtask 1)

	English		**Spanish**	
Class	Training	Test	Training	Test
Human	17046	10642	15787	8920
Generated	16799	11190	16275	11209
Total	33845	21832	32062	20129

– KNN: For this algorithm, we employed the default parameters provided by the scikit-learn Python library [16].

After experimenting with the parameters of each algorithm, we evaluated their effectiveness using confusion matrices. Fig. 2 shows the confusion matrices obtained by the MLP method. Based on the results, the MLP performs better at distinguishing human-written texts in English compared to Spanish, correctly classifying 1260 texts.

This is because the distribution of sets in the Spanish dataset is not balanced; therefore, it is necessary to provide more human-written documents or perform oversampling methods. Despite these observations, we observe that in the Spanish language obtained higher results according to the F-measure (see Table 2).

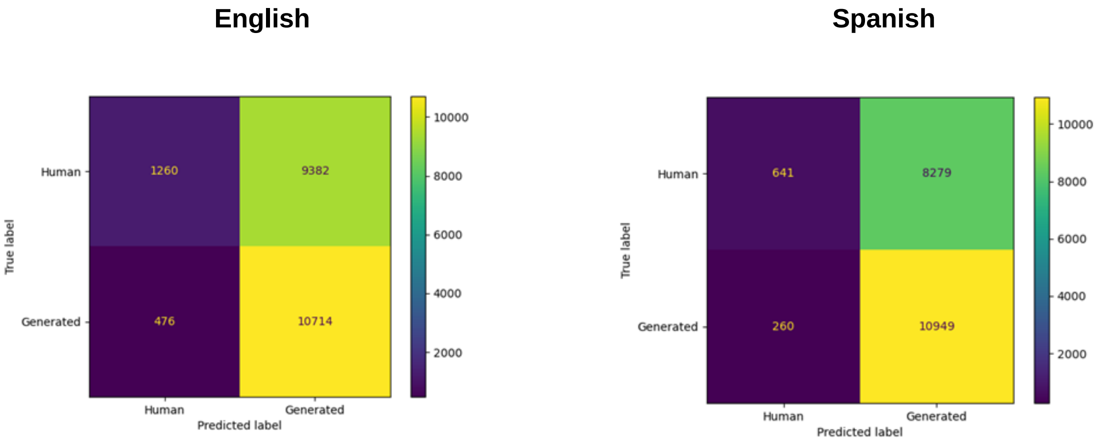
On the other hand, Fig. 4 shows the confusion matrices obtained by the LR method. In general, we notice that LR could better distinguish human-generated texts in English than in Spanish, correctly classifying 3757 human-generated texts. However, the overall performance of both models is not so different regarding F-measure (see Table 2).

According to the confusion matrices displayed in Fig. 3, the NB algorithm better distinguishes human-generated texts in the English language than Spanish texts, classifying 3,431 human-generated texts correctly. In other words, we observe a similar tendency for both languages concerning the previous comparisons. Nevertheless, the classification in the Spanish language obtained higher results according to the F-measure (see Table 2).

Finally, in Fig. 5, we show the confusion matrices obtained by the KNN method. Unlike previous comparisons, we notice that the model generated by the KNN method better distinguishes human-generated texts in Spanish than English texts. Therefore, as expected, the F-measure score is higher in the Spanish set (see Table 3).

4.3 Performance of the Proposed Experimentation with regards to State-of-the-Art/Baseline Methods

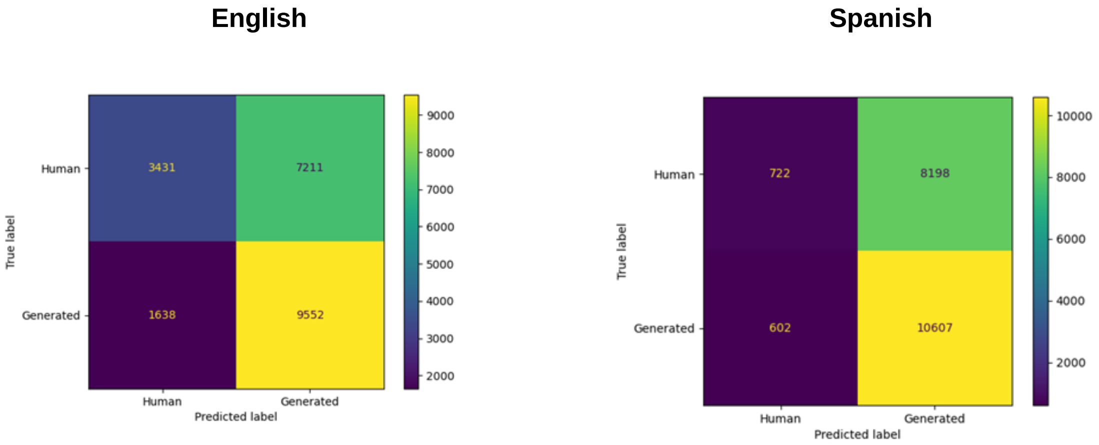
To evaluate the performance of the proposed dimensionality representation, we compare its efficacy concerning state-of-the-art methods and heuristics. Below, we briefly describe the baseline approaches.



English

Spanish

Fig. 2. Confusion matrices obtained by the MLP in English and Spanish languages



English

Spanish

Fig. 3. Confusion matrices obtained by the NB in English and Spanish languages

- BOW+LR: This baseline represents each document in a BOW at character and word levels, creating n-grams of sizes from 1 to 6 [18, 26]. Finally, the resultant representation was input to an LR with default parameters [16].

- LDSE (Low-Dimensionality Semantic Embeddings): This method represents text documents as probability distributions of tokens in the different classes of LDSE [21]. Afterward, the resultant representations are introduced to a Support Vector Machine (SVM) classifier with default parameters [16].

- Random: This baseline considers the class balance for any subtask and language. Therefore, the value of the baseline random is 0.5000.

- SB-FS and SB-ZS: The AuTexTification organizers used the Symanto Brain API to create the Few-shot (SB-FS) and Zero-shot (SB-ZS) models as baselines to classify human and machine-generated texts [23].

- Transformer: It is a method based on the HuggingFace ecosystem to fine-tune pre-trained transformers [31] with default parameters [23].

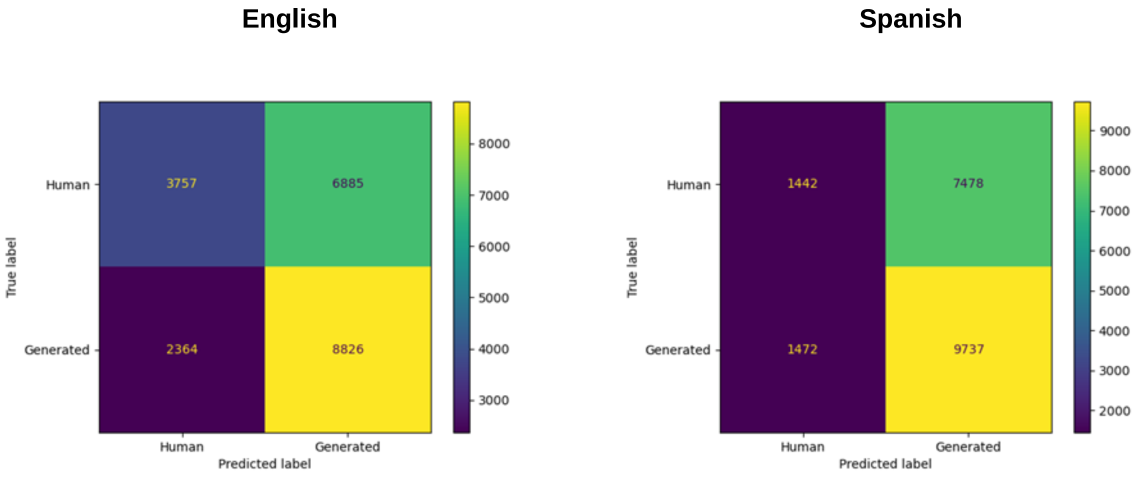


Fig. 4. Confusion matrices obtained by the LR in English and Spanish languages

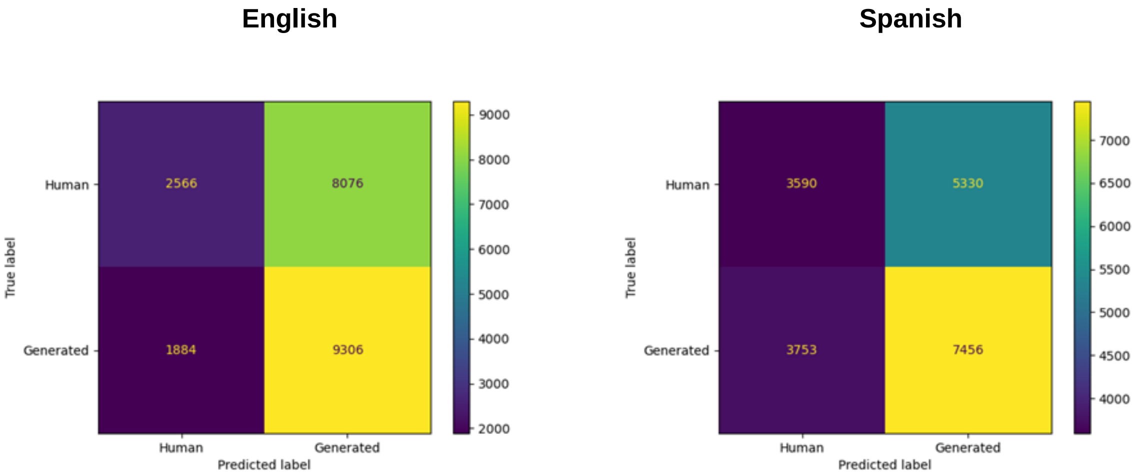


Fig. 5. Confusion matrices obtained by the KNN in English and Spanish languages

In addition to the before-mentioned baselines, the following list describes the state-of-the-art methods that have achieved the best results.

- TALN-UPF: The proposed method of this team was centered on measuring “predictability” (i.e., how likely a given text is according to several LLMs [19]). In addition, they considered linguistic and semantic information from texts to train a neural network.
- CIC-IPN-CsCog and Drocks: The first team implemented a method based on BERT and GPT-2 Small [2] models, while the team Drocks employed a method based on an ensemble of neural models that generate probabilities from various pre-trained LLMs, which were then used as input features for a traditional machine learning classifier.

According to the results shown in Table 3, the MLP (highlighted in bold) performed better than all baseline approaches and some state-of-the-art

Table 2. Performance of machine learning algorithms using the proposed dimensionality reduction

	English			Spanish		
Algorithm	**Recall**	**Precision**	**F-measure**	**Recall**	**Precision**	**F-measure**
MLP	**0.95746**	0.53314	**0.68491**	**0.97680**	0.56943	**0.71945**
LR	0.78874	0.56177	0.65618	0.86868	0.56561	0.68513
NB	0.85362	**0.56983**	0.68343	0.94629	0.56405	0.70680
KNN	0.83164	0.53538	0.65141	0.66518	**0.58314**	0.62146

Table 3. Macro F-measure results between the proposed dimensionality reduction and state-of-the-art methods

Name	English	Spanish
TALN-UPF (Hybrid_plus) [19]	0.8091	0.7077
TALN-UPF (Hybrid) [19]	0.7416	0.6772
CIC-IPN-CsCog (run2) [2]	0.7413	0.5989
Drocks (run2) [1]	0.7330	0.6432
MLP (this work)	**0.6849**	**0.7195**
NB (this work)	**0.6834**	**0.7068**
NB [22]	0.6626	0.6522
BOW+LR (Baseline [23])	0.6578	0.6240
LR (this work)	**0.6562**	**0.6851**
KNN (this work)	**0.6514**	**0.6215**
LR [22]	0.6294	0.5856
MLP [22]	0.6291	0.6128
LDSE (Baseline [23])	0.6035	0.6358
KC [22]	0.5966	0.4987
SB-FS (Baseline [23])	0.5944	0.5605
Transformer (Baseline [23])	0.5710	0.6852
Random (Baseline [23])	0.5000	0.5000
AHC [22]	0.4891	0.4630
SB-ZS (Baseline [23])	0.4347	0.3458

methods for both languages. The main difference between our experimentation and all baselines lies in using lexical features and their corresponding dimensionality reduction (see Section 3.1).

Therefore, we only considered 89 features for each document, whose calculus is based on the most frequent characters of the ASCII code.

Furthermore, the proposed representation has a lower dimensionality than BOW+LR (10,000 features), which in turn shows a better performance.

In comparison to the results obtained from the first experimental approximation (highlighted in italics) [22], the proposed reduction of dimensionality provides a better classification oh human and machine-generated texts, improving in the results of each classifier. For instance, the MLP performs better (0.6849 and 0.7195) than its corresponding first approximation (0.6291 and

0.6128). Therefore, the proposed reduction over the number of features improves this task. Nevertheless, we observed that there is a significant gap concerning state-of-the-art methods. Thus, we need to consider this representation the basis of modern classifiers.

5 Conclusions

In this paper, we have analyzed and evaluated the performance of machine learning algorithms through a dimensionality reduction of lexical features, which are based on character probability distributions. Such a reduction of features is based on the idea that some characters appear more frequently in documents.

Therefore, their representation is more helpful for machine learning algorithms than using all possible features (256). Subsequently, they have been used as inputs to machine learning algorithms, such as MLP, LR, NB, and KNN.

In particular, the MLP performed best for both languages (English and Spanish). In comparison to the baseline and state-of-the-art methods explained in the previous section, the proposed reduction of features provides a better classification between human-written and machine-generated texts.

As mentioned in Section 3, we selected 89 features from 256 previously proposed in [22] to detect human and machine-generated texts, and according to the results shown in Section 4, there are improvements in this dimensionality reduction over baselines and state-of-the-art methods. However, there are areas of improvement to distinguish human-generated texts for both languages better (see Section 4.2).

Therefore, it is necessary to include other features that analyze the texts at syntactic and semantic levels or even use the proposed features to train a neural network (e.g., transformers) to model lexical features.

In future work, we will focus on several aspects of the proposed representation of features. The first is using these features as inputs to state-of-the-art classifiers, such as CNNs, LSTM, or transformers [3]. As the second future work, we seek to incorporate more advanced and explainable text features, such as text complexity [6], readability assessment, vocabulary, and emotion analysis.

Finally, we will analyze the performance of these features for other NLP tasks, such as sentiment analysis and the detection of fake news.

References

1. **Abburi, H., Suesserman, M., Pudota, N., Veeramani, B., Bowen, E., Bhattacharya, S. (2023).** Generative AI text classification using ensemble LLM approaches. Preprint arXiv:230 9.07755.
2. **Aguilar-Canto, F., Cardoso-Moreno, M., Jiménez, D., Calvo, H. (2023).** GPT-2 versus GPT-3 and Bloom: LLMs for LLMs Generative Text Detection.
3. **Arif, M., Ameer, I., Bölücü, N., Sidorov, G., Gelbukh, A., Elangovan, V. (2024).** Mental illness classification on social media texts using deep learning and transfer learning. Computación y Sistemas, Vol. 28, No. 2. pp. 451–464. DOI: 10.13053/cys-28-2-4873.
4. **Chamezopoulos, S., Herrmannova, D., de-Waard, A., Rosati, D., Kashnitsky, Y. (2024).** Overview of the DagPap24 shared task on detecting automatically generated scientific papers. Proceedings of the Fourth Workshop on Scholarly Document Processing Association for Computational Linguistics, Bangkok. pp 7–11.
5. **Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Fiedel, N. (2023).** Palm: Scaling language modeling with pathways. Journal of Machine Learning Research, Vol. 24, No. 240, pp. 1–113.
6. **Ermakova, L., Solovyev, V., Sidorov, G., Gelbukh, A. (2023).** Text complexity and simplification. Frontiers in Artificial Intelligence, Vol. 6, p. 1128446. DOI: 10.3389/frai.2023. 1128446.
7. **Fernández-Hernández, A., Arboledas-Márquez, J. L., Ariza-Merino, J., Zafra, S. M. J. (2023).** Taming the Turing test: exploring machine learning approaches to discriminate Human vs. AI-generated texts. IberLEF@ SEPLN.

8. **Team, G., Anil, R., Borgeaud, S., Alayrac, J. B., Yu, J., Soricut, R., Blanco, L. (2023).** Gemini: A family of highly capable multimodal models. Preprint arXiv:2312.11805.
9. **Hadi, M. U., Al-Tashi, Q., Qureshi, R., Shah, A., Muneer, A., Irfan, M., Zafar, A., Shaikh, M. B., Akhtar, N., Wu, J., Mirjalili, S. (2023).** A survey on large language models: Applications, challenges, limitations, and practical usage. DOI: 10.36227/techrxiv. 23589741.v1.
10. **Ippolito, D., Duckworth, D., Callison-Burch, C., Eck, D. (2020).** 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.
11. **Liu, B. (2015).** Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press, Cambridge.
12. **Neri-Mendoza, V., Ledeneva, Y., García-Hernandez, R. A., Hernández-Castañeda, A. (2023).** Generic and update multi-document text summarization based on genetic algorithm. Computación y Sistemas, Vol. 27, No. 1, pp. 269–279. DOI: 10.13053/cys-27-1- 4538.
13. **Neri-Mendoza, V., Ledeneva, Y., García-Hernández, R. A., Hernández-Castañeda, Á. (2024).** Relevance of sentence features for multi-document text summarization using human-written reference summaries. Mexican Conference on Pattern Recognition Cham: Springer Nature Switzerland. Vol. 14755pp. 319–330. DOI: 10.1007/978-3-031-62836- 8_30.
14. **Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., Lowe, R. (2022).** Training language models to follow instructions with human feedback. 36th Conference on Neural Information Processing Systems, pp 1–15.
15. **Parmar, M., Deilamsalehy, H., Dernoncourt, F., Yoon, S., Rossi, R. A., Bui, T. (2024).** Towards enhancing coherence in extractive summarization: Dataset and experiments with LLMs. Preprint arXiv:2407.04855.
16. **Pedregosa, F., Varoquaux, G., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É. (2011).** Scikit-learn: Journal of machine Learning research, Vol. 12, pp. 2825–2830.
17. **Penedo, G., Malartic, Q., Hesslow, D., Cojocaru, R., Cappelli, A., Alobeidli, H., Pannier, B., Almazrouei, E., Launay, J. (2023).** The refined web dataset for falcon LLM: outperforming curated corpora with web data, and web data only. Advances, pp. 79155–79172.
18. **Pizarro, J. (2019).** Using N-grams to detect bots on Twitter. Lugano, Switzerland.
19. **Przybyła, P., Duran-Silva, N., Egea-Gómez, S. (2023).** I've seen things your machines wouldn't believe: Measuring content predictability to identify automatically-generated text. IberLEF@ SEPLN.
20. **Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2019).** Language models are unsupervised multitask learners. OpenAI blog Vol 1, No. 1, pp. 1–24.
21. **Rangel, F., Franco-Salvador, M., Rosso, P. (2018).** A low dimensionality representation for language variety identification. In: Gelbukh, A. (eds) Computational Linguistics and Intelligent Text Processing, CICLing 2016, Lecture Notes in Computer Science, Springer, Cham. Vol 9624. DOI: 10.1007/978-3-319-75487-1_13.
22. **Rojas-Simón, J., Ledeneva, Y., García-Hernández, R. A. (2024).** Classification of human and machine-generated texts using lexical features and supervised/unsupervised machine learning algorithms. In: Mezura-Montes, E., Acosta-Mesa, H. G., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., Olvera-López, J. A. (eds) Pattern Recognition, MCPR 2024, Lecture Notes in Computer Science, Springer, Cham. Vol 14755. DOI: 10.1007/97 8-3-031-62836-8_31.
23. **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. Preprint arXiv:2309.11285.

24. **Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S. (2022).** BLOOM: A 176B-parameter open-access multilingual language model.

25. **Shamardina, T., Mikhailov, V., Chernianskii, D., Fenogenova, A., Saidov, M., Valeeva, A., Artemova, E. (2022**). Findings of the the RuATD shared task 2022 on artificial text detection in Russian. Preprint arXiv: 220 6.01583.

26. **Sidorov, G., Gelbukh, A., Gómez-Adorno, H., Pinto, D. (2014).** Soft similarity and soft cosine measure: Similarity of features in vector space model. Computación y Sistemas, Vol. 18, No. 3, pp. 491–504. DOI: 10.13053/cys-18-3-2043.

27. **Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., Radford, A., Krueger, G., Wook-Kim, J., Kreps, S., McCain, M., Newhouse, A., Blazakis, J., McGuffie, K., Wang, J. (2019** Release strategies and the social impacts of language models. DOI: 10.48550/arXiv.1908.09203.

28. **Spitale, G., Biller-Andorno, N., Germani, F. (2023).** AI model GPT-3 (dis) informs us better than humans. Science Advances, Vol. 9, No. 26, DOI: 10.1126/sciadv.adh1850.

29. **Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G. (2023).** LLaMA: open and efficient foundation language models. DOI: 10.48550/arXiv. 2302.13971

30. **Widder, D. G., Nafus, D., Dabbish, L., Herbsleb, J. (2022).** Limits and possibilities for ethical ai in open source: A study of deepfakes. 2022 ACM Conference on Fairness, Accountability, and Transparency. pp 2035–2046. DOI: 10.1145/3531146.35337.

31. **Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von-Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le-Scao, T., Gugger, S., Drame, et al. (2020).** Transformers: state-of-the-art natural language processing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Association for Computational Linguistics, Stroudsburg, pp. 38–45. DOI: 10.18653/v1/20 20.emnlp-demos.6.

32. **Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., Choi, Y. (2019).** Defending against neural fake news. NIPS’19: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vol. 32, pp. 9054–9065.

Article received on 20/08/2024; accepted on 21/10/2024.
**Corresponding author is Yulia Ledeneva.*