# Do Large Language Models Leave Fingerprints? Some Experiments in LLM Text Detection

Alex Amari

December 2024

## Introduction

Different Large Language Models (LLMs) exhibit distinct "personalities" and generation patterns, even when given identical prompts. These differences stem from differences in model training data, architecture, size, system prompts, and hyper parameters like temperature in softmax. This project investigates whether these differences can be leveraged to determine the model "lineage" (source model) of LLM-generated text through classification tasks. Specifically, I sought to address 3 questions in a very limited LLMs experiment:

1. Can LLMs detect their own text?
2. Can standard classifiers do better?
3. How much text do we need to identify model lineages?

This report walks through some brief experiments to address these topics. It concludes with a brief discussion of my results, links to all the code, data, model checkpoints, and a basic front-end Hugging Face interface that attempts to distinguish Claude, ChatGPT, and human texts under narrow conditions.

## Base Dataset and Data Generation

I first sought to build a dataset of LLM texts that could be paired with human text for classification and analysis. Therefore the project required a baseline dataset of human-written text that would serve as a foundation for LLM replication. The ideal dataset would contain English texts of a few hundred tokens, covering diverse subjects while maintaining certain stylistic consistencies. Moreover, they would need to be processable by the LLMs, for example by not containing any threatening or racist remarks. These texts would then be used to generate "synthetic twins" - the same content rewritten in each LLM's distinctive style. Finally, I wanted to ensure that no LLM-generated text would contaminate the human dataset, so I specifically searched for texts from prior to 2022.

The IMDB movie review dataset, made available by Stanford AI Laboratory in 2011, met these criteria well. The reviews cover thousands of different films from around the world, all written in English. The reviews were often highly polarized, but never crossed the boundary into threatening or "NSFW" content. From the original 25,000 reviews, I randomly selected 2,000 reviews between 125-275 tokens, covering roughly the middle 3 quartiles of the length distribution, to ensure consistency in length and scope.

Using the OpenAI and Anthropic APIs, I generated 6,000 synthetic texts (2,000 each) using three distinct models:

1. GPT-4o-mini
2. GPT-3.5-turbo
3. Claude-3.5-sonnet

These models were chosen strategically to represent: 1) Different training architectures and methods (Anthropic vs OpenAI) 2) Different versions within the same model family (GPT-3.5 vs GPT-4) and 3) Models in widespread public use with reliable API access. Each model received the following prompt:

> "You will receive a text excerpt. Please write a new passage of similar length that covers the same key points and maintains a similar style, but using your own words. Write a new version that:
>
> - Maintains similar length
> - Covers the same main ideas
> - Uses similar tone/style
> - Feels natural and coherent
>
> Return only your new version."

Generations for all three models completed in approximately 6 hours. The OpenAI API completed all generations on the first pass. The Anthropic API occasionally returned rate overflow errors, in spite of a short implicit sleep timer after each generation. A second pass over the dataset completed the few missing texts from the Anthropic API. The final dataset comprises 8,000 texts: 2,000 human originals and 2,000 generated versions from each of the three models for a total of 6,000 AI texts.

The length distributions across all sources remained broadly consistent, suggesting a successful adherence to the length maintenance instruction. However, all LLMs produced synthetic reviews that were slightly shorter than their human counterparts on average.

To eliminate potential classification biases, preprocessing on human and generated texts consisted only of standardizing ascii encoding across all texts. No other changes were made to the human or AI texts.

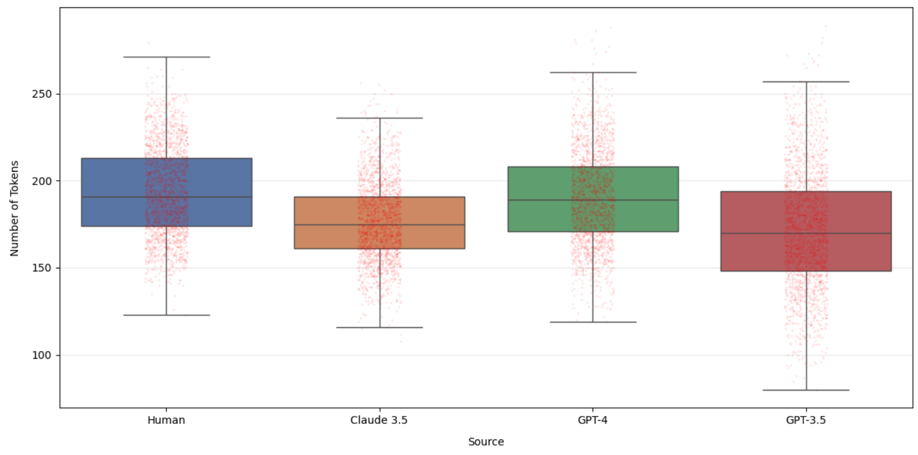
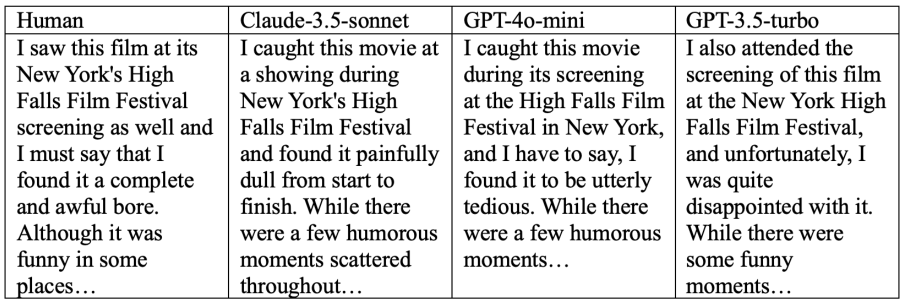


Figure 1: Length Distributions of Human and Generated Texts



| Human | Claude-3.5-sonnet | GPT-4o-mini | GPT-3.5-turbo |
|---|---|---|---|
| I saw this film at its New York's High Falls Film Festival screening as well and I must say that I found it a complete and awful bore. Although it was funny in some places… | I caught this movie at a showing during New York's High Falls Film Festival and found it painfully dull from start to finish. While there were a few humorous moments scattered throughout… | I caught this movie during its screening at the High Falls Film Festival in New York, and I have to say, I found it to be utterly tedious. While there were a few humorous moments… | I also attended the screening of this film at the New York High Falls Film Festival, and unfortunately, I was quite disappointed with it. While there were some funny moments… |

Figure 2: Sample Reviews for Movie Review 196

| Human | | Claude | | GPT-4 | | GPT-3.5 | |
|---|---|---|---|---|---|---|---|
| Word | Freq | Word | Freq | Word | Freq | Word | Freq |
| movie | 3326 | film | 2568 | film | 3617 | film | 2850 |
| film | 2384 | like | 1052 | one | 1107 | movie | 2536 |
| one | 1507 | one | 1037 | like | 933 | like | 1009 |
| like | 1294 | movie | 936 | movie | 808 | one | 894 |
| good | 1003 | even | 904 | even | 701 | even | 751 |
| would | 807 | though | 784 | would | 670 | characters | 588 |
| even | 789 | character | 560 | yet | 646 | would | 537 |
| really | 772 | particularly | 551 | films | 586 | character | 528 |
| see | 752 | story | 547 | might | 568 | however | 524 |
| time | 746 | could | 534 | could | 565 | despite | 523 |

Table 1: Most Frequent Words by Source (Excluding Stop Words)

## Some brief exploration of the texts

To better understand potential differences between human and AI-generated movie reviews, I conducted a basic lexical analysis. After removing stopwords, I examined token frequencies across all four sources (Table 1) and performed chi-square tests of homogeneity on selected key words (Figure 3). While many patterns emerged, three stood out as particularly interesting.

First, the most common content words showed distinct preferences across sources. GPT-3.5 used "movie" substantially more frequently than other sources (2,536 occurrences vs. 1,000 for others), while all sources heavily used "film." This hints at potential vocabulary limitations or preferences in the models' training.

Second, human reviews showed notably higher usage of informal or colloquial language. The word "like" appeared 1,294 times in human reviews compared to around 1,000 in AI reviews, and "really" made the top 10 list only for human reviews. This suggests that despite instructions to maintain style, the models tend toward slightly more formal language.

Most tellingly, the models showed clear preferences for formal transition words. Words like "however" (524), "despite" (523), and "yet" (646) appeared frequently in AI-generated reviews but didn't make the human top 10. Chi-square tests confirmed these differences were highly significant (all $p < 0.001$), with transition words showing the strongest effect sizes (Cramer's $V > 0.15$).

These patterns suggest that even when explicitly instructed to mimic human writing style, LLMs maintain certain "tells" in their word choice.

## Question I: Can LLMs detect their own text?

A superficial analysis of human texts and their LLM counterparts revealed that distinguishing between the two was almost impossible for me as a human observer. Subtle stylistic choices sometimes hinted at a text's human origin. For in-

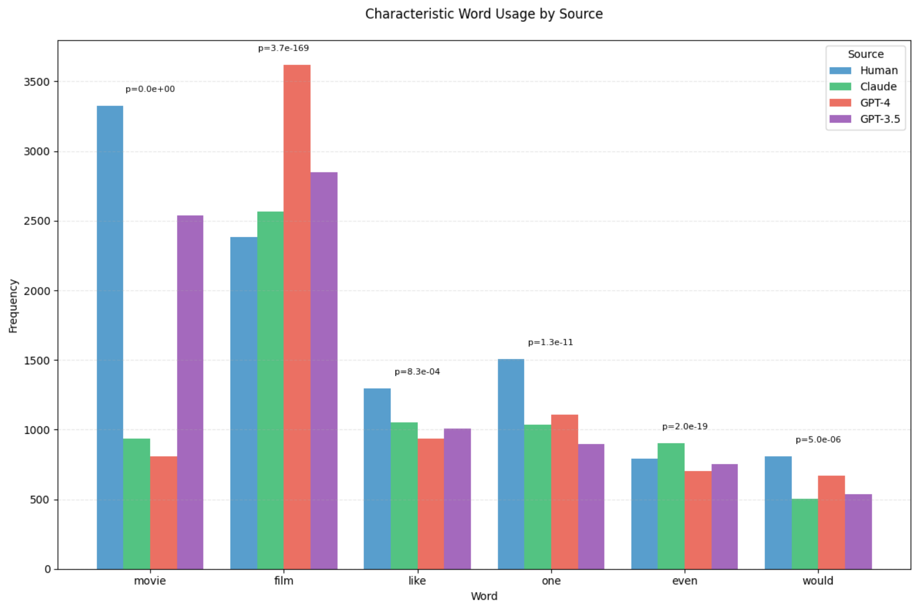


Figure 3: Term Frequencies with P-values on Chi-square

stance, a phrase like, “Anybody with a 50 IQ or above would hate this movie...” might be rephrased by the LLMs as, “Anybody with reasonable intelligence would hate this movie,” or something equally softened. This difference likely reflects the LLMs’ fine-tuning to adhere to content guidelines and avoid overtly offensive language.

To establish a baseline for discrimination, I tested whether the LLMs themselves could identify the source of a given text. My hypothesis was that the models might possess an implicit “sense” of their own outputs, based on the probability distributions underlying their token predictions.

I randomly sampled 100 indices from the dataset, resulting in 400 reviews (100 each from human, Claude, GPT-4, and GPT-3.5). The LLMs were tasked with classifying the source of each review using the following prompt:

> “You will be shown a movie review. In the set of reviews you’ll analyze, exactly 25 percent were written by humans, 25 percent by Claude, 25 percent by GPT-4, and 25 percent by GPT-3.5.
>
> Your task is to identify the most likely source of each review. You must respond with exactly one of these four options:
>
> - “Human”
> - “Claude”
> - “GPT-4”
> - “GPT-3.5”

> Important: Return ONLY one of these four options with no additional commentary or explanations."

This setup treated the task as zero-shot classification, with an additional statistical prior explicitly provided to the models: they were told the exact distribution of sources (25% each from human, Claude, GPT-4, and GPT-3.5). My hypothesis was that if LLMs could recognize characteristic patterns in writing style, this balanced framing would help elicit meaningful classifications.

The results revealed two striking patterns. First, when performing this task, LLMs showed strong biases that seemed more reflective of their training data than any ability to recognize writing styles. They overwhelmingly favored labeling texts as "Human," even when explicitly told that 75% of the reviews were AI-generated. This tendency artificially inflated their accuracy on human text classification (Figure 3) through what amounts to a default prediction strategy rather than actual detection capability.

Second, and perhaps more telling, was the models' treatment of Claude-generated text. None of the models regularly attributed text to Claude, with Claude itself only self-identifying once in 400 samples, and GPT-4 doing so just 3 times. GPT-3.5, notably, never predicted Claude as a source - but this makes sense given its training cutoff predates Claude's widespread availability. This pattern strongly suggests the models aren't actually recognizing stylistic patterns in the text, but rather making predictions based on what sources they've been trained to expect. The fact that even Claude rarely identifies its own text as Claude-generated further reinforces this interpretation.

These findings challenge the notion of LLM "self-recognition" as a meaningful capability. Rather than detecting genuine stylistic fingerprints, the models appear to base their predictions primarily on source frequencies in their training data. This might explain why newer models, despite their increased capabilities, appear even less adept at this task than their predecessors - they may be better at avoiding unfounded attributions when faced with uncertainty.

## Question II. Can standard classifiers do better?

I next sought to treat this as a standard 4-class classification task. First, I embedded the texts.

The embedding strategy employed BERT-Large (accessed via the Transformers library) to produce 1,024-dimensional vectors for each text. This choice was deliberate, as I hypothesized that higher-dimensional embeddings could capture subtle, nuanced differences in language style and structure between the outputs of various LLMs, potentially facilitating better discrimination between classes.

To enhance the quality of embeddings, I used an attentional pooling strategy. Specifically, attention mechanisms were applied to weigh tokens within each text based on their relative importance, allowing the embeddings to emphasize semantically significant elements. This method captures contextual nuances and relationships that might be critical for distinguishing between similar

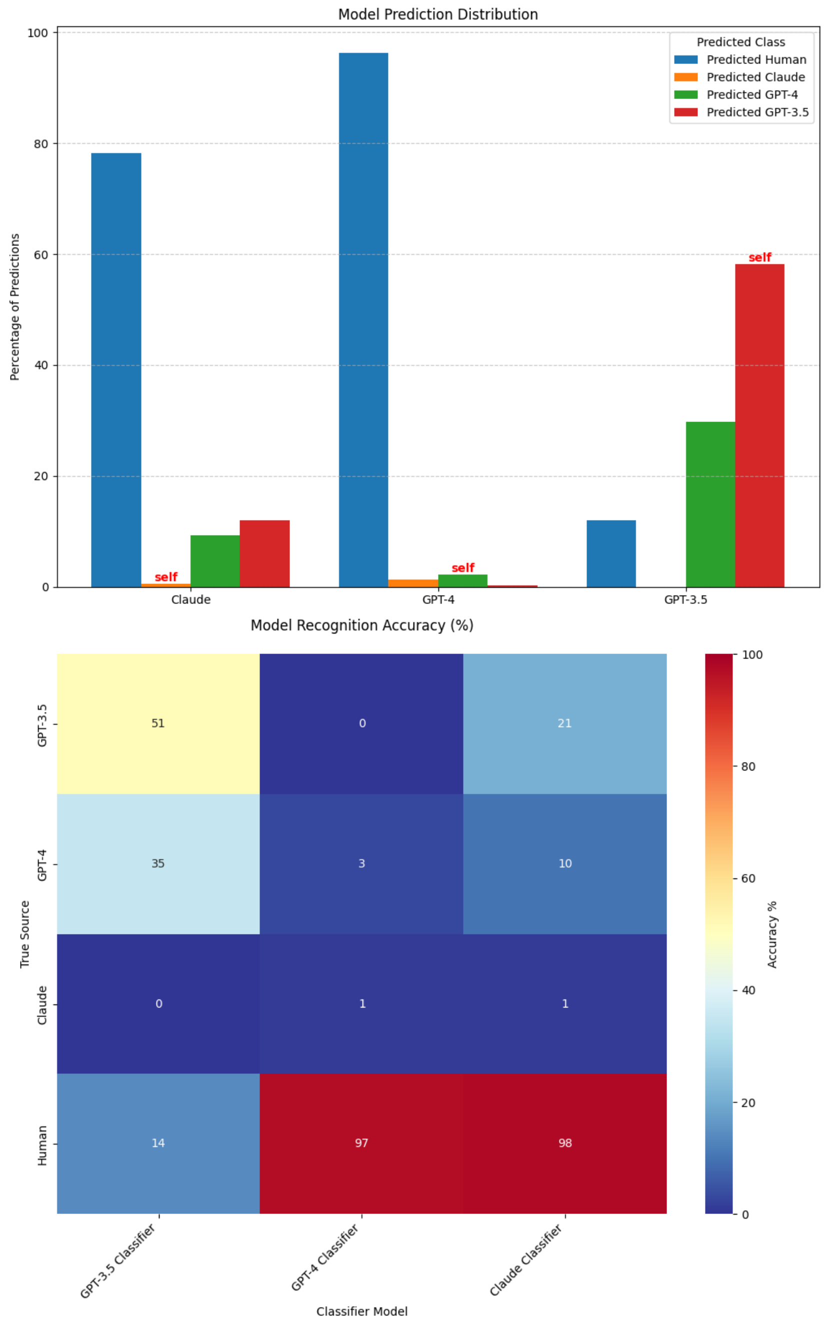


Figure 4: LLM Self-recognition performance

text sources, such as different versions of GPT or human-written reviews. For instance, attentional embeddings may give greater weight to tone, sentence construction, or word choice, all of which could subtly differ between model outputs.

With the resulting set of 8,000 embeddings (2,000 for each class: human, GPT-3.5, GPT-4, and Claude), I tested a variety of classifiers to evaluate whether the task of identifying model lineage was feasible. The classifiers tested included:

- Logistic regression
- A simple multilayer perceptron (MLP)
- Support vector machine (SVM)
- XGBoost
- Random Forest

All classifiers were trained and evaluated using a consistent 80-20 train-test split on the embedding dataset. This ensured that the evaluation metrics were directly comparable across methods, providing insight into which approach was most effective for this classification problem.

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1** |
|---|---|---|---|---|
| Logistic Regression | 0.844 | 0.845 | 0.844 | 0.844 |
| Neural Network | 0.832 | 0.831 | 0.832 | 0.831 |
| SVM | 0.807 | 0.807 | 0.807 | 0.807 |
| XGBoost | 0.736 | 0.735 | 0.736 | 0.735 |
| Random Forest | 0.653 | 0.650 | 0.653 | 0.651 |

Table 2: Comparison of Model Performance Metrics

The performance appears suspiciously strong at first glance, particularly for logistic regression. It is noteworthy that logistic regression outperformed more complex models, such as neural networks and XGBoost, suggesting that the classes might exhibit patterns in the BERT-Large embedding space that are exploitable by simpler models. However, visualizations using t-SNE and UMAP (Figure 5) reveal that while there are clusters corresponding to different sources, these clusters are not linearly separable. The embeddings suggest overlap between human and GPT-4 texts, with Claude and GPT-3.5 forming tighter concentrations within the larger embedding space. This overlap aligns with the observation that human-generated texts exhibit greater stylistic diversity, while LLM outputs are more constrained by their training and generation processes. The results suggest that the strong performance of logistic regression and SVM likely stems from the embedding space capturing subtle, non-linear patterns.

Additionally, in both the t-SNE and UMAP visualizations, a small side cluster comprising approximately 5–10% of the points was observed, containing samples from all four classes. This cluster likely represents texts with shared

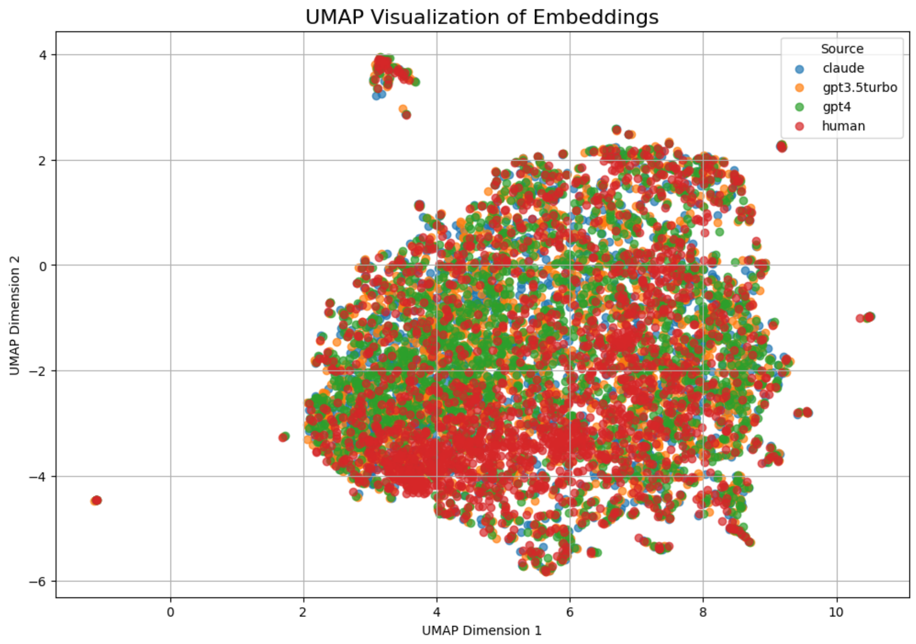


Figure 5: UMAP of Embeddings

stylistic or lexical traits that reduce distinctions between sources, such as highly generic or neutral phrasing. The presence of this cluster suggests certain texts occupy ambiguous embedding regions, potentially contributing to the overlap and misclassification observed in the model performance.

To see if I could surpass the performance of logistic regression, I built a more complex 5-layer MLP with dropout regularization and batch normalization. This model managed to outperform the baseline logistic regression but only by approximately 2

One intuitive result is that the models exhibit higher confusion between different versions of the same model (i.e., GPT-3.5 and GPT-4) compared to cross-model or cross-source comparisons (e.g., Claude vs. GPT, or LLM vs. human). This is evident in the confusion matrices for both the baseline logistic regression and the "complex" MLP, as shown in Figure 6.

## Question III. How much text do we need?

If LLMs leave "fingerprints" that enable classification in high-dimensional embedding spaces, an important question is: how much text is actually needed to detect these fingerprints? To explore this, I repeated the same procedure of embedding texts using BERT-Large and training a logistic regression classifier, but this time I progressively truncated the original texts.

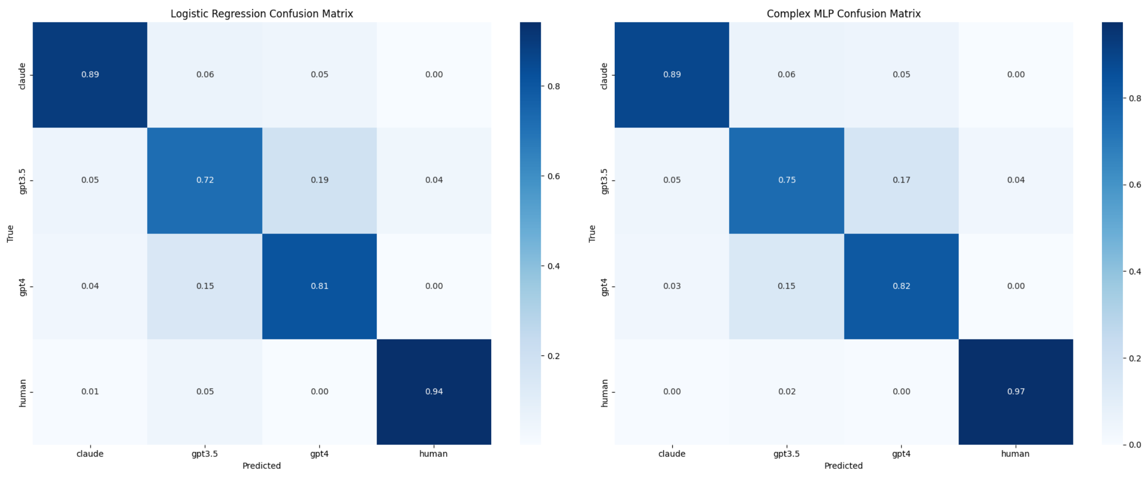


Figure 6: LogReg and MLP Confusion Matrices

The truncation process began with reducing the texts to the first 100 tokens (roughly two-thirds of the average review length). I then repeated the procedure for lengths of 75 tokens, 50 tokens, 25 tokens, 10 tokens, 5 tokens, and finally just 3 tokens. The F1 scores for each truncation are shown in Figure 3.

As expected, F1 scores degrade progressively as text length decreases. Specifically, there is an approximately 0.15 drop in F1 score when reducing the text length from 100 tokens to 50 tokens, followed by more rapid declines with further truncation. Even with just 3 tokens per text, the classifier achieves an accuracy approximately 5% above random chance.

I found this finding notable: the first 3 tokens appear sufficient to “buy” a small but measurable accuracy boost, while adding 3 more tokens to near-complete texts contributes comparatively less. This suggests that the semantic and stylistic fingerprints left by LLMs are detectable even in the opening words of a text, where patterns in phrasing, style, or token choice can differ subtly from human writing.

## Key takeaways

- Humans and LLMs use words differently. Basic lexical analysis suggests that there are a few potential “tells” for naive detection of AI generated vs. human text. Not surprisingly, human movie reviews favor informal and colloquial language, using words like “like” and “really” more frequently, while AI reviews exhibit a strong preference for formal transition words such as “however,” “despite,” and “yet.” These differences, confirmed by chi-square tests with significant effect sizes, are likely generalizable to some other domains of human and LLM text. If this pattern holds broadly, then

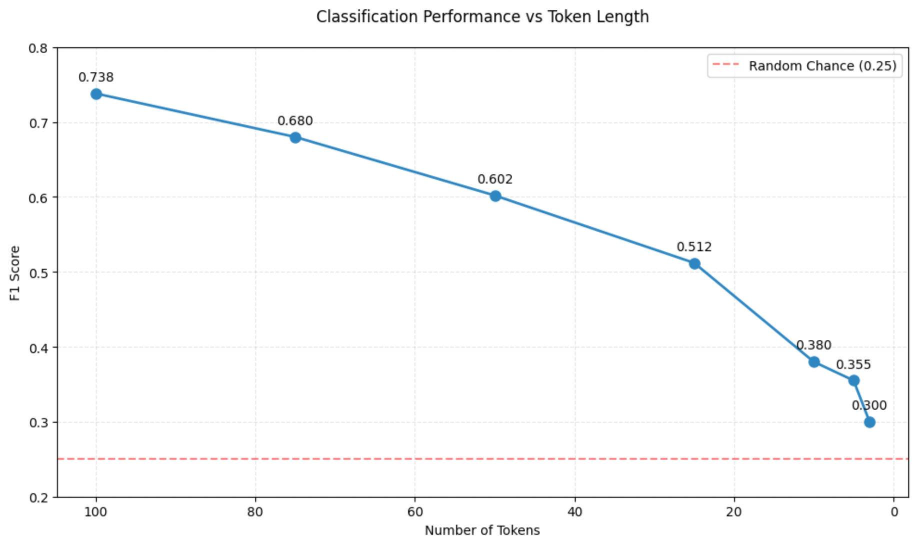


Figure 7: F1 Scores for Different Text Truncations with LogReg

adding tokens like “really” and “like” to AI-generated text is probably among the best “bang for your buck” approaches to beating LLM text detectors.

- Popular LLMs like Claude and GPT-4o perform poorly at identifying LLM-generated text when treated as a zero-shot task. Newer models, in particular, demonstrate a strong bias toward attributing text to humans and are especially poor at self-recognition. Given these results, I won’t be asking LLMs to try to identify whether something is human or not.

- When the task is narrowly constrained, such as rewriting and classifying movie reviews, high-dimensional attentional embeddings enable reasonably accurate classification, achieving over 80% accuracy in some cases. Class-specific accuracies align broadly with intuition: there is greater confusion between versions of the same model (e.g., GPT-3.5 and GPT-4) compared to cross-model or cross-source confusion (e.g., GPT vs. Claude or LLM vs. human). The strong performance of simpler models like logistic regression and SVM might have suggested some linear separability in the BERT-large embedding space, but further inspection of the space suggested otherwise. More investigation into why the simpler classifiers outperform here is something I’ll tackle in a future project.

- Under these constrained conditions, “fingerprints” left by different LLMs are evident, likely reflecting differences in training data, architectures, fine-tuning processes, and system prompts. These fingerprints are more pronounced when projected into higher-dimensional spaces using transformers, and they begin to emerge even within just a few tokens.

## Open Questions and Future Directions

I believe these findings raise many additional questions and potential research directions:

**Generalization to Larger Datasets:** The usual question: How well do these results generalize to larger and more diverse datasets? Would the observed patterns hold across different types of content, such as technical writing, creative prose, or social media posts?

**Scaling to More Models:**

Another usual question: How would the approach scale with a larger set of models, including unfiltered or less-common LLMs like Unholy or NSFW-6b? At what point might class proliferation effects dilute classification accuracy?

**Few-Shot or Many-Shot Self-Recognition:**

Some less usual questions: Could LLMs' self-recognition accuracy improve if the task is framed as a few-shot or many-shot problem? For instance, what if models are provided with examples of text generated by themselves or other models before performing classification? Could this approach help evaluate models' "metacognition" or ability to understand their own outputs?

**Alternative Classification Strategies:**

Also, could entirely different methods work better for attribution? For example, training attentional autoencoders to encode texts into a latent space and then reconstruct them. Reconstruction errors might serve as a proxy for attribution, with lower errors from a model-specific autoencoder suggesting a match to the model's text (e.g., a GPT-4o autoencoder producing low reconstruction error for GPT-4o text).

**Formalizing Detection Thresholds:**

Finally, can we identify some sort of mathematical "proof" around the detection thresholds for LLM-generated text. In other words, can we define the conditions under which it is possible—and not possible—to discriminate between AI-generated and human-authored content? Such a framework might extend beyond text to other content modes like images, audio, or video, offering broader utility across AI detection tasks.

**Utility (i.e., Would somebody be willing to pay for this?):**

There is considerable value in being able to reliably identify AI-generated content, especially in sensitive or high-stakes domains. Applications could include detecting misinformation, verifying content in legal contexts, moderating social media platforms, and validating authorship in academic or creative work. The popularity of tools like GPTZero—despite their widely recognized performance limitations—underscores the demand for this kind of technology. Teachers, legal professionals, social media moderators, and others could all benefit from effective AI detection.

That said, there are massive challenges here. Current detectors, including my own experiments and tools like GPTZero, are highly vulnerable to straightforward "hacks." Simple techniques, such as introducing a few typos, adding periods between tokens, or using invisible font colors, can render these tools ineffective. This highlights a critical limitation: many existing approaches are not robust enough for practical, real-world deployment.

Given these limitations, the experiments conducted here are far from being ready for a paid service. However, I believe there is some utility in a research context. The results and questions raised by this project point to a rich area of study with significant implications. Future research could delve deeper into detection robustness, explore generalized frameworks for identifying AI-generated content, and develop tools better suited for real-world deployment.

## A Quick Front-end

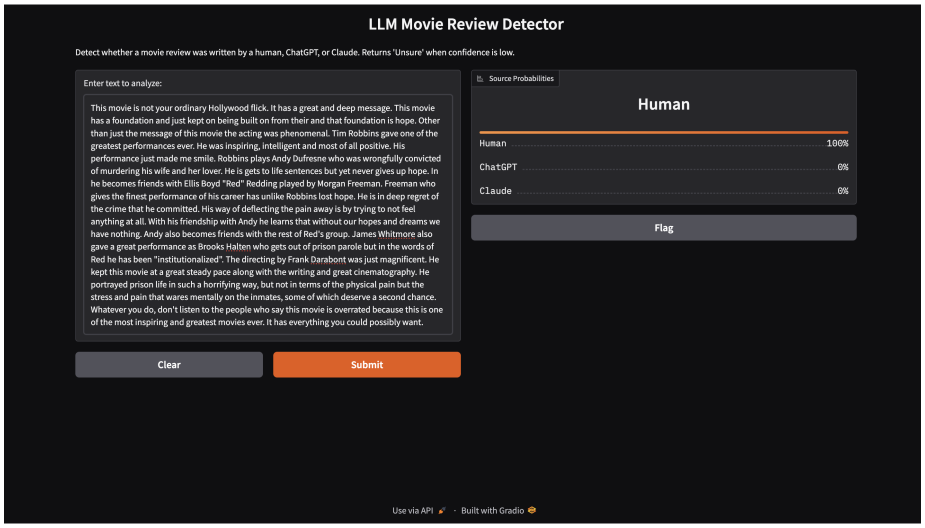
The final MLP for movie review classification is deployed on Hugging Face (Figure 6):

`https://huggingface.co/spaces/datboyalex/LLM_movie_review_detector`

Due to the discontinuation of GPT-3.5 turbo on ChatGPT, the GPT-4o-mini and GPT-3.5-mini classes have been aggregated into a single "ChatGPT" class for simplicity in this implementation.

As expected, the model performs well on reviews from its original dataset. However, in my limited testing with new IMDB movie reviews, it achieves only about 50% accuracy. This lower performance is likely due to overfitting on the training set, which was relatively small, as well as shifts in online writing styles since the original IMDB reviews were produced. For instance, the model frequently overpredicts recent human reviews (e.g., reviews for the 2024 movie Wicked) as AI-generated. That said, it's entirely possible that some IMDB reviewers are in fact using tools like ChatGPT or Claude to author or co-author their reviews.

The model's decision boundaries are another limitation. It often outputs 100% confidence for its predictions, likely a reflection of overly-rigid classification. Interestingly, the tool demonstrates some capability to correctly identify


LLM Movie Review Detector

Detect whether a movie review was written by a human, ChatGPT, or Claude. Returns 'Unsure' when confidence is low.

Enter text to analyze:

This movie is not your ordinary Hollywood flick. It has a great and deep message. This movie has a foundation and just kept on being built on from their and that foundation is hope. Other than just the message of this movie the acting was phenomenal. Tim Robbins gave one of the greatest performances ever. He was inspiring, intelligent and most of all positive. His performance just made me smile. Robbins plays Andy Dufresne who was wrongfully convicted of murdering his wife and her lover. He is gets to life sentences but yet never gives up hope. In he becomes friends with Ellis Boyd "Red" Redding played by Morgan Freeman. Freeman who gives the finest performance of his career has unlike Robbins lost hope. He is in deep regret of the crime that he committed. His way of deflecting the pain away is by trying to not feel anything at all. With his friendship with Andy he learns that without our hopes and dreams we have nothing. Andy also becomes friends with the rest of Red's group. James Whitmore also gave a great performance as Brooks Halten who gets out of prison parole but in the words of Red he has been "institutionalized". The directing by Frank Darabont was just magnificent. He kept this movie at a great steady pace along with the writing and great cinematography. He portrayed prison life in such a horrifying way, but not in terms of the physical pain but the stress and pain that wares mentally on the inmates, some of which deserve a second chance. Whatever you do, don't listen to the people who say this movie is overrated because this is one of the most inspiring and greatest movies ever. It has everything you could possibly want.

Source Probabilities

Human

```
Human ........................................ 100%
ChatGPT ...................................... 0%
Claude ....................................... 0%
```

Figure 8: Front end interface, tested on a real human review of The Shawshank Redemption (1994) not appearing in the original dataset

LLM-generated texts unrelated to movies. However, it is important to stress that this tool is highly limited and should not be used for formal LLM detection.

So, while the model serves as a marginally interesting proof of concept, it is not reliable for robust LLM detection. For more details on its performance and limitations, refer to the attached demonstration video included with this report.

## Data, Documentation, and Code

All project-related data, documentation, and code are available on GitHub. You can access the repository at:

`https://github.com/alex-amari/llms_final_project`.

## Technology and Sources

The entire project was implemented in Python, primarily using Google Colab for development and experimentation. All model training was conducted on an NVIDIA A100 GPU, while the front-end was hosted on Hugging Face Spaces using a CPU Basic cluster.

- **Transformers Library**: Used for embedding generation with BERT-Large.
- **IMDB Movie Reviews Dataset**: Used as human baseline texts.

- **BERT-Large**: Provided high-dimensional embeddings for text classification.
- **Claude 3.5 Sonnet**: Utilized for project assistance.
- **Anthropic API**: Utilized for text generation
- **GPT-3.5 Turbo and GPT-4o Mini**: Documentation for these models informed text generation processes.
- **GPT-o1**: Utilized for project assistance.
- **GPTZero**: Highlighted as an example of current AI detection tools.

## Acknowledgments

Huge thanks to Profs Kyle Jensen and K. Sudhir for such an awesome course. Thanks to awesome TAs Rudy Cordero and Richard Archer, who went above and beyond this whole quarter. Thanks to Prof. Casey King from Yale FDS and the Jackson School, whose 'Intro to AI' course helped inspire this project. And thanks to Prof. Tauhid Zaman for the general mentorship and OpenAI API key.