

Exploring Text Representations for Detecting Automatically Generated Text

Zayra Villegas-Trejo^{1,*}, Helena Gómez-Adorno² and Sergio-Luis Ojeda-Trueba³

¹*Facultad de Ciencias, Universidad Nacional Autónoma de México, Mexico City, Mexico*

²*Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Mexico City, Mexico*

³*Instituto de Ingeniería, Universidad Nacional Autónoma de México, Mexico City, Mexico*

Abstract

In today's rapidly advancing world of technology, artificial intelligence (AI) models have emerged that can generate text automatically. It has become increasingly challenging to discern the difference between machine-generated text and human-written text simply by reading it. This capability of AI poses a problem when it comes to creating fake content or malicious use of these models. This article presents our approach to the AuTextification task at IberLEF 2023, focusing on two subtasks. The first subtask involves binary classification, distinguishing between text written by humans and text generated by AI. The second subtask is a multi-class problem involving six text generation models (A, B, C, D, E, and F). Both subtasks are conducted in English and Spanish languages. Our objective is to accurately determine whether a given text is authored by a human or generated by AI and also to detect the text generation model used. We extract features such as Bag-of-Words (BoW), N-gram structure, and others. Experimental evaluation is performed using Logistic Regression, Random Forest, and Support Vector Machine algorithms. Our results demonstrate that incorporating additional features improves the accuracy of text identification.

1. Introduction


In the current era of artificial intelligence and machine learning, more specifically in the area of Natural Language Processing (NLP), the generation of text with artificial intelligence has been trending lately and has shown significant advances. Users utilize cutting-edge applications like ChatGPT [1] to request information, reviews, presentations, speeches, or opinions. The development of AI language models has enabled the generation of text that can be difficult to distinguish from that written by humans. AI's ability to mimic a human author's writing style presents a significant challenge for systems. In this paper, we explore the challenges and implications associated with identifying AI- and human-generated text through the AuTextification task [2] on the detection of text generated automatically by text generation models at the Iberlef 2023 workshop [3].

People can differentiate when a text was written by an artificial intelligence or a human. This approach was implemented by Dugan, Ippolito, Kirubarajan and, Callison-Burch, who developed the system RoFT(Real or Fake Text) [4] as a game. In this application, people could punt their identification skills thanks to a series of exercises. Also, they gradually improved their

IberLEF 2023, September 2023, Jaén, Spain



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

skills during the experiments [4]. Other researchers used the same methods and obtained similar conclusion [5, 6]. Nevertheless, many people used their linguistic knowledge, language ideas, and personal preferences during the experiments to determine the texts [5]. In consequence, their assessment is subjective. These kinds of experiments resulted in inefficient results [5, 7], as Clark et al. [6] mentions "some evaluators focused on surface-level text qualities to make their decisions and underestimated current NLG models capabilities". Therefore, the evaluations' reliability depends on the evaluators' qualification, making high-quality annotations [4], and the personal perception and knowledge of the participants.

Another way to detect automatically generated text is with automatic systems. They are obtained by the latest generation models. Some companies, such as Google and OpenAI are developing these tools.

Recently OpenAI created a system to detect text written by ChatGPT and other IAs [8]. It consists of a classification tool used model that was altered based on multiple source samples. On the other hand, there is no certainty that Google can detect generated text, as John Mueller replied in a much-cited interview in April 2022 "I can't claim that", when he is asked whether Google can differentiate between a human text or AI algorithm text. Although there are ways to identify where the generated text comes from, this is not the most accurate and reliable method. Even they are more improvements in models and bots are advancing quickly. Under this limitation, Feizi and researchers from the University of Maryland used AI-based paraphrasing tools to rephrase AI-generated text and fed it into various detectors. They got the accuracy of most detectors dropped to nearly 50% [7]. In addition, they use a test called "impossibility result" to show that models of AI become more human-like in the distribution of words in the generated text, and detectors will have a hard time handling them [7].

Our objective is to work on traditional machine learning models to detect text written by humans and other texts generated by different machines, studying the word frequency and some patterns in punctuation and length of documents.

2. Corpus

The Autextification [2] proposed a shared task to identify texts written by AI. This task is carried on along with a series of NLP-related tasks at the IberleF 2023 workshop [3], where the objective is to promote research in the detection of automatically generated text-by-text generation models. The Autextification task comprises two subtasks, subtask_1 (binary classification) and subtask_2 (multiple classification). The corpus presented in Subtask_1 has 33845 instances in English and 32062 in Spanish. Meanwhile, there are 22416 texts and 21935, respectively, for the corpus of subtask_2.

In both corpora, the information is presented in three columns: "id", "text" and "label". The classes available in Subtask_1 are: "generated" and "human" and subtask_2 includes the classes: "A", "B", "C", "D", "E" and "F", which represent five different language models. Tables 1 and 2 present the class distribution of both subtasks. For each subtask we have two subsets of the corpus, train data, and test data. Train data contains the information structured as it was mentioned above and is used to train the models. Test data contains only the "id" and "text" and we used to make the predictions. You can also have visualization data in the section of

Appendix 6.

Table 1

Class distribution of the Subtask_1 train corpus

Lenguaje	English	Spanish
generated	16779	16275
human	17046	15787

Table 2

Class distribution of the Subtask_2 train corpus

Lenguaje	English	Spanish
A	3562	3422
B	3648	3514
C	3687	3575
D	3870	3788
E	3822	3770
F	3827	3866

3. Methodology

We approached both subtasks as a supervised learning problem, the subtask_1 as a binary classification problem, and subtask_2 as a multiclass classification problem. First, we performed some basic pre-processing to the texts, then performed feature extraction to obtain different representations of the texts, and finally, we experimented with various machine learning algorithms. Also, we evaluated the models with the F1-score.

3.1. Pre-processing

First, we analyzed the data to keep relevant information. We consider relevant characteristics like stopwords, symbols, digits, capital letters, the number of punctuation marks, and other linguistic considerations to identify human text and generated texts. The style of the text is an important factor in achieving our goal.

Although we had the hypothesis that not pre-processing the data would yield better results, we did both procedures to compare them. The experiments were performed on two data-set: pre-processed data and raw data. The pre-processed data was cleaned, considering only the next:

- Remove all special character
- Lowercase all the words
- Tokenize
- Remove stopwords

3.2. Train and Test Split

We employed the Stratified K-Fold as implemented in the Scikit-learn [9] library to make splits of five equal groups (folds) while maintaining the proportion of samples from each class in each fold, reflecting the distribution of classes in the original dataset. This ensures the training and testing subsets contain representative samples from all classes.

3.3. Feature Extraction

We applied different techniques to find patterns in the text. We used a Bag-of-Words (BoW) to divide the text per word and organize it by repetition of the frequency. We experimented with techniques such as character N-Grams, for example, Tri-Grams, (2,3)-Grams, and (2,5)-Grams. Finally, we complemented the best N-Gram feature set with additional stilometric features to train the model. The stilometric features are:

- number of digits (d)
- number of others (W)
- number of spaces (s)
- number of stop-words
- length of characters
- number of comas (,)

3.4. Machine Learning Algorithms

In our experiments, we evaluated widely used supervised machine learning algorithms:

- Logistic Regression (LG)
- Random Forest (RF)
- Support Vector Machine linear (SVC)
- Gradient Boost (GB)
- XGBoost (XGB)

In both cases, we evaluated these models with the F1-score to compare them with the respective methods or techniques; then, we only kept the best results to continue experimenting.

4. Experiments

As mentioned before, we employed techniques such as Bag-of-Words (BoW) and N-grams to conduct our experiments. To enhance our model, we incorporated additional features, specifically six stylometric features. In Table 3, the average of the occurrence of each feature in the class “generated” and the class “human”, and the difference of each feature occurrence between the generated and human texts are presented. A smaller difference indicates a lesser contribution of that particular feature to our model. We only present the results for subtask_1_en, as similar findings were observed in the other tasks.

Table 3
Average of ExtraFeatures

Features	AVGgenerated	AVGhuman	Difference
num_dig	5.71	3.53	2.17
num_oth	58.74	62.00	3.25
num_spa	51.24	54.09	2.85
num_com	1.96	2.02	0.05
num_stop	8.58	7.60	0.98
num_len	297.12	313.49	16.3

The experiments were conducted in two phases. In the first phase, we trained the models and prepared the data for testing. The second phase involved performing the actual tests and obtaining predictions to be submitted to the AuTextification task.

Regarding the results obtained from the pre-processed data, the Support Vector Machine (SVC) model achieved the highest F1 score of 82.1% using the Tri-Grams representation scheme. This result was specifically observed for subtask_1 in English, but a similar trend was observed for subtask_1 in Spanish.

In the case of raw data, the scores of most models showed improvement, except for SVC, which actually decreased to an F1 score of 80.2%. Consequently, we focused our subsequent experiments on the improved models obtained from the second set of cases for each subtask. The scores for each case can be found in Table 4 and Table 5.

Table 4
F1 Scores of Subtask_1_en on pre-processed data

Algorithm	BoW	Tri-Grams	(2,5)-Grams	+ExtraFeatures
LR	0.694	0.713	0.735	0.752
RF	0.733	0.711	0.722	0.736
SVC	0.801	0.821	0.815	0.814
GB	0.684	0.697	0.691	0.701
XGB	0.751	0.760	0.768	0.784

Table 5
F1 Scores of Subtask_1_en on raw data

Algorithm	BoW	Tri-Grams	(2,5)-Grams	+ExtraFeatures
LR	0.764	0.787	0.815	0.816
RF	0.774	0.770	0.770	0.773
SVC	0.800	0.802	0.801	0.799
GB	0.743	0.758	0.767	0.784
XGB	0.791	0.819	0.837	0.847

The results varied depending on the language being analyzed. In certain instances, specific types of characteristics did not have a significant impact or improve the results. For instance,

the character length did not enhance the performance of the models in English or Spanish. Therefore, the inclusion of this characteristic was unnecessary for both languages. Another example pertains to the use of stop-words. Initially, we hypothesized that stop-words would have a significant influence, but we did not observe any substantial difference in their usage and their impact on the results.

The most effective approach was utilizing (3,3)-Grams for SVC and (2,5)-Grams with extra features for the other models. In terms of subtask_1 in English, Table 5 indicates that the XGB algorithm achieved the highest score of 84.7% using (2,5)-Grams and adding ExtraFeatures. As for subtask_1 in Spanish, the XGB algorithm attained a score of 85.9% (see Table 6). In subtask_2, we achieved maximum scores of 49.5% and 52.2% for English and Spanish, respectively. This information is presented in Tables 7 and 8.

Table 6

F1 Scores of Subtask_1_es

Algorithm	BoW	Trigrams	(2,5)grams	+ExtraFeatures
LR	0.792	0.800	0.826	0.825
RF	0.790	0.786	0.802	0.813
SVC	0.812	0.803	0.802	0.800
GB	0.733	0.754	0.766	0.789
XGB	0.747	0.820	0.843	0.859

Table 7

F1 Scores of Subtask_2_en

Algorithm	BoW	Trigrams	(2,5)grams	+ExtraFeatures
LR	0.339	0.398	0.445	0.446
SVC	0.458	0.465	0.442	0.421
GB	0.366	0.410	0.472	0.496
XGB	0.439	0.485	0.491	0.495

Table 8

F1 Scores of Subtask_2_es

Algorithm	BoW	Trigrams	(2,5)grams	+ExtraFeatures
SVC	0.513	0.522	0.517	0.498
XGB	0.506	0.507	0.519	0.522

5. Conclusions

In this study, we conducted experiments to determine the optimal algorithm and feature extraction method for identifying texts generated by artificial intelligence or written by humans.

Additionally, we examined texts authored by different artificial intelligence systems. Our findings revealed that the XGB model using (2,5)-Grams and adding stylometric features, performed best across all subtasks. However, while the results were favorable for Subtask 1, they were less promising for Subtask 2.

Moreover, in the experiments conducted by Daphne Ippolito, the highest achieved score was 70% [4]. We obtained superior results in identifying human and machine-written texts through machine learning compared to working with annotators. However, we struggled to differentiate between various machine-generated texts, with scores of 49.5% and 52.2%. These results indicate a reliance on probability rather than the model's ability to accurately classify the information.

These outcomes emphasize the need to continue improving our experiments and exploring new strategies for identifying written and generated texts. Utilizing linguistic features, we observed that factors such as the number of punctuation marks, digits, symbols, capital letters, etc., contributed to achieving improved results.

6. Appendix

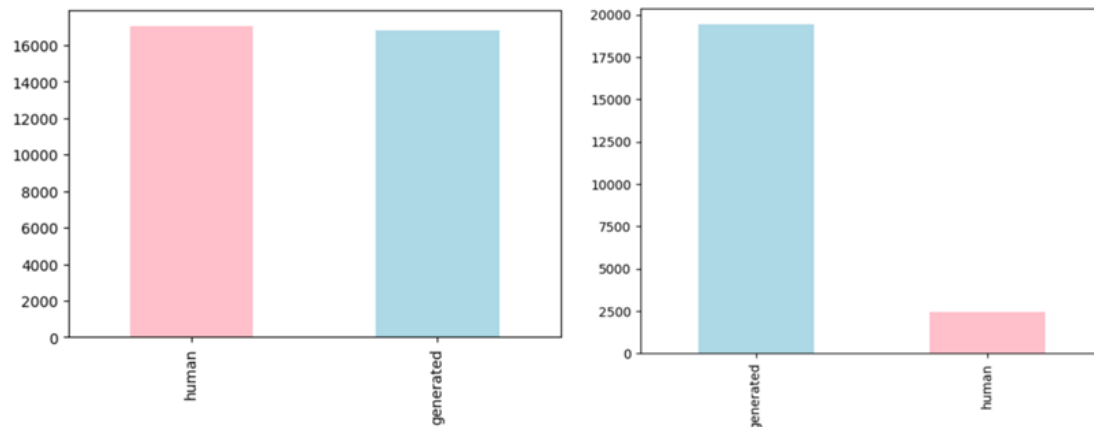


Figure 1: Train data subtask1 English / Predictions of test data subtask1 English

Acknowledgments

This work has been carried out with the support of DGAPA-UNAM PAPIIT project number TA101722. The authors also thank CONACYT for the computing resources provided through the Deep Learning Platform for Language Technologies of the INAOE Supercomputing Laboratory. We also want to thank Eng. Roman Osorio for supporting the student administration of the project.

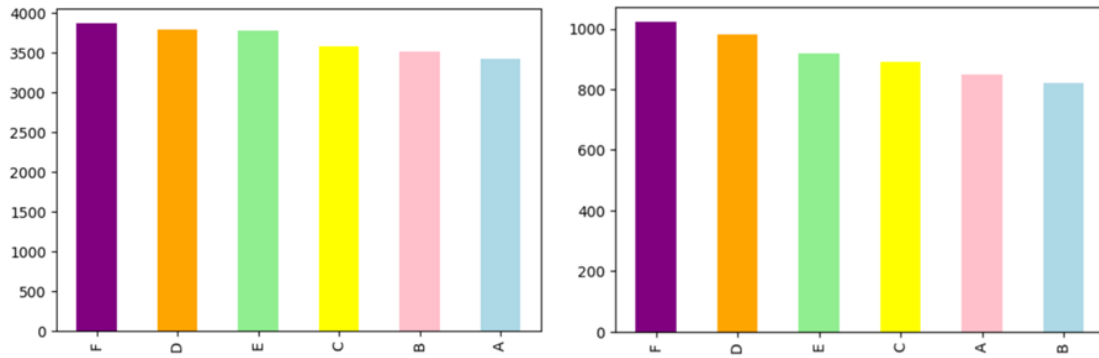


Figure 2: Train data subtask2 Spanish / Predictions of test data subtask2 Spanish

References

- [1] ChatGPT, 2023. URL: <https://chat.openai.com/>.
- [2] A. M. Sarvazyan, J. Á. González, M. Franco Salvador, F. Rangel, B. Chulvi, P. Rosso, Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains, in: *Procesamiento del Lenguaje Natural*, Jaén, Spain, 2023.
- [3] S. M. Jiménez-Zafra, F. Rangel, M. Montes-y Gómez, Overview of IberLEF 2023: Natural Language Processing Challenges for Spanish and other Iberian Languages, *Procesamiento del Lenguaje Natural* 71 (2023).
- [4] L. Dugan, D. Ippolito, A. Kirubarajan, C. Callison-Burch, Roft: A tool for evaluating human detection of machine-generated text, *arXiv preprint arXiv:2010.03070* (2020).
- [5] K. Ethayarajh, D. Jurafsky, The authenticity gap in human evaluation, *arXiv preprint arXiv:2205.11930* (2022).
- [6] E. Clark, T. August, S. Serrano, N. Haduong, S. Gururangan, N. A. Smith, All that's 'human' is not gold: Evaluating human evaluation of generated text, *arXiv preprint arXiv:2107.00061* (2021).
- [7] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, S. Feizi, Can ai-generated text be reliably detected?, *arXiv preprint arXiv:2303.11156* (2023).
- [8] OpenAI, Ai text classifier, 2023. URL: <https://beta.openai.com/ai-text-classifier>.
- [9] Scikit-learn, ????. URL: <https://scikit-learn.org/stable/>.