Taming the Turing Test: Exploring Machine Learning Approaches to Discriminate Human vs. AI-Generated Texts

Alberto Fernández-Hernández[1], Juan Luis Arboledas-Márquez[2], Julián Ariza-Merino[2] and Salud María Jiménez-Zafra[3,*]

[1] *Enterprise Business Unit Department, Vodafone Group Plc, 28042, Spain*
[2] *OpenSpring IT IBERIA S.L., 28036, Spain*
[2] *OpenSpring IT IBERIA S.L., 28036, Spain*
[3] *Computer Science Department, SINAI, CEATIC, Universidad de Jaén, 23071, Spain*

Abstract

This paper describes the participation of the turing-testers team in the AuTexTification shared task, at IberLEF 2023, on Automated Text Identification. In this shared task, two subtasks has been proposed for English and Spanish: subtask 1, on determining whether a text has been automatically generated or it is a human-written text and, subtask 2, on the attribution of Large Language Models to automatically generated texts. We have addressed subtask 1 in both languages testing traditional machine and deep learning approaches and exploring with the integration of metadata related to readability, comprehensibility, complexity, sentiment, emotion and toxicity of texts. In total, 76 runs were submitted for subtask 1 in English and 52 runs for subtask 1 in Spanish. The best run sent by our team placed in position 27th for English and position 9th for Spanish with a macro F1-score of 64.32 and 66.5, respectively.

Keywords

Automatically generated texts, human-written texts, large language models, human vs. ai generated texts

1. Introduction

The generation of automatic content through Natural Language Processing (NLP) systems with powerful Large Language Models (LLMs) such as Generative Pre-trained Transformer (GPT) [1, 2], Language Model for Dialogue Applications (LaMDA) [3, 4], Pathways Language Model (PaLM) [5], BLOOM [6], ChatGPT [7, 8] or Bard [9] has been a breakthrough. These models are pre-trained on a large amount of human texts from the Internet and have been designed to automatically generate content. Most of them are publicly available, which enables its research and good use in different fields and tasks. However, they can also be used by malicious users or bots to spread fake news, reviews or opinions [10], to help cybercriminals to prepare malicious code and write credible and personalized phishing messages, or even to prompt injection attacks

IberLEF 2023, September 2023, Jaén, Spain

*Corresponding author.
✉ albertofernandezh98@gmail.com (A. Fernández-Hernández); juanluis.arboledas@openspring.com (J. L. Arboledas-Márquez); julian.ariza@openspring.com (J. Ariza-Merino); sjzafra@ujaen.es (S. M. Jiménez-Zafra)
iD 0000-0003-3274-8825 (S. M. Jiménez-Zafra)

CEUR Workshop Proceedings (CEUR-WS.org)

[11]. This is already a reality and has led some conferences such as the 40th International Conference on Machine Learning (ICML 2023) to ban text generated by LLMs unless the text produced is presented as part of the experimental analysis of the work.

In this context, it is organized the shared task *AuTexTification: Automated Text Identification* [12], as part of IberLEF 2023 [13], a shared evaluation campaign for NLP systems in Spanish and other Iberian languages. The goal of this task is to advance research on the detection of automatically generated text by developing systems that exploit clues about the linguistic form and meaning of texts to distinguish whether they have been automatically generated or whether they are human-written texts. Specifically, two subtasks are proposed, both in English and Spanish. The first one to classify a text as human or generated and, the second one, to determine which text generation model created an automatically generated text.

We have participated in substask 1 for both languages. The primary objective of this subtask is to classify a given text as either human-generated or automatically generated. To address the problem, the current State Of the Art (SOTA) research and commercial approaches predominantly rely on transformer models [10][14], which have proven to be highly effective in solving such problems. However, AI-generated text often differs from human-authored text in various lexical, complexity and grammatical aspects [15]. While transformers have shown remarkable success, it is essential to compare their performance against traditional machine learning approaches, in order to shed light on the strengths and limitations of each approach. Consequently, we have made a decision to explore an alternative traditional machine learning approach, employing these aspects via feature engineering. By selecting the most relevant features, we aim to evaluate and compare a "traditional Machine Learning approach" based exclusively in specific linguistic features against modern millions-of-parameters transformers architecture.

The rest of the paper is structured as follows. Section 2 provides a description of the task and the dataset. Next, Section 3 presents the methodology we followed for detecting human and automatically generated texts. Subsequently, Section 4 shows all the details related to the experimental setup. Later, the results obtained and a discussion of them are reported in Section 5. Finally, Section 6 summarizes the main insights and draws future work directions.

2. Task description

The AuTexTification shared task aims to differentiate automatically generated texts from human texts and to identify the generation model used for the first case. For this, two subtasks are proposed:

- **Subtask 1: Human or Generated**. It is a binary classification task in which, given and Spanish or English text, it should be determined if the text has been automatically generated or by contrast it is from a human.
- **Subtask 2: Model Attribution**. It is a multi-class classification task which consists of, given an automatically generated text in Spanish or English, identifying which text model has generated it. The possible classes are A, B, C, D, E or F. Each class represent a text generation model and the model label mapping is: "A" - "bloom-1b7", "B" - "bloom-3b", "C" - "bloom-7b1", "D" - "babbage", "E" - "curie", "F" - "text-davinci-003".

To participate in the task, the organizers established the following restrictions:

1. Pre-trained models publicly available in the literature can be used. However, **only text derived from training data is allowed to be used**. That is, data augmentation, additional self-supervised pre-training or other techniques involving the use of additional text must be performed only with text derived from the training data.
2. The use of knowledge bases, lexicons and other structured data resources is **allowed**.
3. The use of data from one subtask in another subtask is **not allowed**.

Registered teams could participate in any of the tasks for any language and three system submissions were allowed per task and language. As it has been mentioned in the Introduction section, our team participated in subtask 1 for both Spanish and English. All subtasks were evaluated and ranked using the macro F1-score. The official evaluation script is available at the following link https://github.com/autextification/AuTexTificationEval and all the details of the competition can be found at https://sites.google.com/view/autextification.

On the other hand, regarding the dataset provided [16], it consists of texts from five different domains, including legal documents, how-to articles and social media, to cover a wide variety of writing styles: from more structured and formal to less structured and informal. At a first stage, training was made available in order to the participants develop their systems. Later, test sets were released to rank the participant systems. The distribution of the dataset for *Subtask 1: Human or Generated* is presented in Table 1. We do not report statistics from the subtask 2 dataset as we did not participate in it.

Table 1
Dataset for Subtask 1: Human or Generated

lang	label	train	test	total
es	generated	16,275	11,209	27,484
	human	15,787	8,920	24,707
	total	32,062	20,129	52,191
en	generated	16,799	11,190	27,989
	human	17,046	10,642	27,688
	total	33,845	21,832	55,677

3. Methodology

We present the methodology employed in our study for detecting text generated automatically using different approaches. By delineating the methodology, we aim to provide a comprehensive understanding of the techniques and procedures used in our research, ensuring transparency and reproducibility. We first outline the Exploratory Data Analysis (EDA) performed and then the approaches used, from a traditional Machine Learning approach (i.e. using tabular data) to Deep Learning using transformers.

3.1. Exploratory Data Analysis

In this subsection, we take a deep dive into the train dataset provided by the organizers. As it can be seen in Table 1, it contains a similar amount of texts written in English and Spanish and a balance distribution between both labels: human and generated.

In order to decide which model is more suitable for this subtask, we performed an analysis of the length of the texts from two points of view: number of words and number of sentences.

3.1.1. Number of words

The strategy followed to calculate the total number of words was to divide the texts using the white space character. Next, Figure 1 and Figure 2 show the distribution by label and language, respectively.

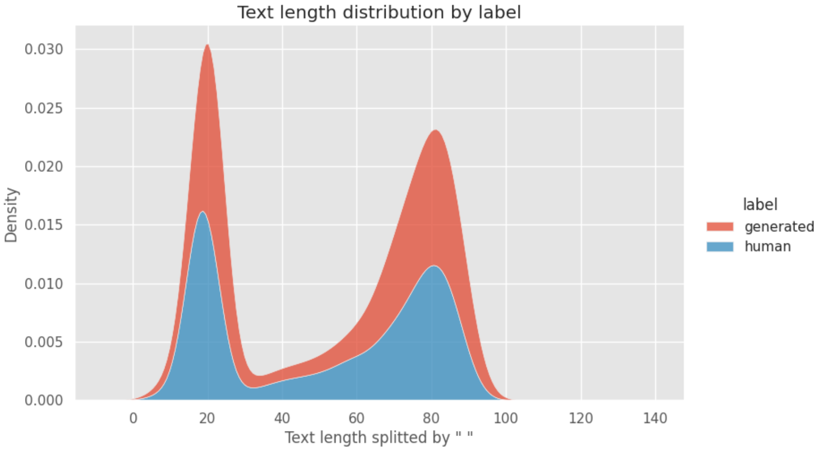


Figure 1: Number of words distribution by label

3.1.2. Number of sentences

On the other hand, to calculate the number of sentences per text, the following regular expression was used:

```
r‘(?<!\w\.\w.)(?<![A-Z][a-z]\.)(?<=\.|\?\!)\s’
```

It considers any alphanumeric character as a sentence until a sentence separator (period, question mark or exclamation mark) is found. The distribution of sentences per label and language are shown in Figure 3 and Figure 4, respectively.

3.1.3. Discussion

By analyzing the text length by number of words and number of sentences, our analysis reveals that there is no significant difference in both features between human and machine-generated texts. This finding suggests that solely relying on number of words or number of sentences as

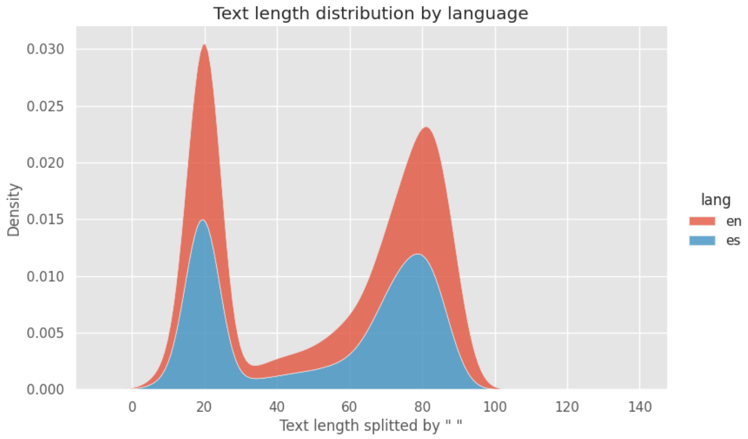


Figure 2: Number of words distribution by language

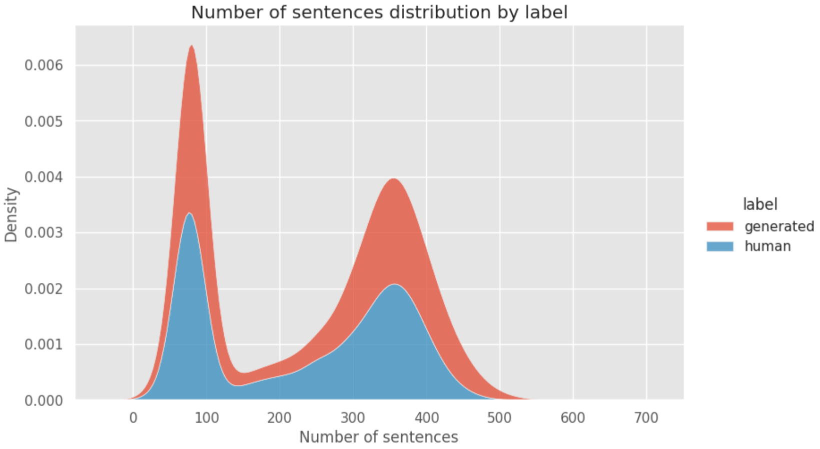


Figure 3: Number of sentences distribution by label

features for distinguishing between human and machine-written texts may not be effective. However, despite not showing any correlation at first sight, it is important to consider that there may exist other features that have a hidden linear or non-linear relationship with the target variable. Therefore, we decided to engage in extensive feature engineering to uncover additional text features that could be informative for our approach.

Furthermore, we observe that the maximum text length across all texts is approximately 80. This observation indicates that the dataset primarily consists of short-length texts. Consequently, this presents an opportunity to employ transformer-based models such as BERT [17], as they can handle sequences up to the maximum input size without sacrificing performance.

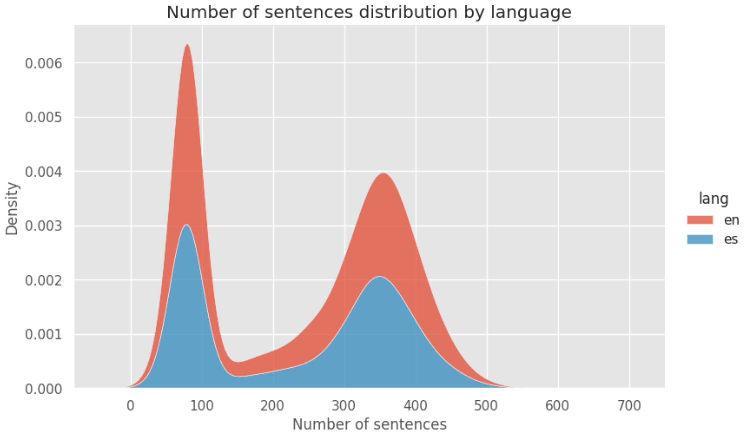


Figure 4: Number of sentences by language

3.2. Approaches

Our study consists of testing three different approaches:

- Traditional Machine Learning based on textual features. As LLMs models are becoming increansingly popular, we want also to explore a traditional approach based on tabular data, i.e. text-based features, avoiding complexity of millions or billions of parameters used in transformers and the amount of resources they need.
- Deep Learning approach: transformers. As LLMs are giving extraordinary results on several text-based tasks, we also explore this approach.
- Hybrid solution: combining the Deep Learning approach with the best metadata from the first solution.

3.2.1. Approach 1: Traditional Machine Learning based on textual features

Machine learning models play a crucial role in recent driven-data applications, a traditional approach (compared to recent Deep Learning techniques) that requires a critical step: transforming raw data into meaningful features, or a manual feature engineering. Consequently, in this subsection we deep dive into the features chosen for this use case.

Throught Human vs recent ChatGPT Comparison Corpus, differences and gaps between AI-generated and Human texts have been extracted [15], including:

- AI-generated texts shows less bias and harmful information, "neutral on sensitive topics", barely showing any sign of political tendency or discriminatory toxicity. Consequently, toxicity features (e.g via Transformers outputs), combined with transformer-based sentiment features, can be crucial to determine whether a text is subjective or apparently neutral.

- It expresses less emotion in its responses, whereas human add extra punctuation signs and grammar features in context to express their feelings. As a result, transformers-based sentiment and emotional features could determine if the text shows any signs of "sentimentalism".
- Compared to ChatGPT, human answers are relatively shorter. In this case, general features including number of characters, syllables, words, sentences or text depth could be useful to classify AI-generated texts.
- On the other hand, human texts tend to use larger vocabulary. Lexical-based features that measure text/vocabulary complexity could be used to evaluate it.
- Related to Part of Speech, ChatGPT-like models gennerally use more nouns, verbs, determinants and auxiliary verbs, while using less adverbs. According to the study, texts that contains a high proportion of nouns and verbs often signifies an argumentative nature, showing informativeness and objectivity.

Based on the foregoing, related features are taken into account to train and evaluate a Machine Learning model.

General features

We take into account the following general features:

1. Number of characters (nchar)
2. Number of syllables (nsyllab)
3. Number of words (nword)
4. Number of rare words, infrequency words (nrword)
5. Number of punctuation signs (PUNCT)
6. Number of sentences (nsent)
7. Number of complex sentences, those with composed verbs (ncompsent)
8. Average sentence length (avgsentl)
9. Number of nouns (NOUN)
10. Number of digits or numeric nouns (NUM)
11. Number of proper nouns (PROPN)
12. Number of pronouns (PRON)
13. Number of determinants (DET)
14. Number of verbs (VERB)
15. Number of auxiliary verbs (AUX)
16. Number of adverbs (ADV)
17. Number of punctuation signs (PUNCT)

Readability features

Readability refers to the ease with which a reader can understand a given text. It is influenced by factors such as sentence structure, word complexity, and overall linguistic coherence. By assessing the readability of a text, the objective is to determine whether text complexity and readability can be used to distinguish between human and generated texts. To do so, several readability features are included. For Spanish, we use the Spaulding's readability score [18], the Huerta's readability index [19], the Flesch-Szigrist legibility measure[20], the Crawford Score [21], and the SOL readability [22] index. For English, we employ the ARI Index [23], the Mu readability score [24], the Minimum Age index [25]. The readability features have been extracted using the text-complexity library [26].

Comprehensibility score: Polini

The comprehensibility score of Gutiérrez de Polini et al. [27], typically used for school texts (mostly 6th grade), it is designed to measure text comprehensibility taking into account the number of words, the average number of letters per word and their variance. We have included this feature in the experimentation with both languages, English and Spanish. The Polini comprehensibility score has been calculated using the text-complexity library [26].

Complexity features

Text complexity is how easy or hard a text is to read, based on quantitative and qualitative text features. In this work, we have analyzed the following complexity-related features:

1. Texts depth [28]. The depth of syntactic parse trees is also a useful metric to capture syntactic complexity as long sentences can be syntactically complex or contain a large number of modifiers. Related to texts depth, three new features are explored:
   a) Maximum depth of a text.
   b) Minimum depth of a text.
   c) Average sentence depth.
2. Index of Low Frequency Words or ILFW [29]. It is based on the number of different content words per sentence (Lexical Complexity Index, LC) and on measuring the number of low frequency words per 100 content words.
3. Sentence Complexity Index or SCI [29]. This index measures the number of words per sentence.
4. Lexical Diversity Index or LDI, by dividing the number of distinct content (Part of Speech) words by the total number of sentences.
5. Lexical Complexity Index or LC, by dividing LDI by ILFW.

All these complexity features have been taken into account for English and Spanish and have been extracted using the text-complexity library [26].

Sentiment, emotion and toxicity features

We conduct multiple sentiment, emotional, and toxicity analysis tasks, based on the inference outputs of different transformer models.

1. Sentiment features: twitter-xlm-roberta-base-sentiment from Cardiff University [30]
   - Sentiment positive (fe_text_sentiment_pos)

  - Sentiment neutral (fe_text_sentiment_neut)
  - Sentiment negative (fe_text_sentiment_neg)

2. Emotion features: xlm-emo-t, from Milan Natural Language Processing Group [31]
  - Joy (fe_text_emotion_joy)
  - Sadness (fe_text_emotion_sadness)
  - Fear (fe_text_emotion_fear)

3. Toxicity features: Detoxify library [32]
  - Toxic (toxic)
  - Very toxic (severe_toxicty)
  - Obscene (obscene)
  - Identity attack (identity_attack)
  - Insult (insult)
  - Threat (threat)
  - Sexual explicit (sexual_explicit)

3.2.2. Approach 2: Deep Learning

As maximum text length across all texts is not extremely large, LLMs or its variants can be also applied. After consideration, we have decided to use BERT [17] as our base pre-trained model, concretely its multilingual version [33], fine-tuning both uncased and cased versions. Thus, we avoid calibrating two different transformers, one for each language.

3.2.3. Approach 3: Deep learning combined with text-based features

Finally, last use case consists of integrating trained multilingual BERT transformer from previous use case with metadata features. Thus, we create a more comprehensive and robust representation of the text. This fusion could allow the model to benefit from both the rich contextual information captured by BERT and the additional insights provided by the metadata features. Figure 5 shows the diagram of this approach.

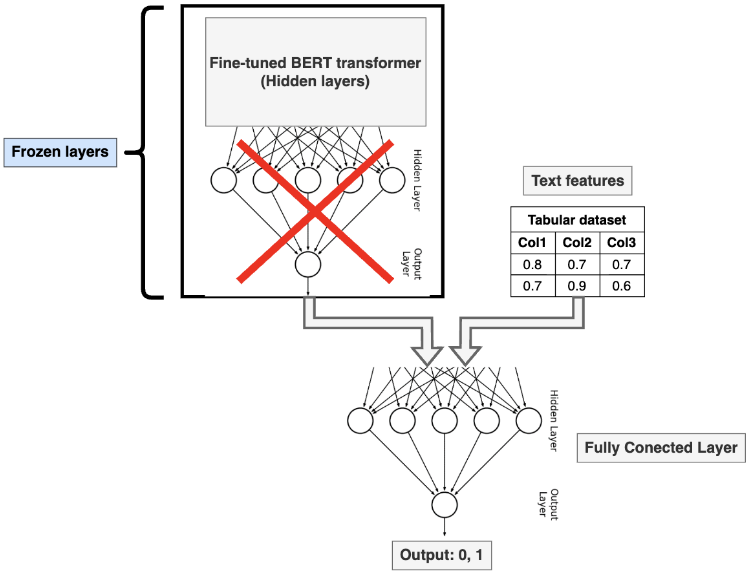
4. Experimental setup

In this section, we will delve into the methodology used to train the model, including the libraries, techniques and parameters employed.

4.1. Approach 1: Traditional Machine Learning based on textual features

4.1.1. AutoML library

To streamline the training process and leverage its powerful machine learning capabilities, we utilized the PyCaret library: an open-source, low-code machine learning library in Python that automates various steps in the machine learning workflow, including data preprocessing, feature selection, model training, hyperparameter tuning, and evaluation[34].



Tabular dataset

Col1	Col2	Col3
0.8	0.7	0.7
0.7	0.9	0.6

Figure 5: Approach 3 diagram

4.1.2. Train-validation split

We initially split our dataset into a training set and a validation set. We allocated 70% of the data for training purposes and reserved the remaining 30% for evaluating the final model's performance in the development phase.

4.1.3. Feature Selection

To enhance the model's predictive capabilities and reduce the dimensionality of the dataset, we employed feature selection techniques. In particular, we utilized the SelectFromModel function from the scikit-learn library [35]: http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFromModel.html. This method allows us to select the most important features by training an estimator and extracting the top-ranked features based on their importance scores. By doing so, we can focus on the most relevant features, which often improves the model performance.

4.1.4. Model Selection

After performing feature selection, we trained multiple machine learning models using PyCaret's automated workflow (CatBoost, Random Forest, extreme Gradient Boosting, Light Gradiente Boosting Machine, Gradient Boosting Classifier, Decision Tree, Logistic Regression, and Quadratic Discriminant Analysis). PyCaret supports a wide range of algorithms and provides a convenient way to compare their performance on the dataset. To select the top-performing models, we sorted them based on their F1 scores, a common metric used in classification tasks that balances precision and recall. The top four models were chosen for further evaluation and comparison.

Overall, the training process involved using PyCaret's automated workflow to preprocess the data, split the dataset into train and test sets, apply feature selection, and train multiple models. The top-performing models were then selected based on their F1 scores, setting the stage for subsequent evaluation and fine-tuning of the models to achieve optimal results.

4.2. Approaches 2 & 3: Deep Learning

The fine tuning process has been carried out using the PyTorch library, carefully selecting the parameters to suit our specific requirements. The learning rate was set to 5e-05. Furthermore, a weight decay of 0.01 was applied to prevent overfitting and improve the model's generalization capabilities. In addition, we opted for the Binary Cross Entropy Loss (BCELoss) function, particularly suitable for binary classification tasks.

To ensure a thorough training process, we performed 15 epochs during the fine-tuning phase, adding an early_stopping parameter to ensure best model is saved.

5. Results and discussion

In this section, we report and discuss the results obtained during the development phase and the official results achieved in the evaluation phase for subtask 1, on determining whether a text has been automatically generated or by contrast it is a human-written text.

5.1. Results in the development phase

5.1.1. Approach 1

In the first approach we tested different traditional machine learning models with textual based features (general, readability, comprehensibility, complexity, sentiment, emotion and toxicity features). Specifically, we have trained with nine different models. Table 2 shows the results of each of the model and the training time (TT). As it can be seen, the best performing strategy in the binary classification subtask was the CatBoost Classifier for all the evaluation measures, except for the recall in which the Random Forest Classifier achieved a slightly better result.

The goal of this experiment was to find out which features could be help to discern between texts written by humans and texts automatically generated with LLMs. The feature importance of the four best models are shown in Figure 6.

Although the distribution of feature importance varies across the models, certain variables are consistently identified as key features for all of them:

- Number of words (nword)
- Number of sentences (nsent)
- Number of characters (nchar)
- Average sentence length (avgsentl)
- Lexical Complexity Index or LC
- Polini's compressibility

model	description	accuracy	AUC	recall	precision	TT (Sec)
catboost	CatBoost Classifier	**0.7854**	**0.8724**	0.7881	**0.7851**	27.5410
rf	Random Forest Classifier	0.7752	0.8598	**0.7904**	0.7683	12.1950
xgboost	Extreme Gradient Boosting	0.7723	0.8593	0.7745	0.7725	11.1620
lightgbm	Light Gradient Boosting Machine	0.7702	0.8556	0.7801	0.7663	1.4380
gbc	Gradient Boosting Classifier	0.7308	0.8131	0.7389	0.7285	15.6090
ada	Ada Boost Classifier	0.6943	0.7691	0.6903	0.6974	3.7990
dt	Decision Tree Classifier	0.6757	0.6757	0.6715	0.6788	1.1930
lr	Logistic Regression	0.6312	0.6945	0.6226	0.6352	5.5550
qda	Quadratic Discriminant Analysis	0.6285	0.7275	0.3869	0.7528	**0.2840**

Table 2
Performance Metrics for approach 1 (traditional machine learning models with metadata) in the development phase

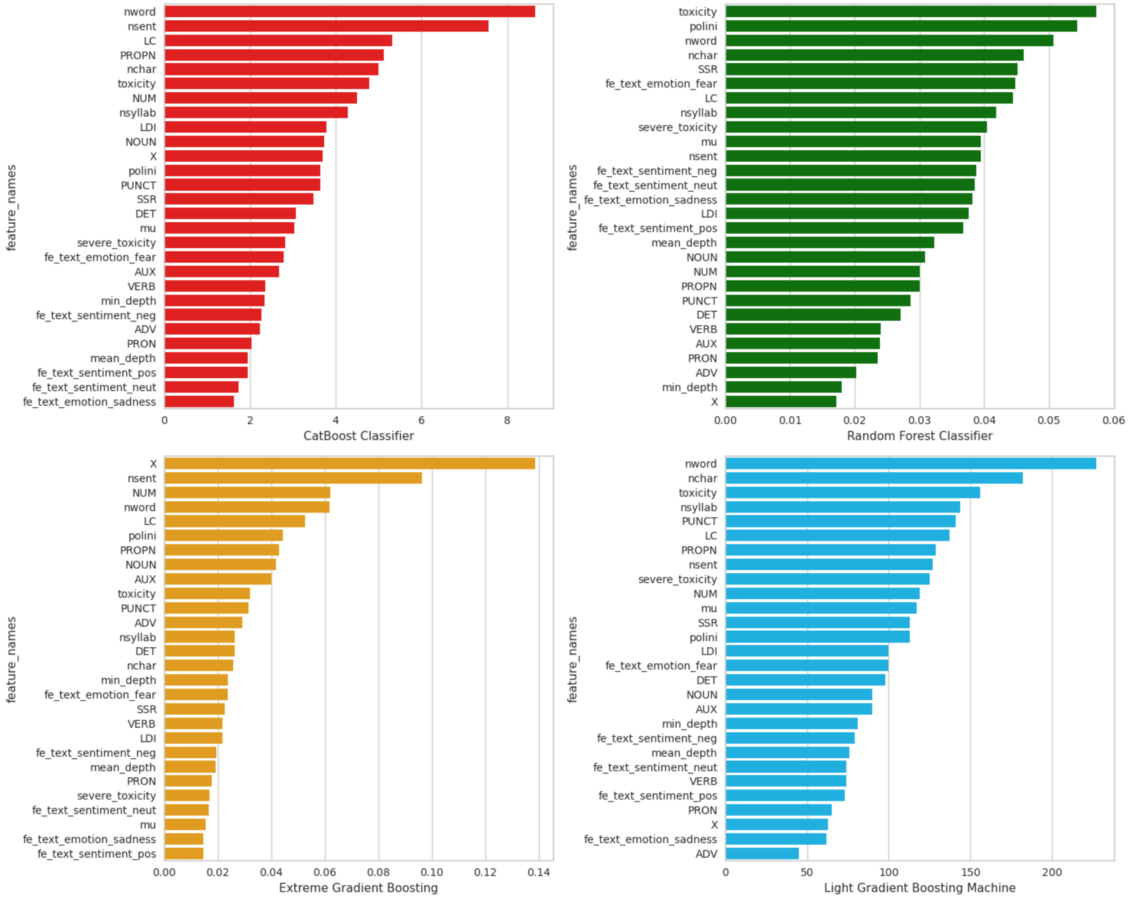


Figure 6: Feature importance for best models

- SSR index
- LDI index

5.1.2. Approach 2

In the second approach, we evaluated the Multilingual BERT versions: cased and uncased, fine-tuning them. After modeling, results are obtained for both versions, as shown in Table 3. Analyzing the results, it was observed that the cased version produced marginally superior scores. However, considering the insignificant difference in performance, we have decided to utilize the uncased version due to its simplicity.

Table 3
Fine-tuning results for approach 2 (deep learning) in the development phase

model type	training loss	validation loss	accuracy	precision	recall	f1
mult. BERT uncased	**0.149700**	0.267662	0.922773	0.897985	0.954888	0.925563
mult. BERT cased	0.158400	**0.248109**	**0.923380**	**0.898779**	**0.955190**	**0.926126**

5.1.3. Approach 3

The third approach consists of training the trained multilingual BERT model from the previous section, coupled with additional text features obtained from approach 1. Among the various potential features available, the Number of Sentences, Lexical Complexity Index, Average Sentence Length, Polini's Compressibility, SSR, and LDI indexes were determined to be the most suitable choices. These six features were selected due to their non-trivial nature and the difficulty in calculating and interpreting them using transformers. In contrast, features such as toxicity, sentiment, and emotions are relatively easier to calculate and interpret with them. The chosen method represents a comprehensive and nuanced approach to training a model that incorporates a range of complex textual features that traditional transformer-based models do not easily capture. Table 4 presents the results of this third approach in which it can be seen that the new model slightly performs the results obtained from the previous section.

Table 4
Results for approach 3 (deep learning with metadata) in the development phase

model type	training loss	validation loss	accuracy	recall	f1
BERT uncased + metadata	0.0100	0.1587	0.9354	0.9926	0.9392

5.2. Results in the evaluation phase: official results

This section presents the results we obtained in the evaluation phase of the AuTexTification shared task for subtask 1 in English and Spanish. The organizers selected the macro F1-score to evaluate and rank the runs submitted by the participating teams. Each team were allowed to make a maximum of 3 submissions for each language. We selected our 3 runs based on the experiments conducted on the development phase. The results for each of the runs and the models used, are shown in Table 5.

The official results for subtask 1 in English and Spanish are presented in Table 6 and Table 7, respectively. These tables show the results of the submissions ranked in the top three positions, in the middle, in the last position and the three runs of our team, the turing-testers.

The best results for discriminating between human texts and automatically generated texts were obtained for both English and Spanish with the model multilingual BERT uncased fine-tuned from run2, placed in position 27th for English and position 9th for Spanish with a macro F1-score of 64.32 and 66.5, respectively, compared to the 80.91 and 70.77 scores obtained by the teams ranked in 1st position.

The rest of the runs in which metadata is included in a traditional machine learning model and in a deep learning model work similarly in English, but in Spanish there is a slight difference in which the shallow learning model provides better results.

Table 5
Results for subtask 1 in the evaluation phase

run	Model	Macro-F1 (en)	Macro-F1 (es)
run1	CatBoost Classifier	60.07	62.77
run2	mult. BERT fine-tuned	64.32	66.05
run3	mult. BERT trained from run2 (frozen layers) + metadata	60.64	59.23

Table 6
Official leader-board for subtask 1 (English)

Rank	Team	Run	Macro-F1
1	TALNP-UPF	Hybrid_plus	80.91
2	TALNP-UPF	Hybrid	74.16
3	CIC-IPN-CsCog	run2	74.13
...			
27	**turing-testers**	**run2**	**64.32**
...			
33	**turing-testers**	**run3**	**60.64**
...			
35	**turing-testers**	**run1**	**60.07**
...			
38	AIUB	run1	59.40
...			
76	UAEMex	run1	33.87

6. Conclusions and future work

In this paper, we have described the participation of the turing-tester team in subtask 1 of the shared task AuTexTification in IberLEF 2023, related to distinguishing human texts from texts generated by artificial intelligence.

In conclusion, our experiments demonstrate the effectiveness of fine-tuning multilingual BERT uncased for detecting AI-generated texts. On the other hand, incorporating additional features to our LLM achieves precisely the opposite effect than expected: the metrics worsen even when compared to a traditional ML model, suggesting that BERT architecture is capable of retrieving enough text-based information, features, and context to classify them.

Although our experiments provide insight into the effectiveness of different models in detecting AI-generated text, much remains to be explored in this area. One avenue for future research is error analysis to identify and understand misclassified texts, typically via Explainable-AI tools. We also plan to explore other model architectures and additional text features to further enhance model performance. For example, recent advances in transformer-based models, such as GPT-3 or LLaMA/LLaMA-2, have shown promising results in detecting AI-generated texts. Investigating the performance of such models and comparing them to our approach would provide valuable insights into the effectiveness of different model architectures.

Acknowledgments

This work has been partially supported by Project CONSENSO (PID2021-122263OB-C21), Project MODERATES (TED2021-130145B-I00) and Project SocialTox (PDC2022-133146-C21) funded by MCIN/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR, Project PRECOM (SUBV-00016) funded by the Ministry of Consumer Affairs of the Spanish Government, Project FedDAP (PID2020-116118GA-I00) supported by MICIN-

Table 7
Official leader-board for subtask 1 (Spanish)

Rank	Team	Run	Macro-F1
1	TALNP-UPF	Hybrid_plus	70.77
2	Linguistica_F-P_et_al	run1	70.60
3	ROBERTA (BNE)	baseline	68.52
...			
9	**turing-testers**	**run2**	66.05
...			
24	**turing-testers**	**run1**	62.77
...			
26	turquoise_titans	run1	61.79
...			
32	**turing-testers**	**run3**	**59.23**
...			
52	LKE_BUAP	run3	31.60

N/AEI/10.13039/501100011033 and WeLee project (1380939, FEDER Andalucía 2014-2020) funded by the Andalusian Regional Government. Salud María Jiménez-Zafra has been partially supported by a grant from Fondo Social Europeo and the Administration of the Junta de Andalucía (DOC_01073), and the article 83 contract with the company OpenSpring IT IBERIA S.L. (EXP. 2022_161).

References

[1] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.

[2] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., Training language models to follow instructions with human feedback, Advances in Neural Information Processing Systems 35 (2022) 27730–27744.

[3] D. Adiwardana, T. Luong, Towards a conversational agent that can chat about… anything, Google AI Blog (2020).

[4] E. Collins, Z. Ghahramani, Lamda: our breakthrough conversation technology, Google AI Blog (2021).

[5] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al., Palm: Scaling language modeling with pathways, arXiv preprint arXiv:2204.02311 (2022).

[6] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, et al., Bloom: A 176b-parameter open-access multilingual language model, arXiv preprint arXiv:2211.05100 (2022).

[7] OpenAI. (2023). ChatGPT (May version) [Large language model], https://chat.openai.com, 2023.

[8] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu, Z. Wu, D. Zhu, X. Li, N. Qiang, D. Shen, T. Liu, B. Ge, Summary of chatgpt/gpt-4 research and perspective towards the future of large language models, 2023. arXiv:2304.01852.

[9] Bard Google, https://https://bard.google.com/, 2023.

[10] G. Jawahar, M. Abdul-Mageed, V. Laks Lakshmanan, Automatic detection of machine generated text: A critical survey, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 2296–2309.

[11] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, M. Fritz, More than you've asked for: A comprehensive analysis of novel prompt injection threats to application-integrated large language models, arXiv preprint arXiv:2302.12173 (2023).

[12] 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.

[13] S. M. Jiménez-Zafra, F. Rangel, M. Montes-y Gómez, Overview of IberLEF 2023: Natural Language Processing Challenges for Spanish and other Iberian Languages, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2023), co-located with the 39th Conference of the Spanish Society for Natural Language Processing (SEPLN 2023), Online. CEUR. org, 2023.

[14] A. Pegoraro, K. Kumari, H. Fereidooni, A.-R. Sadeghi, To chatgpt, or not to chatgpt: That is the question!, arXiv preprint arXiv:2304.01487 (2023).

[15] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, Y. Wu, How close is chatgpt to human experts? comparison corpus, evaluation, and detection, arXiv preprint arXiv:2301.07597 (2023).

[16] A. Sarvazyan, J. Ángel González, M. Franco, F. M. Rangel, M. A. Chulvi, P. Rosso, Autextification dataset (full data), 2023. URL: https://doi.org/10.5281/zenodo.7956207. doi:10.5281/zenodo.7956207.

[17] J. D. M.-W. C. Kenton, L. K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NAACL-HLT, 2019, pp. 4171–4186.

[18] S. Spaulding, A spanish readability formula, The Modern Language Journal 40 (1956) 433–441.

[19] J. Fernández Huerta, Medidas sencillas de lecturabilidad, Consigna 214 (1959) 29–32.

[20] F. Szigriszt Pazos, Sistemas predictivos de legilibilidad del mensaje escrito: fórmula de perspicuidad (1992).

[21] A. N. Crawford, A Spanish language Fry-type readability procedure: Elementary level, volume 7, Evaluation, Dissemination and Assessment Center, California State University . . . , 1984.

[22] A. Contreras, R. Garcia-Alonso, M. Echenique, F. Daye-Contreras, The sol formulas for converting smog readability scores between health education materials written in spanish, english, and french, Journal of health communication 4 (1999) 21–29.

[23] R. Senter, E. A. Smith, Automated readability index, Technical Report, Cincinnati Univ OH, 1967.

[24] M. M. Baquedano, Legibilidad y variabilidad de los textos, Boletín De Investigación Educacional [Artículo De Revista] 21 (2006) 13–25.

[25] J. A. García Lopez, Legibilidad de los folletos informativos, Pharm. care Esp (2001) 49–56.

[26] R. López-Anguita, J. Collado-Montañez, A. Montejo-Ráez, The text complexity library, Procesamiento del Lenguaje Natural 65 (2020) 127–130.

[27] L. Gutiérrez de Polini, et al., Investigación sobre lectura en venezuela, Primeras Jornadas de Educación Primaria (1972).

[28] H. Saggion, S. Štajner, S. Bott, S. Mille, L. Rello, B. Drndarevic, Making it simplext: Implementation and evaluation of a text simplification system for spanish, ACM Transactions on Accessible Computing (TACCESS) 6 (2015) 1–36.

[29] A. Anula, Lecturas adaptadas a la enseñanza del español como l2: variables lingüísticas para la determinación del nivel de legibilidad, La evaluación en el aprendizaje y la enseñanza del español como LE L 2 (2008) 162–170.

[30] F. Barbieri, L. Espinosa Anke, J. Camacho-Collados, Xlm-t: A multilingual language model toolkit for twitter, arXiv e-prints (2021) arXiv–2104.

[31] F. Bianchi, D. Nozza, D. Hovy, XLM-EMO: Multilingual emotion prediction in social media text, in: Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 195–203. URL: https://aclanthology.org/2022.wassa-1.18. doi:10.18653/v1/2022.wassa-1.18.

[32] L. Hanu, Unitary team, Detoxify, Github. https://github.com/unitaryai/detoxify, 2020.

[33] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186. URL: https://doi.org/10.18653/v1/n19-1423. doi:10.18653/v1/n19-1423.

[34] M. Ali, PyCaret: An open source, low-code machine learning library in Python, 2020. URL: https://www.pycaret.org, pyCaret version 1.0.0.

[35] 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, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.