Detecting LLM-Generated Texts

Amina Manseur[*]
École Nationale de la Statistique et de l'Administration Économique (ENSAE)
Data Science, Statistiques et Apprentissage (DSSA)
amina.manseur@ensae.fr

Abstract

The increasing deployment of Large Language Models (LLMs) raises new challenges for distinguishing human-written texts from AI-generated content. In this project, we address this issue through a binary classification task and compare various types of text representations. Our study contrasts deep embeddings (BERT, RoBERTa, Sentence-BERT), handcrafted linguistic features, and classical shallow representations (TF-IDF, Bag-of-Words) using standard machine learning classifiers. Experiments conducted on a balanced corpus demonstrate the superiority of deep embedding-based approaches, particularly RoBERTa, while also highlighting the trade-offs between model performance, interpretability, and computational efficiency. However, these detection methods remain sensitive to the characteristics of the training corpus and may see their performance decline when faced with texts from different thematic or stylistic contexts.

1 Introduction

The emergence of Large Language Models (LLMs) such as GPT-3, GPT-4 and LLaMA marks a turning point in the history of automatic natural language processing. These models can now generate fluid, coherent and structured texts that often closely resemble human writing, enabling their rapid adoption across a wide range of applications.

However, this rapid integration raises new issues relating to the traceability, authenticity and attribution of text production. For example, knowing whether a text has been generated by an LLM has become crucial for preserving academic integrity, ensuring the reliability of information in the media, protecting consumers against manipulated content, and maintaining transparency in critical sectors such as health, justice or administration.

In this context, automatically identifying the origin of a text - whether human or LLM - is becoming a necessity.

This problem is generally approached in the form of a supervised binary classification task, where the aim is to assign each text a label corresponding to its origin. While the task may seem natural, it remains complex: generative models are evolving rapidly, adopting increasingly diverse styles that are difficult to differentiate from human handwriting, while existing detectors are sometimes limited in robustness or generalization.

A comprehensive survey proposed by Wu et al. [2023] highlights four main families of methods used to detect texts produced by LLMs:

[*] ENSAE 3A, DSSA. https://github.com/AminaManseur29/ENSAE_NLP_project

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

1. **Statistical detectors**: These approaches are based on simple linguistic indicators such as perplexity [2], bigram frequency [3], burstiness [4], or textual entropy [5]. They exploit the differences in textual distribution between human and AI writing. Although fast, interpretable and requiring no access to source models, these methods often lack robustness in the face of reformulations, paraphrases or short texts.
2. **Neural detectors**: These methods use fine-tuned BERT, RoBERTa or DeBERTa architectures to classify texts according to their origin. Other variants rely on sentence embeddings extracted by models like Sentence-BERT, coupled with lightweight classifiers (SVM, random forests, MLP). Although effective at capturing complex latent signals, these detectors remain susceptible to adversarial editing attacks (paraphrasing, synonyms, reorganization) to fool automatic detectors without altering their meaning, and may suffer from a lack of cross-domain generalization.
3. **Watermarking techniques**: The idea here is to introduce a "watermark" into the generated text, by slightly skewing the selection of tokens or modifying the sampling distribution. This watermark, invisible to the naked eye, can be lexical or structural. While this approach works well for controlled models, it remains vulnerable to text transformations (paraphrasing, translation), and is unusable in the case of open-source models, or pre-existing content.
4. **Human-assisted methods**: Some approaches combine human judgment with analysis support tools (such as GLTR, which visualizes the probability of tokens). However, several studies have shown that humans alone have difficulty detecting well-generated texts, their performance often being comparable to chance in the absence of algorithmic assistance or specific training.

In parallel, Aich and Das [2022] proposed an interpretable and feature-based detection approach grounded in explicit linguistic characteristics — including stylistic markers (e.g., punctuation, use of capital letters), lexical complexity (e.g., TTR, MTLD), and psychological signals (e.g., sentiment score). Although this method shares similarities with traditional statistical approaches, it distinguishes itself by leveraging a structured and interpretable set of linguistic indicators, even proving effective against advanced generative models such as GANs

Building on this idea, the present project aims to systematically compare linguistic feature-based representations with deep embedding-based approaches. Using a balanced and annotated dataset (human-written texts vs. texts generated by GPT-2, GPT-3, and LLaMA), we benchmark the performance of several classical classifiers, evaluate the robustness of each feature representation, and explore the trade-offs between accuracy, interpretability, and computational efficiency in LLM-generated text detection.

2 Proposal and experimental justification

2.1 Choice of binary classification framework

As presented in the state-of-the-art [Wu et al., 2023], existing detection methods fall mainly into four categories: statistical methods, neural methods, watermarking, and human-assisted methods.

Among these approaches, statistical and neural methods are largely based on supervised classification techniques, where the aim is to assign a label ("Human" or "Generated by LLM") to each text, based on observable features.

Thus, formulating detection as a binary classification task naturally aligns with the main strategies in the literature, particularly in contexts where watermarking is not available and human intervention is limited.

[2] **Perplexity** measures a language model's surprise at a given text: low perplexity indicates a text deemed highly predictable by the model.

[3] **Bigram frequency** refers to the rate of appearance of pairs of consecutive words in a text; low diversity may indicate an artificial style

[4] **Burstiness** measures the local concentration of the same word in a text; human texts often show greater repetition around key concepts

[5] **Textual entropy** evaluates the lexical diversity of a text: low entropy reflects limited and predictable vocabulary use

Adopting this framework offers several advantages:

- It allows the training of flexible and scalable models on annotated datasets.
- It enables standardized and quantitative evaluation of detection performance.
- It facilitates the fair comparison of different approaches under a unified experimental protocol.

Therefore, this project considers the detection task as a binary classification problem, aiming to predict whether a given text is human-written (label 0) or generated by an LLM (label 1).

2.2 Proposed experimental approach

In order to systematically evaluate the ability of different textual representations to distinguish human texts from generated texts, the following experiment is proposed:

2.2.1 Comparison of representation types

Drawing inspiration from the work of Aich and Das [2022], which demonstrated that simple linguistic features can be both interpretable and effective, we propose to compare two broad families of text representations. The first family includes **embedding-based representations**, with vectors automatically learned from deep language models such as BERT, RoBERTa, and Sentence-BERT. The second relies on **manual and interpretable linguistic features**, explicitly capturing aspects like stylistic text composition, lexical complexity, and sentiment scores. The linguistic features used are defined and documented in the appendix (Table 7).

To broaden the comparison, we also include two classical and lightweight text representations: **Bag-of-Words (BoW)**, which encodes texts as raw word count vectors without considering word importance or context, and **TF-IDF (Term Frequency–Inverse Document Frequency)**, which refines this representation by down-weighting common words and emphasizing terms that are more discriminative across documents.

This comparative study aims to answer several key questions: Are deep embeddings necessary to detect AI-generated text? Can simple linguistic signals suffice for accurate detection? What trade-off exists between model performance and interpretability?

2.2.2 Training and evaluation protocol

For each type of feature representation, three supervised classification models are trained: **logistic regression**, **linear support vector machine (SVM)**, and **XGBoost**. These models are selected for their ability to handle both dense and sparse feature spaces.

The experimental procedure follows a standardized protocol. The dataset is split into 80% training and 20% testing. Hyperparameter optimization is conducted via grid search using 3-fold cross-validation, applied solely on the training set. The final model evaluation is performed on the unseen test set.

The performance of the models is evaluated according to several standard binary classification metrics:

- **Accuracy** (overall proportion of correct predictions),
- **F1-score** (harmonic mean of precision and recall),
- **Precision** (proportion of true positives among predicted positives),
- **Recall** (proportion of actual positives correctly predicted),
- **Training time** (computational efficiency)

This choice of metrics allows for a detailed evaluation of both the predictive quality and the time cost of learning.

3 The data

3.1 Source and constitution of the corpus

The dataset used in this project is derived from the work of Wang et al. [2023], made publicly available through their GitHub repository [Wang, 2023]. While their original study introduces a sentence-level detection challenge for AI-generated text and proposes a novel method based on log-probability features, it also provides a rich and structured corpus of human-written and LLM-generated texts, which we leverage for our binary classification task.

The corpus includes two types of textual data:

- **LLM-generated texts** from three major models: GPT-2, GPT-3, and LLaMA.
- **Human-written texts** collected from the original datasets.

In order to ensure a perfect balance between classes, a total of 12,000 examples were selected, consisting of **6,000 human-written texts** and **6,000 LLM-generated texts**. Among the generated texts, **2,000 examples from each model** (GPT-2, GPT-3, and LLaMA) were randomly sampled from the available datasets.

Random sampling was performed to avoid overspecialization of the models towards a specific generator and to simulate a more general detection scenario. Each text is labeled in a binary format, 0 if the text written by a human and 1 if it is generated by an LLM.

This design allows for evaluating the **robustness and generalization abilities** of detection models across multiple LLM families. The objective is to train classifiers capable of distinguishing AI-generated texts, regardless of the specific generator used.

3.2 Descriptive analysis of the dataset

In this section, we provide an overview of the textual dataset used for training and testing the classification models. The goal is to highlight key linguistic characteristics that may distinguish human-written texts from those generated by large language models (LLMs).

In this section, we present a descriptive analysis of the **entire dataset**, combining both training and test sets, to highlight key linguistic characteristics that may help distinguish human-written texts from those generated by large language models (LLMs). A separate analysis comparing the training and test subsets individually is provided in the Appendix 6.2.

3.2.1 Class distribution

The corpus is evenly balanced with a total of 12,000 texts: 6,000 written by humans and 6,000 generated by three different LLMs (GPT-2, GPT-3, and LLaMA). Each machine source contributes equally.

Label	Count	Percentage (%)
Human	6,000	50.00
GPT-2	2,000	16.67
GPT-3	2,000	16.67
LLaMA	2,000	16.67

Table 1: Distribution of text sources in the full dataset.

3.2.2 Text length

We measured the number of words and characters in each text. Human-written texts tend to be longer and more variable than those generated by LLMs.

Label	Mean Words	Median	Std	Min	Max
Human	235.1	184	186.5	1	2521
GPT-2	208.5	217	96.4	2	513
GPT-3	211.5	203	64.4	29	640
LLaMA	178.2	183	81.8	8	511

Table 2: Text length (in words) by label.

3.2.3 Lexical diversity

Lexical richness was assessed using two complementary metrics: the **Type-Token Ratio (TTR)**, which measures the proportion of unique words relative to total words, and the **Measure of Textual Lexical Diversity (MTLD)**, which provides a more stable estimate less sensitive to text length.

Label	TTR (mean)	MTLD (mean)
Human	0.575	1.86
GPT-2	0.556	1.94
GPT-3	0.556	1.86
LLaMA	0.580	1.77

Table 3: Lexical diversity metrics by label.

Although the differences remain relatively small, human and LLaMA-generated texts tend to display slightly higher lexical variety (TTR), while GPT-2 achieves the highest MTLD score. These patterns suggest subtle variations in vocabulary richness across sources.

3.2.4 Punctuation usage

Punctuation features include the total number of punctuation marks, the number of unique punctuation types, and the frequency of exclamation marks. Human texts tend to use more varied and abundant punctuation.

Label	Punctuation Count	Unique Types	Exclamation Frequency
Human	42.8	6.5	0.0013
GPT-2	38.5	6.3	0.0008
GPT-3	29.3	5.0	0.0003
LLaMA	31.1	5.8	0.0017

Table 4: Mean punctuation metrics by text label.

3.2.5 Sentiment score

The sentiment score is computed based on lexical sentiment resources and measures the affective polarity of the text.

Label	Mean Score	Std Dev
Human	0.0043	0.0111
GPT-2	0.0047	0.0115
GPT-3	0.0073	0.0115
LLaMA	0.0052	0.0126

Table 5: Sentiment polarity score by label.

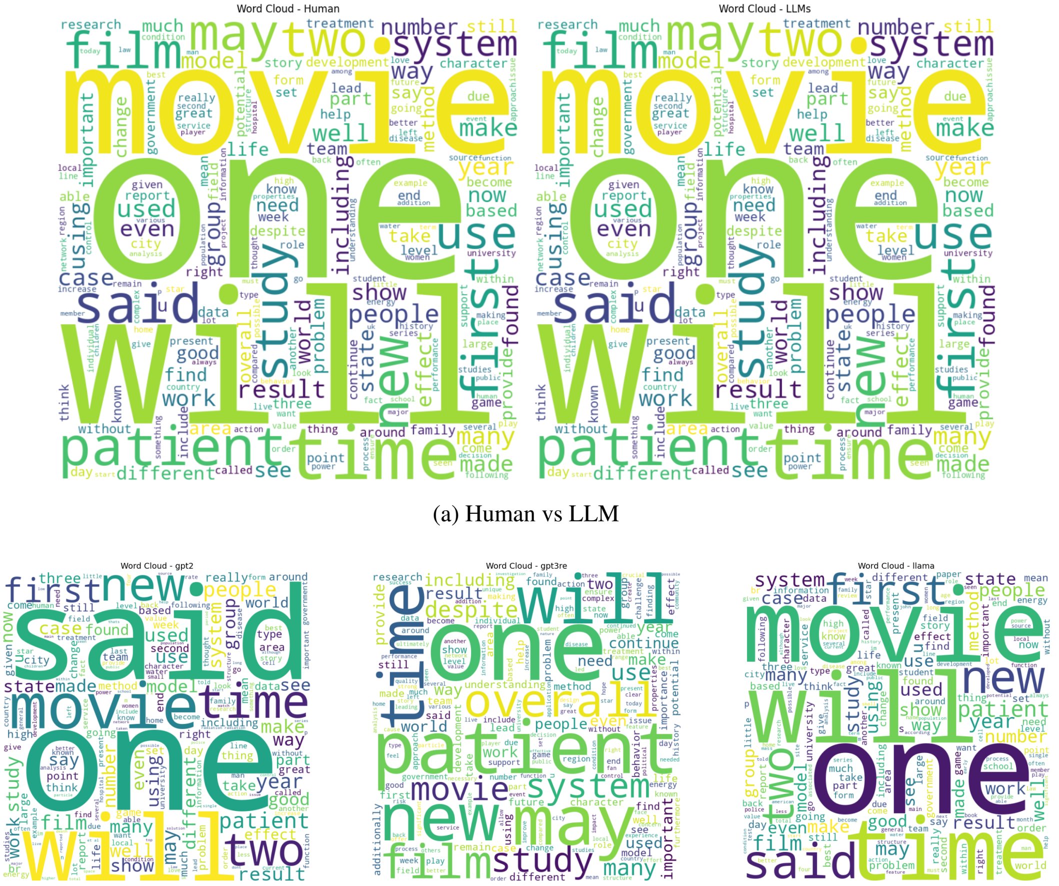
Sentiment analysis reveals that all sources exhibit near-neutral affective polarity on average, with slight variations across models. GPT-3 texts show marginally higher positivity than others.

3.2.6 Word frequency and Word Clouds

A comparison of the most frequent words across labels reveals both shared vocabulary (e.g., “one”, “will”, “said”, “new”) and topical variations (e.g., “patients”, “study” in GPT-3, or “movie” in human and GPT-2 texts). These trends reflect both generic structures and domain-specific biases of LLM generations.

- **Human:** said, one, will, br, two, time, first, new, people, movie
- **GPT-2:** one, said, will, two, first, time, new, movie, people, may
- **GPT-3:** will, new, one, may, study, patients, time, overall, important, many
- **LLaMA:** one, will, two, said, movie, time, first, new, study, film

To visually capture these differences, we generated word clouds for each class as well as for the aggregated LLM-generated texts. In each cloud, the size of a word reflects its relative frequency within the corresponding group, offering an intuitive view of lexical prominence.



(a) Human vs LLM

(b) GPT-2

(c) GPT-3

(d) LLaMA

Figure 1: Word clouds highlighting the most frequent tokens in texts generated by humans and LLMs.

Interestingly, the word clouds for human and LLM-generated texts in the aggregate view appear nearly identical, highlighting the lexical overlap between the two categories and underscoring the challenge of distinguishing them based solely on surface-level word frequency.

3.2.7 Summary

Overall, human texts tend to be longer, lexically richer, and more variable in punctuation use. Sentiment is similar across all classes. Frequent word usage reveals shared vocabulary but also thematic preferences across sources.

4 Results of the experiments

4.1 Benchmark results

Table 6 presents the performance of all trained classifiers across different feature representations, providing a comprehensive benchmark of their accuracy, F1-score, precision, recall, and training time.

Table 6: Benchmark results for different feature types and classifiers.

Feature Type	Classifier	Accuracy (%)	F1-Score (%)	Precision (%)	Recall (%)	Training Time (s)
BERT	Logistic Regression	80.8	80.7	80.8	80.8	7.6
BERT	Linear SVC	81.3	81.2	81.3	81.3	11.5
BERT	XGBoost	78.5	78.5	78.5	78.5	76.7
RoBERTa	Logistic Regression	89.7	89.7	89.7	89.7	14.4
RoBERTa	Linear SVC	89.8	89.7	89.8	89.8	10.3
RoBERTa	XGBoost	86.8	86.8	86.8	86.8	71.4
SBERT	Logistic Regression	60.5	60.5	60.5	60.5	3.9
SBERT	Linear SVC	60.5	60.5	60.5	60.5	6.9
SBERT	XGBoost	58.3	58.3	58.3	58.3	28.5
TF-IDF	Logistic Regression	62.7	62.7	62.7	62.7	1.8
TF-IDF	Linear SVC	62.5	62.5	62.5	62.5	4.4
TF-IDF	XGBoost	65.1	65.1	65.1	65.1	43.3
Bag of Words	Logistic Regression	62.1	62.1	62.1	62.1	2.6
Bag of Words	Linear SVC	62.1	62.1	62.1	62.1	8.6
Bag of Words	XGBoost	69.0	69.0	69.0	69.0	10.0
Handcrafted Features	Logistic Regression	66.6	66.6	66.6	66.6	3.2
Handcrafted Features	Linear SVC	66.8	66.8	66.9	66.8	1.4
Handcrafted Features	XGBoost	76.0	76.0	76.0	76.0	1.7

4.2 Analysis of the results

4.2.1 Overall performance trends

Across all feature types, deep embedding-based representations (BERT, RoBERTa) achieve the highest detection performances. In particular, RoBERTa embeddings consistently outperform all other methods, reaching up to 89.8% accuracy with Linear SVC.

BERT embeddings come second, with an accuracy around 81%, noticeably lower than RoBERTa but still significantly better than classical or handcrafted features.

In contrast, Sentence-BERT embeddings perform poorly (around 60% accuracy), suggesting that this model — optimized for semantic similarity — is less suited for distinguishing between human and LLM-generated text.

4.2.2 Comparison of feature types

- **RoBERTa**: Best results overall ($\sim 90\%$), strong across all classifiers.
- **BERT**: Good performance ($\sim 80\%$), but below RoBERTa.
- **Handcrafted features**: Reasonable performance (up to 76% with XGBoost), showing that linguistic signals still provide valuable detection cues.
- **Bag-of-Words** and **TF-IDF**: Moderate performance ($\sim 62 - 69\%$), better with XGBoost.
- **Sentence-BERT**: Lowest performance ($\sim 58 - 60\%$), indicating poor discriminative capacity in this context.

4.2.3 Classifier impact

Across feature types, **Linear SVC** slightly outperforms Logistic Regression and is usually more stable across different representations. **XGBoost** shows better results when using simpler features (TF-IDF, BoW, handcrafted features), but its advantage is less pronounced for deep embeddings.

Additionally, **training time** varies considerably:

- Logistic Regression and Linear SVC are very fast.
- XGBoost has longer training times, especially with embeddings (up to **76 seconds** with BERT compared to only **7 seconds** for logistic regression on the same representation.).

4.2.4 Trade-offs observed

While **RoBERTa embeddings provide the best performance**, they require extracting deep contextual embeddings, which can be computationally more expensive.

Conversely, **handcrafted linguistic features** offer a good trade-off between performance, interpretability, and training efficiency, with significantly lower computational costs.

This suggests that the choice between deep embeddings and handcrafted features may depend on practical constraints like available computational resources, the need for model interpretability, and the domain of application.

4.3 Summary

These results highlight the crucial role of feature selection in the task of LLM-generated text detection. While deep contextual embeddings, particularly those from RoBERTa, deliver superior performance, they require more computational resources and offer limited interpretability. In contrast, handcrafted linguistic features, despite slightly lower performance, provide a viable and efficient alternative, especially when interpretability and training efficiency are critical. Depending on the application context — whether it prioritizes accuracy, transparency, or scalability — different trade-offs between deep and shallow representations may be preferable.

5 Conclusion

This work addressed the task of detecting texts generated by large language models (LLMs) through a comparative study of different feature representations and classification models. By evaluating deep embeddings (from BERT, RoBERTa, and Sentence-BERT), classical vectorization methods (TF-IDF, Bag of Words), and interpretable handcrafted features, we provided a detailed benchmark across a balanced and multi-source corpus.

RoBERTa embeddings achieved the best performance, followed by BERT. However, handcrafted features proved competitive while offering advantages in terms of interpretability and computational efficiency. Descriptive analysis further highlighted key differences between human-generated and machine-generated text, particularly in terms of length, diversity, and punctuation.

Nevertheless, several limitations remain for this type of detection methods:

- **Vulnerability to paraphrasing and rewriting:** Small lexical or syntactic changes can significantly reduce detection performance.
- **Domain generalization:** Models trained on specific topics or styles may struggle to generalize to out-of-distribution texts or different domains.
- **Transparency of LLMs:** When generated texts are obtained from unknown or evolving LLMs, it becomes increasingly difficult to detect them reliably.

To overcome current limitations, future improvements could include combining interpretable artisanal features with deep integrations to balance performance and explainability. Ensemble methods that aggregate predictions from multiple classifiers, each trained on different types of representations (e.g., RoBERTa, TF-IDF, or linguistic signal integrations), could improve robustness against adversarial editing. Finally, integrating contextual metadata (such as the source domain or task type associated with the text) could help detection models better generalize their results across applications and writing styles.

References

Jyotika Aich and Dipankar Das. A linguistic approach towards detecting gan-generated text. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6552–6563, 2022.

Pengyu Wang. Seqxgpt: Project github repository. `https://github.com/Jihuai-wpy/SeqXGPT`, 2023. GitHub repository. Accessed: 2024-04-25.

Pengyu Wang, Wayne Xin Zhao, Qingyao Ai, and Ji-Rong Wen. Seqxgpt: Sentence-level ai-generated text detection. *arXiv preprint arXiv:2310.08903*, 2023. URL `https://arxiv.org/abs/2310.08903`.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F. Wong, and Lidia S. Chao. A survey on llm-gernerated text detection: Necessity, methods, and future directions. *CoRR*, abs/2310.14724, 2023. URL `https://arxiv.org/abs/2310.14724`.

6 Appendix

6.1 Appendix 1

The handcrafted features used in our detection model are inspired by the methodology of Aich and Das [2022]. Designed to preserve interpretability, these features capture a variety of linguistic signals that may help distinguish human-written text from content generated by large language models (LLMs). They are grouped into three complementary categories: stylistic markers, indicators of linguistic complexity, and affective (psychological) cues.

Category	Variable Name	Description
Stylistic	quote_frequency (QF)	Frequency of quotation marks (e.g., ", ’, «, »).
	punctuation_count (PF)	Frequency of standard punctuation marks such as ., ;, :, !, ?.
	unique_punctuation_count (PT)	Number of distinct punctuation types used.
	exclamation_frequency (EF)	Frequency of exclamation marks (!).
	stopword_frequency (SWF)	Frequency of stopwords based on the NLTK list.
	camel_case_frequency (CCF)	Frequency of words starting with a capital letter followed by lowercase letters (e.g., CamelCase).
	negation_frequency (NF)	Frequency of negation words such as "not", "no", "never", etc.
	proper_noun_frequency (NPF)	Frequency of proper nouns (POS tags NNP and NNPS).
	user_mentions_frequency	Frequency of the @ symbol (user mentions).
	hashtag_frequency	Frequency of hashtags (#).
	misspelled_words	Frequency of words considered invalid by PyEnchant.
	oov_frequency	Frequency of words not found in SentiWordNet.
	noun_frequency	Frequency of nouns (POS tags NN, NNS, NNP, NNPS).
	past_tense_frequency	Frequency of past tense verbs (POS tags VBD, VBN).
	verb_frequency	Frequency of all verbs (POS tags VB, VBP, VBZ, VBD, VBG).
	interrogative_frequency	Frequency of interrogative words (POS tags WRB, WDT, WP).
Complexity	word_count	Total number of words in the text.
	mean_word_lenght	Average number of characters per word.
	ttr	Type-Token Ratio: ratio of unique words to total words.
	mtld	Measure of lexical diversity (TTR over growing text segments).
Psychological	sentiment_score	Sum of SentiWordNet scores for all valid vocabulary words in the text.

Table 7: Handcrafted linguistic features used in the classification model.

6.2 Appendix 2

To ensure consistency and fairness in the evaluation, we analyzed the distribution of linguistic characteristics between the training and test sets.

6.2.1 Label distribution

Label	Train Count	Train %	Test Count	Test %
Human	4790	49.90	1210	50.42
GPT-2	1595	16.61	405	16.88
GPT-3	1614	16.81	386	16.08
LLaMA	1601	16.68	399	16.62

Table 8: Class distribution in training and test sets.

6.2.2 Feature consistency

The means of the main handcrafted features are very close between the training and test datasets. For instance:

- **Word count:** 289.2 (train) vs 296.5 (test)
- **TTR:** 0.570 (train) vs 0.567 (test)
- **MTLD:** 1.85 (train) vs 1.87 (test)
- **Sentiment score:** 0.0050 (train) vs 0.0052 (test)
- **Punctuation count:** 37.7 (train) vs 38.7 (test)

Standard deviations and ranges are also similar, suggesting no major distribution shift. This indicates that the model is trained and evaluated under comparable data conditions, reinforcing the validity of generalization performance.

Further details and full descriptive statistics are available in the notebook in the Github repository `https://github.com/AminaManseur29/ENSAE_NLP_project.git`.