LLM generated text detection

Mello DON
ENSAE student
mello.don@polytechnique.edu

Abstract

Large language models (LLMs), such as ChatGPT, are now capable of generating text with quality comparable to that of humans, posing major challenges for automatic detection. This work focuses on the binary classification task of distinguishing human-written answers from those generated by an LLM, within a question-answering framework. Relying on the HC3 dataset (1), we explore several families of models: logistic regression on TF-IDF representations, the addition of heuristic features (length, number of sentences, lexical diversity), and embeddings from pre-trained models such as BERT. Our experiments show that a simple combination of TF-IDF and heuristic features can achieve up to **98%** accuracy, outperforming human-level performance reported in the literature. We also discuss the limitations of this approach, particularly in the face of prompt tuning or concatenation of multiple generations, and propose directions for more robust generalization to newer generative models.

1 Introduction

Large Language Models (LLMs), such as GPT-4, Claude, or Mistral, have revolutionized automatic text generation. Their ability to produce coherent, informative, and stylistically diverse content makes them increasingly difficult to distinguish from texts written by humans. While these advances open up promising opportunities in various domains (education, automation, writing assistance), they also raise significant challenges in terms of traceability, authenticity, and ethics.

In this context, the automatic detection of LLM-generated text has become a crucial issue. It addresses concrete needs: verifying the authenticity of academic work, identifying automated disinformation campaigns, or detecting text generated for fraudulent purposes.

This task can be framed as a binary classification problem: given a text, determine whether it was written by a human or generated by an LLM. While this question may seem straightforward, it presents numerous methodological and practical challenges, particularly due to the diversity of generative models, the complexity of natural languages, and evasion strategies designed to avoid detection.

In this project, we focus specifically on the detection of LLM-generated text in English. We rely on recent benchmarks from the literature, as well as a review of existing approaches (traditional, neural, and zero-shot methods). Our goal is to assess the relevance of these techniques in a multilingual setting, and to propose a model adapted to French-language texts.

We structure this work as follows: the next section introduces the datasets and exploratory analysis; we then review the main detection methods; we present our own detection model; we detail its implementation; finally, we discuss the results and outline directions for future research

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

2 Related Work

The detection of text generated by language models has rapidly emerged as a research field in its own right, following the rise of LLMs. Existing methods can be broadly categorized into three families: traditional statistical methods, supervised neural approaches, and unsupervised methods based on heuristics or perplexity metrics.

2.1 Traditional Methods

Early attempts at detection relied on simple statistical models, often based on n-gram representations or stylistic features (average sentence length, lexical diversity, etc.). Linear classifiers such as logistic regression or SVMs were successfully applied to small datasets, especially for detecting texts generated by GPT-2 or GPT-3 when they first appeared.

However, these approaches quickly show their limitations when confronted with longer, more diverse texts, or outputs produced by more powerful and better-calibrated models.

2.2 Neural Approaches

With the advent of models like BERT and RoBERTa, fine-tuned classifiers trained on corpora of human and machine-generated texts have achieved significantly better performance. These approaches benefit from the rich contextual representations extracted by transformer architectures.

More recently, dedicated detection models such as `DetectGPT`, the `OpenAI Text Classifier`, and `GPTZero` have been proposed. These systems rely either on fine-tuned models or on likelihood scores computed by LLMs without fine-tuning.

One recurring challenge of such approaches is generalization: a model trained to detect GPT-2 content often struggles to identify text from GPT-4 or Claude, especially when the prompt distribution differs.

2.3 Unsupervised and Heuristic Methods

Unsupervised detection strategies have also been explored. The most well-known relies on the notion of perplexity: a measure of how unpredictable a text is to a given language model. Formally, perplexity quantifies how "surprised" a model is by the words it reads; a text composed of highly likely words for the model will have low perplexity, whereas more unusual wording will lead to higher perplexity. In the context of generation detection, the idea is that if a text is evaluated using the same LLM that may have produced it, its perplexity is generally lower than that of a human-written text, which often contains more diverse and less predictable lexical choices.

The `GLTR` project (Giant Language Model Test Room) illustrates this approach by visualizing, token by token, the probability distribution predicted by a language model. This allows visual detection of overly predictable text, a common characteristic of LLM-generated content: a systematic use of highly expected words with little surprising variation.

Finally, some methods aim to introduce "watermarks" directly during generation—hidden signatures embedded into the text. While promising, such methods remain vulnerable to paraphrasing or machine translation.

2.4 Current Challenges

The most pressing challenges today include:

- **Robustness to diverse models and languages**: most detection systems are trained in English and exhibit significant performance drops in French or other languages.
- **Detection under paraphrasing or prompt tuning**: users can often bypass detectors with simple rewording strategies.
- **Fairness and bias**: some detectors exhibit bias toward certain writing styles or even specific social groups.
- **Transparency and accessibility**: many high-performing detectors are proprietary (e.g., OpenAI's classifier), making auditing and evaluation difficult.

In the remainder of this work, we focus primarily on detecting the use of LLMs in question-answering tasks. Our goal is to empirically evaluate various detection methods on English-language data, with an emphasis on their ability to distinguish between human-written and machine-generated responses.

3 Cadre expérimental

3.1 Description du corpus

To conduct our experiments, we use the HC3 corpus (*Human ChatGPT Comparison Corpus*). This dataset was specifically designed for studying the detection of text generated by language models, by comparing answers produced by ChatGPT with authentic human responses in a question-answering setting.

The corpus spans several distinct thematic categories:

- **reddit_eli5**: question-answer pairs from the r/explainlikeimfive subreddit, where users explain complex concepts in simple terms;
- **open_qa**: open-domain question answering on a wide range of general topics;
- **wiki_csai**: questions and answers from specialized wikis on artificial intelligence and computer science;
- **finance**: economic and financial topics from expert forums and financial knowledge bases;
- **medicine**: medical QA pairs from healthcare-related resources.

Human answers were collected from credible platforms, with the guarantee that all content predates the public release of ChatGPT (November 2022), ensuring the authenticity of the human-written responses compared to generative model outputs. While some answers may contain factual inaccuracies or errors, this does not impact our main objective: we are interested purely in stylistic differentiation between human and machine-generated answers, regardless of factual correctness.

The corpus contains over 25,000 question-answer pairs, each including:

- a **question**,
- a **human-written answer**,
- an **answer generated by ChatGPT**.

Each response is labeled with a binary annotation (human or ChatGPT), making this dataset particularly suitable for supervised classification.

The generated responses were obtained via the official ChatGPT API interface (based on the **GPT-3.5** series), in English—consistent with all other questions and responses in the corpus.

To better understand the dataset's composition, we visualize the distribution of examples across the thematic categories. This visualization is presented in Figure 1.

As shown in Figure 1, the question-answer pairs are unevenly distributed across topics, with particularly strong representation in the reddit_eli5 and open_qa categories. This imbalance is an important consideration for the experimental analysis, as thematic skew may introduce potential bias in the classification task. For example, certain domains like medicine exhibit highly specific writing styles, which justifies their partial removal from our later experiments in order to avoid excessive bias.

During our exploratory analysis of the HC3 corpus, we observed that human answers in the medicine category consistently exhibit a marked stylistic bias: **each human answer begins with a greeting**, typically something like *"Hello"* or similar. This highly regular stylistic pattern makes classification artificially easier by providing an obvious signal to distinguish human text from that generated by ChatGPT.

We illustrate this phenomenon with the following real example from the corpus:

- **Question**: Noticed a yellowish sag in the gums of my 13 months old baby. What is it? Hi doctor, My daughter is 13 months old. I just noticed big yellowish sag at the place of her first molar gum. I am much worried. Please help me.

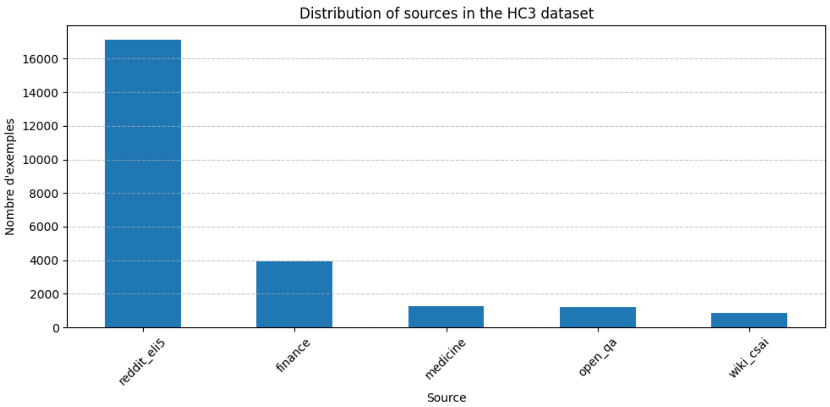

Figure 1: Distribution of examples in the HC3 dataset by thematic source.

- **Human answer**: Hello. Revert back with the photos to a dentist online → `https://www.icliniq.com/ask-a-doctor-online/dentist`
- **ChatGPT answer**: It is difficult to accurately diagnose a condition without examining the affected area in person. However, it is possible that the yellowish sag in your daughter's gum could be a gum boil, also known as a dental abscess. [...] I recommend that you do not try to treat the condition yourself, as it is important to receive a proper diagnosis and treatment from a healthcare professional.

This example shows that the human-written response begins with a clear greeting (*"Hello"*), which is absent from the ChatGPT-generated response. Such a pattern can be easily exploited by a classifier without requiring any deep semantic understanding of the text. To ensure a fairer evaluation and to avoid introducing an artificial bias in favor of automatic classification, we therefore choose to **exclude the `medicine` category** from our experiments going forward.

3.2 Dataset Analysis and Feature Extraction

Before training our detection models, we perform a statistical analysis of the dataset and extract relevant linguistic features.

Descriptive Statistics. We begin by comparing the length of human-written and machine-generated responses, measured in number of words. Table 1 presents the main descriptive statistics.

	Human	**ChatGPT**
Number of examples	22,619	22,619
Mean	151.61	177.49
Standard deviation	180.34	58.41
Minimum	2	1
25th percentile	48	138
Median (50th percentile)	97	177
75th percentile	190	214
Maximum	7904	639

Table 1: Descriptive statistics of response lengths (word count).

We observe that responses generated by ChatGPT are, on average, longer than human responses (177 vs. 151 words), and their lengths are much more standardized (with a significantly lower standard deviation). Human-written answers exhibit much greater variability, as reflected in the wide range between the shortest and longest observed responses.

These differences are illustrated in the two plots shown side by side in Figure 2.

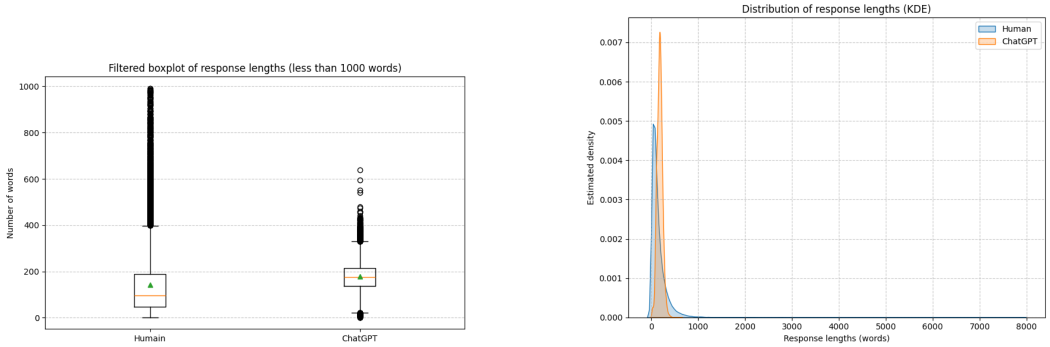

Figure 2: Left: boxplot of human and ChatGPT response lengths. Right: kernel density estimation (KDE) of response length distributions.

The boxplot shows that human responses include more extreme values (rare long answers), while ChatGPT responses are more concentrated around their median. The KDE plot confirms this observation: the distribution of human response lengths is more spread out with a longer tail, whereas ChatGPT responses are more tightly clustered and symmetric.

These findings suggest that response length—measured both in word count and sentence count—can serve as a useful feature for the classification task. ChatGPT responses tend to be more standardized in length, with less variability compared to human responses, which exhibit more heterogeneous structures. Thus, length can provide a discriminative signal, particularly for detecting overly calibrated or uniformly structured text typical of generated content.

Lexical Analysis. We begin our lexical analysis by visualizing the most frequent words in human and ChatGPT responses. Figure 3 presents the corresponding word clouds.

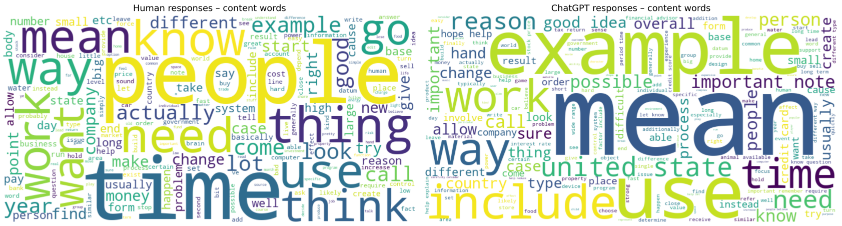

Figure 3: Comparison of word clouds for human responses (left) and ChatGPT responses (right).

A preliminary qualitative inspection reveals notable stylistic and register differences between the two types of responses. Human-written texts tend to emphasize experiential or personal terms such as *people*, *time*, *thing*, *money*, *want*, *go*, and *think*. By contrast, ChatGPT responses employ more formal or technical vocabulary, including terms like *help*, *important*, *include*, *example*, *financial*, *body*, and *stock*.

Some words (*people*, *use*, *way*, *time*, *different*) appear in both categories, but their relative frequency and contextual usage differ. This visualization highlights significant stylistic distinctions that can be leveraged for supervised classification.

To quantify the lexical richness observed in the word clouds, we compute the *Type-Token Ratio* (TTR) for both text categories. The TTR is defined as follows:

$$\text{TTR} = \frac{\text{number of types (unique words)}}{\text{total number of tokens}}.$$

It is a standard measure of lexical diversity: a high TTR indicates more varied vocabulary, while a low TTR reflects greater lexical redundancy.

In our dataset, we obtain the following values:

- **Average TTR for human responses**: ≈ 0.03
- **Average TTR for ChatGPT responses**: ≈ 0.01

These results confirm that human responses demonstrate greater lexical variety, with more synonyms, rephrasing, and stylistic heterogeneity. Conversely, ChatGPT responses are more repetitive, with a more uniform style, recurring phrasing, and frequent use of certain key terms.

TTR thus proves to be a relevant discriminative feature for distinguishing human-written text from language-model-generated responses.

3.3 Modeling and Classification

We build text detection models using ℓ_1-regularized logistic regression (Lasso logistic regression).

Logistic regression models the probability that a given response belongs to the `ChatGPT` class, based on a feature vector $x \in \mathbb{R}^d$, by learning a weight vector w and a bias term b:

$$P(y = 1 \mid x) = \sigma(w^\top x + b), \quad \text{with} \quad \sigma(z) = \frac{1}{1 + e^{-z}}.$$

In our case, we add an ℓ_1 penalty on the coefficients to encourage automatic feature selection:

$$\min_{w,b} \left(\mathcal{L}(w, b) + \lambda \|w\|_1 \right),$$

where $\mathcal{L}(w, b)$ is the logistic loss and λ controls the strength of regularization.

Lasso regularization is particularly well-suited to our task for three reasons:

- It automatically selects the most discriminative features;
- It enhances interpretability by highlighting key indicators;
- It reduces the risk of overfitting in high-dimensional feature spaces (e.g., TF-IDF).

We evaluate four experimental variants:

- Using only heuristic features (length, number of sentences, TTR);
- Using only a TF-IDF vectorization of the text;
- Combining heuristic features with TF-IDF representations;
- Combining heuristic features with BERT embeddings.

Logistic Regression on Heuristic Features

In our first approach, we use only simple structural features:

- Text length (number of words)
- Number of sentences
- Type-Token Ratio (TTR)

These features are standardized before model training. The model is trained on 80% of the data and evaluated on the remaining 20%. We compute accuracy, precision, recall, F1-score, and ROC AUC.

Results: The logistic regression model trained on heuristic features alone achieves an overall accuracy of **76%** on the test set. The classification metrics are summarized below:

Class	**Precision**	**Recall**	**F1-score**
Human	0.77	0.72	0.75
ChatGPT	0.74	0.79	0.76
Macro average	0.76	0.76	0.76

The performance is relatively balanced across the two classes. The model slightly better identifies ChatGPT responses (*recall of 79%*) than human-written ones (*recall of 72%*). Conversely, precision is slightly higher for human responses (77%) than for ChatGPT (74%).

These results show that simple structural features—text length, number of sentences, and lexical diversity (TTR)—capture useful signals for detection. However, they are not sufficient on their own to clearly separate the two text types. In particular, the syntactic and stylistic similarities between well-written human responses and some ChatGPT generations make the classification task more difficult.

This motivates the addition of richer lexical representations via TF-IDF vectorization, in combination with these heuristic descriptors.

Logistic Regression on TF-IDF

To enrich the textual representation beyond heuristic features alone, we use **TF-IDF** (*Term Frequency–Inverse Document Frequency*) vectorization, a classic method in natural language processing for converting raw text into numerical vectors.

TF-IDF weighting is based on two components:

- **TF (Term Frequency)**: frequency of a term in a given document;
- **IDF (Inverse Document Frequency)**: inverse of the term's frequency across the corpus.

The general formula is:

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \log\left(\frac{N}{1+n_t}\right),$$

where:

- t is the term,
- d is the document,
- N is the total number of documents,
- n_t is the number of documents containing term t.

This weighting favors words that are frequent in a document but rare across the corpus, highlighting potentially informative terms.

In our experiment, we apply TF-IDF vectorization on **unigrams and bigrams**, with a vocabulary limited to the top 5000 most frequent tokens. The resulting vectors are used as input to a Lasso logistic regression model (ℓ_1-penalized), which performs both classification and automatic feature selection.

Results. The trained model achieves a remarkable accuracy of **93%** on the test set. Detailed scores are shown below:

Class	**Precision**	**Recall**	**F1-score**
Human	0.94	0.93	0.93
ChatGPT	0.93	0.94	0.93
Macro average	0.93	0.93	0.93

These results mark a significant improvement over the performance of the heuristic-only model. They show that the model successfully captures lexical and syntactic patterns characteristic of each text class, enabling reliable distinction between human and generated content. The symmetry of the metrics confirms the model's robustness and balance.

Analysis of Discriminative N-grams. Figures 4 and 4 display the n-grams with the highest weights assigned by the model, favoring respectively human-written (negative weights) and ChatGPT-generated (positive weights) texts.

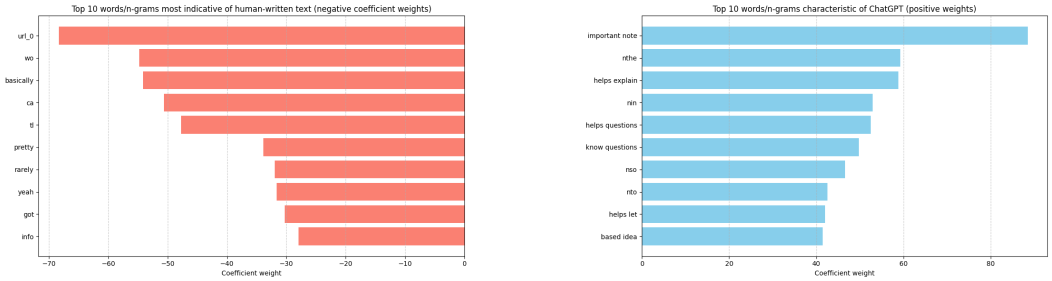

Figure 4: Top 10 most discriminative n-grams: human (left), ChatGPT (right).

Human answers are characterized by informal, spontaneous expressions: `basically`, `yeah`, `pretty`, `got`, as well as fragments like `ca` and `wo`, which reflect casual spoken language. One of the most discriminative n-grams is `url_0`, a token used during preprocessing to replace hyperlinks. Its high frequency in human responses reflects common practices on forums like `r/explainlikeimfive`, where users often include external links. In contrast, texts generated by the `GPT-3.5` series via the `ChatGPT` interface typically do not include hyperlinks, making this feature highly discriminative.

On the ChatGPT side, we observe more formal, instructional, or neutral constructions: `important note`, `helps explain`, `based idea`, `to note`. These phrases are indicative of a guided, explanatory style, consistent with that of a conversational assistant.

Some n-grams such as `nthe`, `nin`, `nto`, and `nso` are also among the most discriminative for ChatGPT. These are likely artifacts from tokenization or automatic bigram segmentation (e.g., `in the`, `on to`), which are more frequent in generated texts due to their more regular and structured nature. While these fragments carry no intrinsic meaning, they can still be exploited statistically by the model.

This analysis shows that stylistic, lexical, and structural differences between human and generated answers can be effectively captured using a simple model trained on TF-IDF representations.

Logistic Regression on TF-IDF + Heuristic Features

In our final experiment, we combine TF-IDF representations with the three previously defined heuristic features: response length, average number of sentences, and type-token ratio (TTR). The goal is to leverage both fine-grained lexical patterns and global structural signals.

The heuristic features are standardized and concatenated with the TF-IDF vectors, forming a single extended feature vector for each response. The model used is again an ℓ_1-regularized logistic regression (Lasso). Cross-validation was used to select the regularization hyperparameter C.

Results. The best model was obtained with $C = 10$. Performance on the test set is as follows:

Class	**Precision**	**Recall**	**F1-score**
Human	0.98	0.98	0.98
ChatGPT	0.98	0.98	0.98
Macro average	0.98	0.98	0.98

The addition of heuristic features to TF-IDF representations yields a notable performance gain compared to the TF-IDF-only model. The overall accuracy reaches **98%**, with perfectly balanced precision, recall, and F1-score across both classes. This confirms that structural features—though weak when used in isolation—provide useful complementary signals when combined with detailed lexical representations.

Analysis. The resulting model leverages characteristic n-grams found in generated texts (instructive phrases, syntactic regularity), while also incorporating global signals such as excessive length or low lexical diversity typical of LLM-generated responses. Conversely, it detects contractions, typos, hyperlinks (`url_0`), and other markers often present in human-written content on forums.

These results demonstrate that combining classical text representations (TF-IDF) with simple textual statistics can achieve very high performance without requiring heavy neural architectures or transformer fine-tuning. The proposed framework is effective, interpretable, and easily adaptable to other detection contexts.

Coefficient Analysis. A closer examination of the learned model coefficients reveals a notable difference in scale between those associated with TF-IDF n-grams (often exceeding ±30, up to ±60) and those linked to heuristic features (ranging from -0.7 to -2.8 in our model). This does not imply that heuristic features are uninformative—it is rather a consequence of how the representations are constructed.

TF-IDF vectors are high-dimensional (5000 features in our case), with each dimension corresponding to a relatively rare unigram or bigram, typically binary or with low non-zero values. In contrast, heuristic features (length, sentence count, TTR) are continuous, standardized (zero mean, unit variance), and document-level. Their variance is controlled, and their influence spans all examples.

In practice, a large TF-IDF coefficient may only affect a few documents (if the n-gram is rare), while a modest heuristic coefficient applies across all examples. The difference in magnitude therefore reflects both the activation frequency of the features and their local vs. global discriminative roles.

In other words, heuristic features act as stabilizers and complements, while TF-IDF n-grams naturally dominate the coefficient space due to their number and granularity.

This confirms the value of combining both types of features: n-grams capture local lexical subtleties, while heuristics provide global, continuous signals that are robust to lexical variation.

Logistic Regression on BERT + Heuristic Features

In this experiment, we replace TF-IDF representations with dense embeddings from a pre-trained BERT encoder (`all-MiniLM-L6-v2`). Each response is thus represented as a dense vector encoding deep lexical and semantic relationships. These representations are then enriched with the three previously used heuristic features (response length, average sentence count, and TTR), which are standardized and concatenated to the BERT embeddings.

We use an ℓ_1-regularized logistic regression model (Lasso), with the regularization parameter C selected via cross-validation. The best model is obtained for $C = 1000$.

Results.

Class	**Precision**	**Recall**	**F1-score**
Human (0)	0.89	0.88	0.89
ChatGPT (1)	0.88	0.89	0.89
Macro average	0.89	0.89	0.89

The model achieves an overall accuracy of **89%**, with nearly perfect balance between the two classes. These results are slightly lower than those obtained with TF-IDF + heuristic features (93–98%), which may be due to several factors:

- BERT embeddings are dense and rich, but the classifier remains linear. The full potential of contextual embeddings is often better exploited through non-linear models (e.g., MLPs, random forests, or full fine-tuning).
- Lexical artifacts that were highly discriminative in the TF-IDF setting (e.g., `url_0`, `nthe`, etc.) are diluted within the dense embedding space, making them harder to detect for a linear model.
- BERT dimensions are less interpretable, which can hinder effective feature selection in a Lasso framework.
- Finally, BERT representations focus on capturing the overall semantic meaning of a text, which may be less useful in a setting where both human and generated responses are semantically similar (since they answer the same question). Despite stylistic differences, the underlying semantic content is often close, which reduces the margin that can be exploited

by a semantic encoder. In contrast, models leveraging lexical or structural signals (TF-IDF, heuristics) can capture more superficial but critical differences for this classification task.

Conclusion. The use of BERT improves generalization to paraphrased or more complex cases, but in our use case—distinguishing between highly structured human and GPT responses—the TF-IDF representation combined with simple heuristics remains superior, at least when using linear models.

This comparison highlights that well-designed, domain-adapted classical methods can outperform more modern approaches in specific contexts.

4 Conclusion and Future Work

In this study, we investigated the task of automatically detecting text generated by language models, within a question-answering setting in English, using the HC3 dataset. We compared several classical classification approaches based on TF-IDF representations, simple heuristic features (length, sentence count, lexical diversity), and contextual embeddings from BERT.

Our results show that a simple Lasso logistic regression model trained on a combination of TF-IDF and structural statistics can achieve an impressive accuracy of **98%** on the binary classification task between human and ChatGPT-generated responses. This performance clearly surpasses that of human expert evaluators, who reached an accuracy of about **90%** under the pairwise evaluation protocol described in the original paper. This confirms that, in a controlled setting with appropriate data preprocessing, simple lexical signals are sufficient to effectively distinguish between human and LLM-generated content.

However, our approach has several important limitations. First, it relies on lexical or stylistic artifacts present in ChatGPT-3.5 responses, such as the absence of links or syntactic regularity, which can be easily altered through reformulation strategies (paraphrasing) or prompt tuning. Second, it does not account for realistic scenarios where a response may be composed from multiple generations, making the boundary between human and machine-written text much fuzzier. Finally, our model is trained in a supervised setting that depends on a clearly defined binary class balance, which limits its robustness in open-ended scenarios.

Future Work. Several directions for improvement can be considered:

- **Assess robustness to prompt tuning**: Modifying the instructions given to the LLM can result in less predictable responses that more closely mimic human style. Studying the impact of such changes on detector performance is crucial.
- **Test on newer models**: The HC3 dataset is based on ChatGPT-3.5; it would be relevant to evaluate our methods on responses from more advanced models such as **GPT-4** or **GPT-4o**, to determine whether the same signals remain effective.
- **Handle multi-generation responses**: In some applications, a user may combine multiple LLM responses into a single answer. This introduces greater stylistic variability and makes detection harder for methods that rely solely on local lexical patterns.
- **Explore unsupervised or hybrid approaches**: Methods based on perplexity or *watermarks* may offer complementary alternatives to supervised classification.

In conclusion, while our models outperform human performance in this specific and constrained setting, reliable and generalizable detection of LLM-generated text remains an open challenge—at the intersection of linguistic analysis, algorithmic security, and ethical AI deployment.

References

[1] Guo, Iyang, Zhang, Xin, Wang, Ziyuan, Jiang, Minqi, Nie, Jinran, Ding, Yuxuan, Yue, Jianwei, & Wu, Yupeng. *Can ChatGPT outperform humans in detecting AI-generated text?* arXiv preprint arXiv:2301.07597 (2023). `https://arxiv.org/abs/2301.07597`

[2] Wu, Junchao, Yang, Shu, Zhan, Runzhe, Yuan, Yulin, Wong, Derek Fai, & Chao, Lidia S. *A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions*. arXiv preprint arXiv:2310.14724 (2023). `https://arxiv.org/abs/2310.14724`

[3] Su, Yongye, & Wu, Yuqing. *Robust Detection of LLM-Generated Text: A Comparative Analysis*. arXiv preprint arXiv:2411.06248 (2024). `https://arxiv.org/abs/2411.06248`