Precog - Technical Report

Venya Velmurugan

February 2026

1 Task 0: The Library of Babel

1.1 Implementation

This goal of this task is to construct an authorship controlled dataset where style, not topic, is the primary distinguishing factor. The dataset contains three classes: human-written paragraphs by some authors, AI-generated paragraphs in a neutral voice, and AI-generated paragraphs conditioned to mimic the chosen human authors. All classes share the same thematic anchors and are normalized to a consistent paragraph length (100–200 words) to prevent trivial cues such as length or formatting from driving classification.

1.1.1 Dataset Structure

- Class 1 — Human text: Original paragraphs extracted from novels by Jane Austen and Elizabeth Gaskell
- Class 2 — AI-neutral text: AI-generated paragraphs on the topics present in the human datasets in a neutral style
- Class 3 — AI-styled text: AI-generated paragraphs on the same topics mimicking Austen or Gaskell writing style

Each paragraph is treated as an independent sample. Topic labels are shared across classes so that downstream models must rely on authorship rather than semantic content.

1.1.2 Author Selection

Jane Austen has always been one of my favourite authors so I decided to choose her and my aim while choosing the other author (ELizabeth Gaskell) was that the stylistic differences between the 2 authors are minimum so that the classification would be more difficult. When authors differ strongly in period, register, or syntactic conventions, classifiers may reach high performance by exploiting surface level cues like spelling variation, orthographic conventions, or extreme syntactic differences, rather than learning meaningful stylistic patterns.

By choosing authors with broadly comparable narrative style, register, and genre, the dataset will no longer rely on obvious author- or era-specific markers and instead encourages models to attend to subtler stylistic regularities. This design choice reduces the risk of inflated performance due to artificial correlations and provide a more stringent test of whether models can capture stylistic fingerprints rather than coarse textual differences.

Jane Austen and Elizabeth Gaskell were selected as they enable a controlled comparison along these dimensions:

By selecting authors with broadly comparable narrative style, register, and genre, the dataset makes sure that classifiers do not rely on obvious author or era-specific markers and instead forces them to pay attention to subtler stylistic regularities. This will help reduce the risk of inflated performance due to artifical correlations and provides a stricter test of whether models can genuinely capture stylistic fingerprints rather than coarse textual differences.

Jane Austen and Elizabeth Gaskell were chosen because they satisfy a controlled comparison:

- strong thematic overlap (family, class, morality, domestic life)
- distinct stylistic features
- same historical period and so similar style and language baseline (19th-century British English)
- clean, public-domain availability via Project Gutenberg

1.1.3 Source Material (Class 1)

Six novels were used, 3 by each author:

Jane Austen

- Pride and Prejudice
- Sense and Sensibility
- Emma

Elizabeth Gaskell

- North and South
- Mary Barton
- Cranford

The raw texts were downloaded from Project Gutenberg and processed into uniform paragraph units.

1.1.4 Preprocessing Pipeline

Project Gutenberg texts contain boilerplate, formatting artifacts, and inconsistent paragraph structure. A cleaning pipeline standardizes the corpus:

All downloaded texts were cleaned to remove:

1. HTML parsing and removal of Gutenberg headers/footers
2. normalization of drop caps and formatting artifacts
3. removal of page markers, chapter labels, and license text, chapter metadata, editorial notes and OCR artifacts
4. filtering of non-narrative fragments
5. re-chunking into 100-200 word windows while preserving sentence boundaries

This re-chunking step ensures that paragraph length cannot act as a shortcut feature during classification.

The resulting human dataset (Class-1) has about 4,000+ paragraphs.

1.1.5 Extraction

Nine corpus-level themes were identified across the six novels:

1. Courtship and Marriage
2. Domestic Life and Family Obligation
3. Social Class and Reputation
4. Moral Judgment and Personal Character
5. Gender Roles and Social Constraint
6. Work, Industry, and Economic Struggle
7. Community, Gossip, and Social Networks
8. Individual Desire vs Social Expectation
9. Change, Mobility, and Social Reform

Topics function as semantic anchors, not per-book labels. All paragraphs generated text within this set of topics to ensure that there is no topic bias.

1.1.6 AI Generation Model

Although the task document suggested using the Gemini API, Gemma was used instead due to accessibility constraints. This choice impacted the paragraph generation methodology for the rest of the task.

1.1.7 Initial Attempt: Deterministic Generation

The first run of the prompts revealed that Gemma's outputs were highly deterministic and it kept producing the same paragraph for the same prompt.

To overcome this problem, multiple prompt-engineering strategies were tried:

All AI text is generated using Google's Gemini API with the Gemma 3-27B model. This model was selected as it could produce large datasets without quota interruption and temperature and sampling width could be varied across dataset variants to control diversity.

1.1.8 Try 1: Topics–Lenses–Constraints (PnC-style prompting)

In the first approach, AI-generated paragraphs were produced using a structured prompt specifying:

- A fixed topic
- A conceptual lens (e.g., ethical, psychological, historical)
- Explicit constraints on tone and content (e.g., begin with so and so, etc.)

While this produced coherent and topic-consistent text, running preliminary analysis on this data (Task 1) found that these samples were highly distinguishable from human text. In particular:

- Hapax legomena counts were significantly lower and TTR value was significantly higher
- Punctuation density was much lesser in the Ai generated text as compared to the human one
- Sentence structures were overly uniform across samples, readability was very high

From these findings, we can understand that the prompt structure itself acted as a strong stylistic prior. Rather than approximating human variability, the Topics–Lenses–Constraints framework let a uniform organizational template that resulted in reduced lexical diversity, constrained punctuation usage, and repetitive sentence patterns, making the AI-generated text easily distinguishable from human writing.

1.1.9 Try 2: Story Generation + Segmentation

To reduce prompt-induced changes, a story based approach was adopted:

- Generate longer narrative passages or short stories
- Segment these stories into 100–200 word paragraphs

This method successfully increased lexical diversity and reduced repetition. However, Task 1 analysis still showed anomalies:

- Overly smooth POS distributions
- Limited syntactic variance between segments
- Residual stylistic homogeneity across paragraphs derived from the same generation
- The segmentation process did not sufficiently break the overall stylistic coherence imposed by the initial generation.

1.1.10 Try 3: Few-Shot Paragraph Prompting (Final Approach)

The final version of the dataset was generated using few-shot prompting, where each paragraph was produced independently with minimal structural guidance. A small number of example paragraphs were included in the prompt to implicitly convey stylistic cues, while avoiding rigid constraints that could introduce artificial regularities.

Each prompt specified:

- the target topic
- for Class 3, a high-level stylistic guide to mimic the chosen author
- a limited set of example paragraphs to ground tone and prose style

1.1.11 Sampling and Decoding Parameters

In addition to prompt design, decoding parameters were systematically varied across dataset construction attempts to control generation diversity and reduce deterministic behavior. These parameters include temperature, top-k, and top-p, which jointly regulate randomness and lexical variation during generation. The exact configurations were adjusted across different dataset variants and are documented in the corresponding notebooks.

In **Try 1** (`Task_0_try1.ipynb`), paragraph-level generation was performed using moderately high diversity settings:

- temperature = 0.95
- top-p = 0.9
- top-k = 50
- max output tokens = 450

Despite these settings, outputs remained highly regular due to the structured nature of the Topics–Lenses–Constraints prompt, resulting in reduced lexical diversity and uniform sentence structure.

In **Try 2** (`Task_0_try2.ipynb`), generation shifted to story-level outputs followed by segmentation, with increased sampling breadth:

- temperature = 1.0

- top-$p = 0.99$
- top-$k = 50$
- max output tokens = 3500

These settings increased lexical diversity at the surface level. However, stylistic coherence across paragraphs persisted because multiple samples originated from the same long-form generation.

In **Try 3** (`Task_0_try3.ipynb`), sampling diversity was further increased and differentiated by generation type. Neutral and styled generations used separate configurations:

- **Neutral generation:**
  - temperature = 1.3
  - top-$p = 0.96$
  - top-$k = 200$
- **Styled generation:**
  - temperature = 1.4
  - top-$p = 0.98$
  - top-$k = 250$

In both cases, max output tokens were set to 2500 to support longer narrative generation prior to paragraph segmentation. These wider sampling settings encouraged rare word usage and stylistic variation, which—when combined with few-shot paragraph prompting—produced outputs that most closely matched human stylometric distributions in Task 1.

Overall, these results indicate that while increased sampling diversity is necessary to reduce deterministic artifacts, it is insufficient on its own. Meaningful alignment with human writing patterns emerged only when decoding adjustments were coupled with changes in prompt structure.

No explicit constraints were imposed on sentence structure, rhetorical organization, or paragraph formatting. This was because in earlier iterations, where structured prompting introduced detectable changes in lexical diversity and punctuation usage. By allowing the model to infer style from examples rather than specifying it through constraints, few-shot prompting reduced prompt-induced scaffolding while preserving stylistic coherence.

Text generated using this approach exhibited greater variance across samples, reduced repetition in lexical choice, and punctuation patterns that more closely aligned with those observed in human-authored paragraphs. Quantitative analysis in Task-1 confirmed that these samples showed fewer statistical deviations from human text compared to earlier prompting strategies.

All samples were constrained to 100-200 words and generated using a fixed set of nine thematic anchors shared across all classes. The class structure and paragraph counts were held constant across dataset variants, enabling direct and controlled comparison in downstream detection experiments.

Only authorship style remains a reliable signal. This makes the dataset suitable for evaluating authorship attribution, AI detection, and stylistic modeling under controlled semantic conditions.

1.2 Theory and Background Knowledge

This task is based on stylometry, the study of quantifiable linguistic features that characterize authorship. Prior work in authorship attribution shows that:

- Style manifests in function word usage, syntactic patterns, and punctuation habits.
- Topic-control in datasets are essential to avoid artificial correlations.

From an NLP standpoint, large language models are challenging because they are optimized to produce text that is fluent and coherent and so the text does not reflects the full variability of human writing. This optimization often results in outputs that appear polished but are uniform with no irregularities, inconsistencies, and stylistic drifts which are commonly found in human writing.

In addition, prompt design plays a crucial role in generating text. Structured prompts can inadvertently impose recurring uniform patterns that make AI-generated text easier to identify. For this reason, dataset construction is not merely a technical preprocessing step, but a methodological choice that strongly influences the conclusions of subsequent experiments.

1.3 Expected Results

Initially, it was expected that:

- Human-written text would show higher lexical irregularity.
- AI-generated text would be smoother, more repetitive, and more uniform.
- Human-mimicked AI (Class 3) would lie somewhere between Class 1 and Class 2.

It was also expected that these differences would be more explicit in early prompt designs, while later refinements would reduce them.

1.4 Results and Observations

Task 1 analysis confirmed these expectations and directly influenced dataset revision:

- Try 1 outputs were trivially separable using simple statistical metrics such as hapax legomena and punctuation density.

- Try 2 outputs improved surface diversity but retained deeper structural regularities.
- Try 3 was closest to human text in lexical richness, POS ratios, and syntactic depth.

Thus, the final dataset used for Tasks 2–4 is the result of iterative refinement of prompt, not a single-shot design.

1.5 Initial Challenges

Several challenges emerged during Task 0:

- Deterministic behavior of the chosen LLM
- Prompt engineering acting as an unintended factor which determines writing style
- Balancing topic control with stylistic freedom
- Avoiding overfitting the dataset to obvious AI markers

An important point to note is that all these challenges were identified only after statistical analysis in Task 1.

1.6 Limitations

- The dataset construction process depends heavily on manual prompt design, which may introduce unintentional biases.
- Generated text reflects the behavior of a single language model and may not represent the full diversity of AI-generated writing.
- Controlling the writing style and making it learn nuances of human writing but without controlling the topics is inherently difficult, and some trade-off is unavoidable.
- Human-written text is sourced from a limited set of authors and genres, which will restrict stylistic diversity based on era and times of the author. Chosen dataset is mainly on 19th century novels, so model in further tasks will become trained to pick up differentiate on that type of data mainly and not contemporary, modern texts.
- Manual inspection alone was insufficient to detect many artifacts, requiring later statistical validation.
- The dataset reflects just 19th century English-language writing only and does not generalize across languages or cultural contexts.

1.7 Learning Outcomes

- Dataset construction is an iterative, hypothesis-driven process
- Design of the prompt can influence stylistic signals more than the model itself
- Statistically analyse the generated dataset to remove too obvious differences before training classifiers

2 Task 1: The Fingerprint

2.1 Implementation

The goal of Task-1 is to verify that the three classes in the dataset (Human, AI and AI-mimic) have considerable measurable statistical differences between them in terms of stylometric features like punctuation density, TTR,etc. Based on the results of this task, we also went back and improved the prompting techniques for Task-0 (as discussed before). We want the classes to be similar enough to avoid trivial shortcuts to distinguish between them, but at the same time we want to make sure that they are not identical. Mathematical distinctness just means there is some measurable signal (even if it's weak and overlapping) so that any success or failure can be interpreted meaningfully.

For each paragraph in the dataset, a set of stylometric features (lexical, syntactic, punctuation-based, and readability features) were extracted considering the paragraphs to be independent samples. In addition, some information-theoretic features were also extracted to measure the predictability vs creativity of AI vs human datasets. The analyses were performed at the paragraph level to maintain consistency with downstream classification tasks. Only the stylometric 8 features are taken forward in Task-2 as they were the specifications.

The final feature set consists of 10 metrics, grouped as follows:

- Lexical richness (MSTTR, Hapax Legomena)
- Syntactic complexity (POS ratios, dependency depth)
- Punctuation density
- Readability (Flesch–Kincaid Grade Level)
- Information-theoretic predictability (Conditional Entropy, Perplexity)

The 8 mandatory features of Task-1 (all excluding Information thoeretic predictability) are stored in a structured format and exported for use in Task-2.

2.1.1 Statistical Significance Testing

To compare human-written and AI-generated text, simple statistical tests were applied to each extracted feature. For every feature, a two-sample t-test was used to check whether the average values for the human and AI classes differed.

Alongside this, Cohen's d was calculated to check how large these differences actually are. With a large number of samples, even very small differences tend to produce extremely low p-values, making statistical significance alone a misleading metric.

So, the p-values are reported only for completeness and Cohen's d is used as the main criteria for interpretation, as it shows the practical importance of stylistic differences rather than their just their identifiability.

2.2 Theory and Background

Stylometry is based on observing the consistent linguistic patterns in the writing styles of authors. These stylometric habits are often subconscious and persist across all topics and works of the author, so they become very useful for authorship detection.

- Lexical Richness : Captures how diverse a writer's vocabulary is.
  - Mean Segmental Type–Token Ratio (MSTTR): Computes the average type–token ratio over fixed-size segments. Preferred over normal TTR to ensure that size of the dataset does not affect this metric (in our task, human dataset has 4k paragraphs which is much more than the AI paragraphs 1k only, so MSTTR will help ensure no dataset length bias)
  - Hapax Legomena : Counts words appearing exactly once in a sample.
- Syntactic Complexity : Shows how writers organize ideas structurally
  - POS Distribution (Adjective-to-Noun Ratio): Can be used to describe descriptive density or how many of the nouns are described and have adjectives attached to them.
  - Dependency Tree Depth: Computed using SpaCy. Measures syntactic nesting and how complex and hierarchical a sentence is.
- Punctuation Density : Frequency of punctuation marks such as exclamation marks, semicolons, and em-dashes. These patterns are difficult to control by prompts as they are creative and emotional aspects of a sentence and AI is not creative. These are usually low unless specified explicitly and thus act as strong stylistic markers.
- Readability: Flesch–Kincaid Grade Level is a readability formula that converts text reading complexity into a U.S. school grade level (e.g., a score of 8 means an 8th grader can understand it). It is calculated based

on average sentence length and average syllables per word, with lower scores indicating easier reading.

$$\text{FKGL} = 0.39 \times \left(\frac{\text{Total Words}}{\text{Total Sentences}} \right) + 11.8 \times \left(\frac{\text{Total Syllables}}{\text{Total Words}} \right) - 15.59$$

- Information-theoretic measures: Beyond surface-level stylometry, we can also include probabilistic metrics that quantify predictability. They can be used as a measure of how creative the AI is.

  Conditional entropy measures the average uncertainty in predicting the next token given its preceding context.
  Formula:

  $$H(T_i \mid T_{<i}) = -\frac{1}{N} \sum_{i=1}^{N} \log P(T_i \mid T_{<i})$$

  where T_i denotes the current token, $T_{<i}$ denotes all preceding tokens, and N is the total number of tokens in the text.

  - **Perplexity**: Perplexity is an exponential transformation of average negative log-likelihood and represents how ”surprised” a language model is by a given text. It is computed as:

    $$\text{Perplexity} = \exp \left(-\frac{1}{N} \sum_{i=1}^{N} \log P(T_i \mid T_{<i}) \right)$$

2.3 Expected Results

- **Lexical Richness**
  - **MSTTR**
    - **Human**: High and more variable MSTTR values.
    - **AI**: Low MSTTR values.
    - **AI-human mimic**: Intermediate MSTTR values.
    - **Reason**: Humans are said to be spontaneous and creative so higher. AI-generated text is expected to show lower MSTTR, due to AI usually being thought of as “less-creative”. AI-human mimic text is expected to lie between the two, as it is meant to try replicating human diversity while at the same time producing text with model-level regularities.
  - **Hapax Legomena**
    - **Human**: High hapax legomena count.
    - **AI**: Low hapax legomena count.
    - **AI-human mimic**: Intermediate hapax legomena count.

        - **Reason**: Human text is expected to have a higher count of hapax legomena, again due to creativity which is expected to lead to low-frequency in lexical repetition. AI-generated text on the other hand is expected to reuse words, resulting in lower hapax counts. AI-human mimic text is expected to be try to expand its vocabulary but at the same time not fully match human levels.

- **Syntactic Complexity**
    - **POS Distribution (Adjective-to-Noun Ratio)**
        - **Human**: Balanced adjective-to-noun ratio.
        - **AI**: High adjective-to-noun ratio.
        - **AI-human mimic**: Reduced but still regular adjective-to-noun ratio.
        - **Reason**: AI-generated text is expected to show a higher adjective-to-noun ratio, as language models tend to over-describe nouns to improve perceived fluency. Human text is expected to show more balanced distributions. AI-human mimic text is expected to reduce this over-description but still remain more regular than human writing.
    - **Dependency Tree Depth**
        - **Human**: High variance and deeper dependency trees.
        - **AI**: Flatter and more uniform dependency trees.
        - **AI-human mimic**: Intermediate depth with reduced variance.
        - **Reason**: Human-written text is expected to exhibit greater variance in trees as well as have deeper dependency trees, as humans have greater capabilities of understanding and therefore generate complex sentences. This is expected to result in nested clauses and irregular sentence construction. AI-generated text is expected to have flatter and more uniform structures. AI-human mimic text is expected to approximate human depth but with reduced variance.

- **Punctuation Density**
    - **Human**: High and varied punctuation usage.
    - **AI**: Low punctuation density.
    - **AI-human mimic**: Increased but regular punctuation usage.
    - **Reason**: Human writing is expected to display richer and more variety in punctuation usage as humans are considered better at expressing their emotions (punctuation marks, question marks, etc.). AI-generated text is expected to have lower punctuation density, particularly for expressive punctuation, unless explicitly mentioned in the prompt. AI-human mimic text is expected to increase punctuation usage but still exhibit more regular patterns than human text.

- **Readability (Flesch–Kincaid Grade Level)**
  - **Human**: Broad variation in readability.
  - **AI**: Narrow readability range.
  - **AI-human mimic**: Partially broadened readability range.
  - **Reason**: Human-authored paragraphs are expected to have natural variation in sentence length and structure making them easy to read while AI-generated text is expected to cluster around a narrower readability range, reflecting optimization for clarity and accessibility. AI-mimic text is expected to partially broaden this distribution while remaining more regular than human writing.
- **Information-theoretic Measures**
  - **Conditional Entropy**
    - **Human**: High conditional entropy.
    - **AI**: Low conditional entropy.
    - **AI-human mimic**: Intermediate conditional entropy.
    - **Reason**: Human-written text is expected to exhibit higher conditional entropy, due to creativity which leads to more irregularity and unpredictability in token sequences. This is because humans have natural stylistic drift, uneven sentence construction, and creative deviations that are difficult to model probabilistically. AI-generated text is expected to have lower conditional entropy, as LLMs are trained to be uniform and so produce high-likelihood, predictable continuations with no creativity. AI-human mimic text is expected to increase entropy relative to AI-neutral text, but still remain more predictable than human writing.
  - **Perplexity**
    - **Human**: High perplexity.
    - **AI**: Low perplexity.
    - **AI-human mimic**: Intermediate perplexity.
    - **Reason**: Human-authored paragraphs are expected have higher perplexity scores when evaluated under a modern language model, as they deviate from the model's learned distribution due to human creativity. AI-generated text is expected to show lower perplexity, showing closer alignment with the model's internal probability distribution. AI-human mimic text is expected to fall in an intermediate range, appearing more human-like while still retaining models regularities.

2.3.1 Interpreting Statistical Significance

In this task, we work with a large number of paragraphs. When the sample size is large, p-values become almost meaningless as even very tiny differences between human and AI text will appear statistically significant.

A p-value only answers the question: Is the difference exactly zero? With thousands of samples, the answer is almost always "no", even when the difference is too small to matter.

Because of this, effect size is more important than statistical significance. We use Cohen's d to measure how large the difference between human and AI distributions actually is:

$$d = \frac{\mu_{\text{human}} - \mu_{\text{AI}}}{\sigma_{\text{pooled}}}$$

This tells us how far apart the two groups are in terms of standard deviations and independent of sample size.

In this analysis, conclusions are therefore based primarily on Cohen's d. P-values are reported only for completeness and are not used to judge whether a stylistic difference is meaningful as it has large 4k + paragraphs dataset.

2.4 Results and Observations

The results are presented separately for each dataset construction attempt (Try 1, Try 2, and Try 3).

Since the dataset contains a large number of samples, effect sizes (Cohen's d) are used as the primary criteria for interpretation, not p-values.

2.4.1 Try 1: Topics–Lenses–Constraints Prompting

In Try 1, all ten features show statistically significant differences between human and AI-generated text, with four features showing large effect sizes. This indicates that the classes are trivially separable in this prompting approach.

Most deviations from expected behavior are because of the highly structured nature of the prompt which details too much in terms of topic, way of starting,etc.

- **Lexical Richness (MSTTR, Hapax Ratio)** deviates from expectations. AI-generated text exhibits higher MSTTR and hapax ratios than human text. This occurs because the Topics–Lenses–Constraints fapproach forces the model to explicitly explore topic-specific vocabulary, artificially inflating lexical diversity while suppressing natural repetition patterns.

- **Syntactic Complexity: Adj–Noun Ratio follows expectations but Dependency Depthdoes not).** AI-generated text has higher adjective-to-noun ratios due to over-describing but at the same time this over-description because of prompt structure also leads to deeper dependency trees.

- **Punctuation Density** behaves as expected. Human text has significantly much much higher use of semicolons, em-dashes, and exclamation marks, with large to medium effect sizes. This shows that AI-generated text does not use much punctuation for expressions if not explicitly instructed to do so.

- **Readability (Flesch–Kincaid Grade)** shows a very large effect size, with AI-generated text being significantly harder to read. This is due to the long, formally structured sentences with over-description and greater depth in dependency tree constructed by the prompt due to the constraints framework.

- **Information-theoretic Measures** follow expectations suggesting that both texts remain broadly model-aligned.

Overall, Try 1 demonstrates that strong prompt scaffolding identified some strong stylistic features that heavily overshadow intrinsic differences between human and AI writing, hence we must try different prompting strategies to overcome this.

2.4.2 Try 2: Story Generation + Segmentation

Try 2 shows reduced separability compared to Try 1, with fewer large effect sizes and more features moving toward expected behavior.

- **Lexical Richness (MSTTR)** aligns with expectations, showing negligible difference between human and AI text. Story-level generation increases repetition within a narrative arc for the AI text, and therefore reduces artificial lexical diversity.

- **Hapax Ratio** is also higher for human text, as expected. Segmenting long stories does not introduce new rare words, revealing the underlying lexical regularity of AI-generated text.

- **Syntactic Complexity** shows medium effect sizes, with human text exhibiting greater depth and variance. This reflects the persistence of global syntactic coherence across segmented AI paragraphs.

- **Punctuation Density** remains strongly discriminative, with large effect sizes for semicolons and medium effect sizes for em-dashes and exclamation marks. Segmentation does not break punctuation regularities imposed during generation.

- **Readability** deviates from Try 1: AI-generated text becomes easier to read than human text. This is expected, as segmentation shortens sentences and reduces syntactic load.

- **Information-theoretic Measures** follow expectations. AI text exhibits lower conditional entropy and substantially lower perplexity, indicating higher predictability under the language model.

Try 2 confirms that segmentation improves surface-level diversity but does not eliminate deeper structural and probabilistic regularities inherited from long-form generation.

2.4.3 Try 3: Few-shot Paragraph Prompting

Try 3 reaches closest to human-generated text metrics across all features and best satisfies the goal of non-trivial separability.

- **Lexical Richness (MSTTR, Hapax Ratio)** behaves as expected, with negligible effect sizes, indicating that few-shot paragraph-level prompting successfully matches human lexical variability.

- **Syntactic Complexity** shows minimal differences. Dependency depth differences become negligible, while adjective-to-noun ratios retain a small effect size, suggesting residual over-description by the model.

- **Punctuation Density** remains the most discriminative category. While exclamations and emdashe usage is similar to ai, semicolon use still shows a large effect size, and exclamation marks show medium effect sizes, showing that expressive punctuation continues to be the most difficult for the model to fully reproduce.

- **Readability** aligns with expectations, showing negligible differences between human and AI text.

- **Information-theoretic Measures** converge substantially. Conditional entropy differences are negligible, and perplexity differences reduce to small effect sizes.

Overall, Try 3 achieves the intended design goal: the classes are mathematically distinct but not trivially separable, making this dataset suitable for downstream classification and interpretability experiments.

2.5 Initial Challenges

As mentioned in Task-0, one of the major challenges while working on Tasks 0 and 1 is that many of which were not obvious during dataset construction and only became clear after running quantitative analyses.

- **Deterministic generation**: Explained in Task-0

- **Prompt-induced stylistic bias**: Some of the initial prompt designs unintentionally imposed rigid structural patterns on the generated text. These patterns introduced strong artificial regularities that dominated stylometric features, making the AI-generated text trivially distinguishable from human writing.

- **Balancing control and freedom**: Maintaining strict topic control while still allowing stylistic variation proved challenging. Prompts that were too constrained improved topical alignment but led to unnaturally uniform writing, without any true style.

- **Delay in detecting limitations**: Many of these issues were not apparent by manual reading the generated paragraphs and only surfaced after statistical analysis in Task 1. As a result, several earlier design choices had to be revisited and revised.

2.6 Limitations

- Focus mostly on surface-level stylometric features. Might miss deeper discourse structure, narrative flow, or pragmatic choices that also contribute to writing style.

- Each feature is analyzed independently. Possible interactions between features (for example, between syntax and punctuation) are not taken into consideration.

- Automatic NLP tools such as SpaCy are used which might lead to some errors in POS tagging or dependency parsing which will then propagate into the extracted features.

- All measurements are done at the paragraph level. This ignores longer-range stylistic patterns that are usually present across paragraphs or at the document level.

- Conditional entropy and perplexity are calculated using a specific language model, so they show predictability relative to that model rather than an absolute measure of creativity.

- Due to the large number of samples, p-values become almost always significant. Although effect sizes are used for interpretation, statistical thresholds should still be treated with caution.

- The observed differences are tied to the specific generation model and prompting strategies used in this project and may not directly generalize to other models or prompting setups.

2.7 Learning Outcomes

- Dataset construction has direct implications for downstream classification results.
- Prompt design was found to introduce stylistic signals that can be as influential as the language model itself (showed importance of being able to write the ”best” prompt)
- Performing statistical analysis before training classifiers helped in identifying and removing trivial shortcuts that could otherwise inflate performance.
- Effect sizes (Cohen’s d) provided more meaningful insight than p-values when working with large datasets and subtle stylistic differences.
- Thinking adversarially during dataset construction helped reduce unintended correlations that models could exploit. (Similar to task 0)
- Repeated refinement and statistical validation, rather than relying on intuition or single-shot design choices.

3 Task 2: The Multi-Tiered Detective

In this task, the goal is to train multiple classifiers to separate human-written text from AI-generated text. These models are then organized into tiers of increasing complexity so that we can understand which kinds of stylistic signals are captured by simple statistical models, semantic representations, and full contextual transformers.

3.1 Implementation

Three detector tiers were implemented across all 3 dataset tries:

- **Tier A (The Statistician)**: Classical machine learning using only stylometric features extracted in Task 1.
- **Tier B (The Semanticist)**: A neural classifier operating over averaged semantic embeddings.
- **Tier C (The Transformer)**: A fine-tuned transformer model using parameter-efficient adaptation.

All models are trained on the same paragraph-level samples and evaluated under identical train–test splits (75-25) to ensure fair comparison.

3.1.1 Tier A: Random Forest Classifier

Tier A uses a Random Forest classifier trained solely on the eight stylometric features recommended in Task 1.

Why Random Forest instead of XGBoost:

- Random Forests are better at scaling features and nonlinear interactions without a lot of hyperparameter tuning.
- They are less prone to overfit on small, low-dimensional feature spaces compared to gradient-descent based models.
- Feature importance scores are more stable and easier to interpret for analysis.

It was preferred over XGBoost because XGBoost has performance advantages that are signifact only in larger, more complex feature spaces, whereas the goal here is interpretability and not accuracy.

3.1.2 Tier B: Semantic Classifier using Averaged GloVe Embeddings

Tier B uses a shallow feedforward neural network trained on averaged word embeddings for each paragraph. It distinguish using distributed word representations. Unlike Tier A, which relies on explicit stylometric features, this tier operates on paragraph-level semantic embeddings and is not influenced syntactic structure, punctuation, or handcrafted stylistic cues.

- **Why GloVe instead of FastText?**

  GloVe embeddings represent word-level semantics without being influenced by subword or character-level information. While FastText also encodes semantic similarity, its use of subword representations and will capture morphological and orthographic patterns.

  Each paragraph is just represented by the mean of its word embeddings, and does not depend on word order but just pure word level semantics as that was the goal of this tier.

- **Handling class imbalance**:

  The dataset contained a larger number of human-written paragraphs than AI-generated ones. To address this imbalance, class weights were applied during training so that errors on the minority class (AI-generated text) are given more weights. This prevents majority-class bias and ensures that model performance is dependent on learnt feature and not just than label frequency.

3.1.3 Tier C: Transformer-based Classifier with LoRA

Tier C is a pretrained transformer model and the goal is to fine tune it to distinguish between human-written and AI-generated text across all three dataset construction strategies.

Why a transformer-based model?

Transformers encode contextual information across entire sequences. Unlike Tier A and Tier B, which are based solely on statistics and semantics respectively, this goal of this tier is to identify subtle style patterns that combine lexical choice, syntax, discourse structure, and contextual dependencies in distingushing text. Hence, transformers are the best to detect if authorship signals are present in full contextual representations, even when topic and length are controlled.

Why DistilBERT?

- Provides strong contextual representations while being significantly smaller and faster than larger transformer models.
- Using a compact model reduces the risk that high accuracy arises purely from model capacity rather than actual style differences.
- A single fixed base architecture is used across all three dataset variants (Try 1, Try 2, Try 3), to ensure that performance differences reflect dataset refinement rather than changes in model size or architecture.

Why LoRA instead of full fine-tuning?

- Full fine-tuning updates all model parameters and can easily overfit when dataset size is limited.
- Low-Rank Adaptation (LoRA) freezes the pretrained transformer weights and introduces a small number of trainable parameters in the attention layers.
- This constrains learning to task-specific adjustments. This tries to ensure that the model adapts to stylistic and authorship cues rather than relearning general language knowledge.

To avoid full fine-tuning of the transformer, Low-Rank Adaptation (LoRA) is applied. LoRA injects small trainable rank-decomposition matrices into the attention projections while keeping the original pretrained weights frozen. In this setup, LoRA adapters are applied to the query and value projection layers of DistilBERT's self-attention blocks. This keeps the number of trainable parameters low while still allowing effective task adaptation.

Training setup: Separate LoRA-adapted DistilBERT models are trained for Try 1, Try 2, and Try 3. All models share the same architecture, LoRA configuration, and optimization settings. Only the training data differs so that the performance is not impacted by dataset construction.

A manual PyTorch training loop is used to maintain full control over optimization, class weighting, and evaluation. Class imbalance is once again a factor which is handled using weighted loss to prevent majority-class bias.

Tier C therefore tests whether authorship and generation signals are present in contextual representations beyond surface-level style signals (Tier A) or plain semantic embeddings (Tier B).

To verify that the models were not overfitting, performance was evaluated under multiple train–test splits. Train and test accuracies were compared, and the generalization gap was monitored across all tiers.

3.2 Theory and Background Knowledge

This task relies on three families of models that differ in how linguistic information is represented and learned: feature-based ensemble models, embedding-based neural models, and contextual transformer models.

Tree-based ensemble models

Random Forest and XGBoost are ensemble learning methods that combines multiple machine learning models to create a single, stronger predictive model. They are built from decision trees that learn a sequence of if–else rules that split the data based on feature thresholds in order to minimize classification error. While individual trees are prone to overfitting, ensembles mitigate this by combining many trees trained on different subsets of the data.

A Random Forest constructs each tree using a random subset of features and training samples, and predictions are aggregated via majority voting. This randomness decorrelates individual trees and improves generalization. Since Random Forests operate directly on explicit input features, they are well-suited for stylometric analysis, where features such as punctuation frequency or lexical ratios are meaningful on their own.

XGBoost is a gradient-boosted tree method, where trees are added sequentially to correct the errors of previous ones. While often more powerful in high-dimensional or highly complex feature spaces, gradient boosting is more sensitive to hyperparameter choices and can overfit when the feature set is small and low-dimensional.

Word embeddings and distributional semantics

Word embeddings are dense vector representations learned from large corpora based on the distributional hypothesis: words that occur in similar contexts tend to have similar meanings. GloVe (Global Vectors) embeddings are trained using global word co-occurrence statistics, resulting in a fixed vector for each word that captures semantic similarity.

When embeddings for all words in a paragraph are averaged, the paragraph is represented as a single fixed-length vector summarizing its semantic content. This averaging removes word order, syntactic structure, and discourse information, leaving only coarse-grained semantic signals. As a result, models trained on averaged embeddings cannot rely on stylistic cues such as sentence structure or punctuation patterns.

Shallow feedforward neural networks

A feedforward neural network consists of a sequence of fully connected layers that apply linear transformations followed by nonlinear activations. A network is considered shallow when it contains only one or two hidden layers. Such architectures are expressive enough to learn simple nonlinear decision boundaries, but not powerful enough to reconstruct complex structure from input representations.

When trained on averaged embeddings, a shallow feedforward network functions as a semantic classifier: it learns to separate inputs based on global semantic patterns rather than syntax or style. This makes it useful for testing whether meaning alone is sufficient for discrimination.

Contextual language models and transformers

Transformer-based models such as BERT represent text using self-attention mechanisms that compute interactions between all tokens in a sequence. Unlike static word embeddings, transformers produce contextualized representations where each token embedding depends on its surrounding words.

This allows transformers to jointly encode lexical choice, syntactic structure, and longer-range dependencies within a single representation. As a result, they can implicitly capture stylistic and authorship-related patterns that are not explicitly engineered as features.

DistilBERT

DistilBERT is a compressed version of BERT obtained through knowledge distillation, where a smaller student model is trained to reproduce the behavior of a larger teacher model. While it has fewer parameters and layers than BERT, it retains most of the representational capacity needed for downstream tasks.

From a theoretical perspective, DistilBERT offers a useful trade-off: it preserves the core transformer architecture and attention-based contextual modeling, while reducing redundancy and capacity. This makes it well-suited for controlled experiments, as strong performance is less likely to arise purely from model size and more likely to reflect genuine signal in the data.

Parameter-efficient fine-tuning with LoRA

Fine-tuning a transformer typically involves updating all model parameters, which can lead to overfitting on smaller datasets. Low-Rank Adaptation (LoRA) addresses this by freezing the pretrained model and introducing a small number of trainable parameters in the attention layers.

These additional parameters form low-rank matrices that modify attention projections, allowing the model to adapt to a new task while preserving its general language knowledge. From a theoretical standpoint, LoRA constrains learning to task-specific deviations, making it suitable for probing stylistic signals without relearning semantics.

Summary

Together, these modeling approaches reflect different assumptions about where authorship signals reside: explicit statistical features, semantic content, or contextual language usage. Comparing their performance helps isolate which level of linguistic representation contributes most strongly to distinguishing human-written and AI-generated text.

3.3 Expected Results

- **Tier A (Random Forest)** was expected to perform strongly when surface-level stylometric signals are present. Since this tier relies only on explicit statistical features, its performance was expected to decrease as prompt refinement reduces obvious stylistic shortcuts from Try 1 to Try 3.

- **Tier B (Semantic Neural Network)** was expected to show moderate performance. Because this tier operates on averaged word embeddings and discards syntax and structure, it was expected to succeed only if semantic distributions differ between human and AI text. As topic alignment improves across dataset versions, performance was expected to drop.

- **Tier C (Transformer with LoRA)** was expected to achieve the highest accuracy across all strategies. By modeling full contextual representations, this tier was expected to capture subtle authorship signals that after surface and semantic cues are minimized.

- Across all tiers, performance was expected to be highest for Try 1 and gradually decline through Try 2 and Try 3, due to prompt optimization which was aimed to reduce easy distinction between human and AI .

3.4 Results and Observations

Results are reported for all three tiers across the three dataset construction strategies. Since class imbalance exists, particular attention is paid to AI-class precision, recall, and F1 score rather than accuracy alone.

3.4.1 Tier A: Random Forest

Tier A performs strongly in Try 1 and Try 2, achieving accuracies of approximately 97.5% with high AI-class precision and recall. This confirms that early dataset versions contain strong stylometric signals that are easily captured by explicit statistical features.

In Try 3, performance drops noticeably (accuracy $\approx$ 93.5%, AI F1 $\approx$ 0.83). This decline aligns with expectations and we can prove that prompt refinement successfully reduced reliance on surface-level stylistic cues. Despite this, Tier A still performs quite well, showing that some measurable stylometric signal remains.

Overall, Tier A demonstrates that handcrafted features are powerful but brittle: they work best when differences are explicit and degrade as datasets become more realistic.

3.4.2 Tier B: Semantic Neural Network

Tier B also shows a decline across dataset versions. In Try 1, the semantic classifier performs well (accuracy $\approx$ 97%, AI F1 $\approx$ 0.93), indicating that early AI-generated text differs semantically from human text in detectable ways.

Performance drops substantially in Try 2 and Try 3, with AI recall falling to approximately 0.66 and then recovering slightly to 0.78, and overall accuracy dropping below 90%. This behavior matches expectations: as topic control improves and semantic distributions align, averaged embeddings lose discriminative power.

These results suggest that semantic content alone is not enough for robust detection once prompt design removes obvious topic and lexical biases. Tier B therefore acts as a useful probe, confirming that later dataset versions are semantically well-matched across classes.

3.4.3 Tier C: Transformer with LoRA

Tier C achieves consistently high performance across all strategies. Accuracy remains above 99% for all tries, with AI-class F1 scores exceeding 0.97 even for Try 3.

Importantly, overfitting analysis shows minimal train-test gaps across multiple splits, including a 60-40 split where test accuracy slightly exceeds training accuracy. This indicates strong generalization rather than memorization.

The sustained performance of Tier C suggests that contextual and discourse-level cues remain present even when surface-level and semantic signals are minimized. This confirms that authorship and generation signals persist in deeper language representations captured by transformer models.

3.4.4 Cross-Tier Comparison

Comparing tiers reveals a clear hierarchy:

- Tier A exploits explicit stylistic features and degrades as those features are weakened.
- Tier B relies on semantic differences, which diminish sharply with improved dataset design.
- Tier C remains robust, indicating that subtle contextual patterns survive even strong controls and highly optimized prompting techniques.

Together, these results show that as simpler signals are progressively removed only models which look deeply at the data, and not just analyse surface level cues, continue to perform well. This confirms that Try 3 is not trivially separable and is suitable for meaningful downstream analysis.

3.5 Initial Challenges

Several challenges emerged during the implementation and evaluation of Task 2:

- **Class imbalance**: The dataset contained significantly more human-written paragraphs than AI-generated ones, requiring careful handling through class-weighted loss to avoid inflated accuracy driven by majority-class predictions.
- **Metric interpretation**: High overall accuracy often masked poor AI-class recall in Tier B, making it necessary to focus on class-specific precision, recall, and F1 scores rather than accuracy alone.

- **Separating signal from capacity**: Especially in Tier C, it was initially unclear whether strong performance reflected genuine authorship signals or sheer model capacity, motivating explicit overfitting checks and controlled splits.
- **Comparability across tiers**: Ensuring fair comparison across fundamentally different model families (feature-based, embedding-based, transformer-based) required consistent data splits and evaluation protocols.
- **Debugging semantic leakage**: Unexpectedly high performance in early Tier B experiments required careful inspection to ensure that no unintended stylistic or structural indicators were leaking into semantic representations.

3.6 Limitations

- Tier-A relies on handcrafted stylometric features and may fail when such features are deliberately obfuscated or manipulated.
- Tier B collapses entire paragraphs into averaged embeddings, discarding word order and discourse structure, which limits its expressive power.
- Tier C performance depends on the pretrained language model used; results may differ for other transformer architectures or pretraining corpora.
- The binary classification setup does not capture finer distinctions between different types of AI-generated text.
- Evaluation is restricted to in-distribution data constructed using specific prompting strategies and may not generalize to unseen generation methods.
- Although overfitting was explicitly checked, extremely high accuracy in Tier C still warrants caution when extrapolating to real-world deployment.

3.7 Learning Outcomes

- Different modeling paradigms capture fundamentally different linguistic signals, and no single approach provides a complete picture.
- High accuracy alone is not enough for understanding model behavior, class-specific metrics and error analysis are essential.
- Semantic similarity does not guarantee stylistic similarity, as shown by the rapid degradation of Tier B performance.
- Transformer models can exploit subtle contextual cues even when surface-level and semantic signals are minimized.

- Careful dataset design is as important as model choice in AI detection tasks.
- Explicit overfitting analysis is critical when working with high-capacity models and small label spaces, especially when it gives high accuracy.

4 Task 3: The Smoking Gun

4.1 Implementation

The goal of Task 3 is to understand *why* the Tier C transformer-based classifier predicts a paragraph as AI-generated, rather than only reporting classification accuracy. To address this, interpretability analyses were applied to the trained DistilBERT+LoRA models across all three dataset construction strategies (Try 1, Try 2, Try 3).

The following analyses were performed:

1. **AI-isms Frequency Analysis**: Compiling a comprehensive catalog of 250+ AI-associated words, adverbs, phrases, and character names, then measuring their frequency in human vs AI text with statistical testing.
2. **SHAP Word Attribution**: Generating word-level contribution scores for representative samples to identify which tokens most influence predictions.
3. **Captum Integrated Gradients**: Computing gradient-based token attributions with respect to both classes to visualize spatial patterns of model attention.
4. **Deep Error Analysis**: Examining all 4 false positives (Human → AI) and all 7 false negatives (AI → Human) from Try 3 using gradient attribution to identify systematic failure modes.
5. **Class 2 vs Class 3 Comparison**: Evaluating whether explicit author-style mimicry (Class 3) reduces detectability compared to generic AI generation (Class 2).
6. **Aggregate Pattern Analysis**: Synthesizing findings across all errors to identify optimal thresholds, common attribution tokens, and confidence distributions.

All analyses were conducted on held-out test samples using the same trained models from Task 2. The focus was on Try 3 (author mimicry with few-shot prompting), which achieved 99.1% accuracy and provided the richest interpretability insights.

4.2 Theory and Background Knowledge

Transformer-based classifiers operate as black boxes because predictions emerge from high-dimensional contextual representations rather than explicit handcrafted features. Interpretability methods provide post-hoc explanations by estimating how input tokens influence model outputs.

Two major families of interpretability methods are relevant:

- **Gradient-based methods** (e.g., Integrated Gradients via Captum): Measure sensitivity by computing gradients of output logits with respect to input embeddings. These methods exploit the differentiability of neural networks to identify which tokens most strongly influence predictions.
- **Perturbation-based methods** (e.g., SHAP): Measure importance by systematically masking input features and observing prediction changes. SHAP provides model-agnostic explanations based on game-theoretic Shapley values.

Because transformer representations are contextual, attribution scores do not necessarily correspond to isolated keywords. Instead, they often reflect:

- **Distributional patterns**: How frequently certain words co-occur
- **Stylistic regularities**: Sentence rhythm, punctuation usage, register consistency
- **Structural cues**: Repetitive phrasing, parallel structures, discourse markers

In AI text detection specifically, interpretability helps distinguish whether models rely on:

- Explicit lexical markers ("AI-isms" like `delve`, `tapestry`)
- Broader stylistic patterns (formal register, uniform structure)
- Topic-based shortcuts (19th century content vs modern concepts)

Important Caveat: Attribution provides *correlational, not causal* explanations. High attribution indicates sensitivity, not necessity. A word could be highly weighted due to correlations with true causal factors. Despite this limitation, attribution remains valuable for diagnosing detector behavior and identifying shortcuts.

4.3 Expected Results

Based on the detectors' strong performance in Task 2 (84–99% accuracy), several outcomes were anticipated:

1. **AI-isms frequency**: AI-generated text should exhibit significantly higher frequencies of known AI-associated vocabulary compared to human text, with statistical significance and medium-to-large effect sizes (Cohen's $d > 0.5$).
2. **Attribution patterns**:
   - For correctly classified AI paragraphs, attribution should highlight both AI-isms and structural markers
   - For human text, attribution should be more diffuse and context-dependent
   - Top-attributed words should overlap substantially with the AI-isms catalog
3. **Class 3 mimicry effectiveness**: Style prompting should reduce AI-isms frequency and lower detection confidence compared to Class 2, potentially dropping detection rates by 10–20 percentage points.
4. **False positives**: Human paragraphs misclassified as AI should exhibit:
   - Elevated AI-isms counts (approaching or exceeding AI mean)
   - High structural regularity resembling AI outputs
   - Formal register with abstract vocabulary
5. **False negatives**: AI paragraphs misclassified as human should show:
   - Near-zero AI-isms density
   - Greater lexical diversity approaching human baselines
   - Informal tone with contractions and colloquialisms
6. **Confidence as reliability indicator**: Errors should show substantially lower confidence than correct predictions, suggesting the model recognizes uncertainty.

4.4 Results

4.4.1 AI-isms Frequency Analysis

A comprehensive catalog of 250+ words, adverbs, phrases, and character names was compiled from prior research and manual inspection. Quantitative analysis revealed strong discriminative power across all three generation strategies:

Try 1 (Topics–Lenses–Constraints Prompting):

- Human text: Mean = 1.92 ± 1.86 AI-isms per paragraph
- AI text (combined Class 2 + Class 3): Mean = 3.54 ± 2.36
- Statistical comparison: $p < 0.001$, Cohen's $d = 0.76$ (medium-to-large effect)

- Frequency ratio: 1.8× more AI-isms in AI text

Try 2 (Story Generation + Segmentation):

- Human text: Mean = 1.92 ± 1.86 AI-isms per paragraph
- AI text: Mean = 3.71 ± 2.54
- Cohen's $d = 0.80$ (large effect, strongest of three tries)
- Frequency ratio: 1.9×

Try 3 (Few-Shot Author Mimicry):

- Human text: Mean = 1.92 ± 1.86 AI-isms per paragraph
- AI text: Mean = 3.18 ± 2.23
- Cohen's $d = 0.61$ (medium effect)
- Frequency ratio: 1.7×

All three comparisons achieved extreme statistical significance ($p < 10^{-19}$), confirming AI-isms as robust, non-random markers of AI authorship.

Key Findings:

1. AI-isms are highly discriminative across all dataset versions
2. Try 2 showed the strongest effect size (0.80), consistent with its near-perfect 99.4% accuracy
3. Try 3 showed reduced AI-isms density compared to Try 1/Try 2, suggesting few-shot prompting successfully suppresses some stereotypical patterns
4. Human text maintains consistent AI-isms levels (1.92) across all tries, validating the catalog's stability

4.4.2 Class 2 vs Class 3 Comparison

Comparing generic AI generation (Class 2) with author-mimicry prompting (Class 3):

Try 1:

- Class 2 detection rate: 22.2%
- Class 3 detection rate: 16.1%
- Evasion benefit: −6.1 percentage points
- AI-isms reduction: 3.92 → 3.16 (−0.76 items)

Try 2:

- Class 2 detection rate: 100%

- Class 3 detection rate: 100%
- Evasion benefit: None (perfect detection on both)
- AI-isms present but insufficient to evade robust detector

Try 3:

- Class 2 detection rate: 100%
- Class 3 detection rate: 94.4%
- Evasion benefit: −5.6 percentage points (7 false negatives from Class 3)
- AI-isms reduction: 3.63 → 2.73 (−0.91 items)

Interpretation: Style mimicry provides *statistically significant but practically limited* evasion. It successfully reduces AI-isms frequency by 0.76–0.91 items per paragraph, demonstrating that prompt engineering can suppress stereotypical patterns. However, evasion only occurs when the base detector is weak (Try 1 at 22% detection) or near-perfect (Try 3 with 7/180 FNs). Against the robust Try 2 detector, mimicry offers no protection.

This finding contradicts the hypothesis that style transfer fundamentally alters detectability—it only provides marginal benefits in edge cases.

4.4.3 SHAP Word Attribution Analysis

SHAP analysis on Try 3 revealed surprising patterns in model decision-making:

Top Contributing Words (highest SHAP values):

- Contextual/narrative: "baker's", "wife—a", "bread", "woman", "voice", "hearth"
- Structural markers: "first", "to", "wasn't", "still", "entirely"
- Period-specific: "known", "heard"

Critical Finding: *None* of these top SHAP words appeared in the AI-isms catalog. This reveals a fundamental gap between human-interpretable markers ("delve", "tapestry", "nuance") and the statistical patterns the model actually exploits.

Interpretation: The detector learns *distributional regularities and contextual patterns* rather than simply flagging suspicious vocabulary. Words like "baker's" and "hearth" are not inherently AI-like, but their co-occurrence patterns, positional distributions, and surrounding context differ systematically between human and AI text.

This explains why:

- The AI-isms catalog is statistically valid (Cohen's d = 0.61–0.80) but incomplete

- Simple word-counting rules achieve only 80.8% accuracy vs 99.1% for neural models
- Adversarial optimization (Task 4) requires more than avoiding catalog words

4.4.4 Captum Integrated Gradients Analysis

Gradient-based attribution provided complementary insights:

High-Attribution Tokens:

- **Character names**: “eleanor”, “marianne”, “beatrice”, “milton”, “thornton”
- **Formal British spellings**: “colour”, “honour”, “endeavour”
- **Polite register markers**: “therefore”, “frequently”, “subjected”
- **Punctuation**: Semicolons (;) received strong positive attribution

Interpretation: The model detects *register mismatch*—AI struggles to consistently maintain formal 19th-century British voice across entire paragraphs. Character names and period-appropriate vocabulary signal authenticity, while modern phrasing or inconsistent register triggers suspicion.

Notably, Captum identified *different* salient features than SHAP:

- SHAP: Contextual words and structural markers
- Captum: Character names and formal vocabulary

This divergence confirms that different attribution methods expose complementary aspects of model behavior. Using multiple techniques provides more robust insights than any single approach.

4.4.5 Deep Error Analysis: All 11 Misclassifications

Try 3 achieved 99.1% accuracy with 11 errors: 4 false positives and 7 false negatives. Gradient attribution was computed on *every single error* to identify systematic failure modes.

False Positives: When Human Writing Appears AI-Generated **Profile**:

- Total: 4 human passages misclassified as AI
- Average confidence: 57.7% (vs 99.1% for correct human predictions)
- Confidence gap: 41.4 percentage points lower than correct predictions

The AI-isms Paradox:

- False positives: Mean = 1.25 ± 0.83 AI-isms
- Correctly classified human: Mean = 1.95 ± 1.32 AI-isms
- **FPs contain fewer AI-isms than correct human predictions**

This contradicts the catalog hypothesis—false positives are *not* triggered by excessive AI vocabulary. Instead, other features cause misclassification.

Attribution Analysis (gradient-based):

Top tokens pushing toward AI classification:

- **Punctuation**: Semicolons (;) appeared as high-attribution tokens in all 4 FPs, suggesting the model associates formal punctuation with AI writing
- **Character names**: "willoughby", "marianne", "milton", "thornton" triggered false alarms despite being authentic Austen/Gaskell characters
- **Formal vocabulary**: "competence", "realised", "militia" contributed to misclassification

AI-isms Present: Minimal catalog words found (5 total across 4 samples):

- Words: "heart", "very"
- Adverbs: "gently", "slowly", "together"
- Primarily subtle adverbs (3/5) rather than distinctive AI markers

Root Cause: The model conflates *formal 19th-century register* (complex punctuation, character-driven narrative) with AI's attempt to emulate formality. Human passages with proper names and semicolons get penalized despite low AI-isms counts.

False Negatives: When AI Successfully Passes as Human Profile:

- Total: 7 AI passages evaded detection
- Average confidence: 65.1% (vs 98.1% for correct AI predictions)
- Confidence gap: 33.0 percentage points lower than correct predictions

AI-isms Evasion Success:

- False negatives: Mean = 2.00 ± 1.20 AI-isms
- Correctly classified AI: Mean = 3.18 ± 2.23 AI-isms
- **FNs contain 37% fewer AI-isms than correct AI predictions**
- One sample achieved perfect evasion (0 AI-isms, Sample 1082)
- Range: 0–4 AI-isms per paragraph

Attribution Analysis (gradient-based):
Top tokens pushing toward Human classification:

- **Character names as camouflage**: "eleanor", "martha", "charlotte", "marianne" pushed predictions toward human—authentic 19th-century names serve as protective cover
- **Authentic period vocabulary**: "pottery", "mill", "vicar", "manchester", "croft" signal genuine historical context
- **Conversational markers**: "aye", "remarked" (2 occurrences), exclamation points suggest natural dialogue

AI-isms Present: 14 catalog items across 7 samples:

- Words: 8 occurrences ("heart", "remarkable")
- Adverbs: 6 occurrences ("very", "really", "quickly", "firmly", "truly", "perfectly")
- Phrases: 0 occurrences
- Character names (from catalog): 0 occurrences

Most common: subtle adverbs ("very", "really") rather than distinctive markers ("delve", "tapestry").

Root Cause: AI text that *stays below 2 AI-isms* and incorporates authentic period names/vocabulary can evade detection. The catalog words present are common intensifiers rather than stereotypical AI patterns. Including authentic 19th-century character names in prompts provides strong protective cover.

4.4.6 Aggregate Pattern Analysis Across All Errors

Confidence as Reliability Indicator:

- False positives: 57.7% average confidence
- False negatives: 65.1% average confidence
- All errors combined: 62.4% average confidence
- Correct predictions: 98.6% average confidence
- **Confidence gap: 36.2 percentage points**

The model shows dramatically lower confidence on errors, suggesting it recognizes ambiguous cases but lacks decisiveness. Errors cluster in the 55–75% confidence range, while correct predictions dominate the 95–100% range.

Optimal AI-isms Threshold:
Using AI-isms count alone as a classification rule:

- Optimal threshold: 5.5 AI-isms

- Achievable accuracy: 80.8%
- Neural model accuracy: 99.1%
- **Accuracy gap: 18.3 percentage points**

This reveals:

- The catalog captures broad patterns but misses distributional nuances
- The 18.3pp gain comes from learning punctuation patterns, character name distributions, and register consistency beyond vocabulary
- AI-isms counts serve as interpretable features for humans but cannot replace neural models

Common Attribution Tokens in Errors:
False Positives (most frequent top-5 tokens):

- ";" (semicolon) — appeared in 4/4 FPs [NOT in catalog]
- "willoughby", "marianne" — character names [NOT in catalog]
- "competence", "realised" — formal vocabulary [NOT in catalog]

False Negatives (most frequent top-5 tokens):

- "martha" — appeared in 3/7 FNs as protective name [NOT in catalog]
- "remarked" — appeared in 2/7 FNs [NOT in catalog]
- "pottery", "mill", "vicar" — period-specific vocabulary [NOT in catalog]

Critical Insight: The tokens causing errors are *almost entirely absent from the AI-isms catalog.* The model's failure modes operate on different features than the human-interpretable markers we identified.

Systematic Failure Modes:

1. **FPs caused by formality**: Semicolons, proper names, complex sentences trigger false alarms on human text
2. **FNs enabled by minimalism**: AI writing with < 2 AI-isms and authentic period vocabulary evades detection
3. **Confidence as proxy**: Errors consistently show 30–40pp lower confidence
4. **Name confusion**: Character names serve dual roles—they trigger FPs when in human text, provide cover when in AI text

4.5 Error Analysis

Several findings contradicted initial hypotheses:

1. Top SHAP Words Don't Match AI-isms Catalog **Expected**: SHAP analysis would highlight words like "tapestry", "delve", "nuance" from the AI-isms catalog.

Actual: Top SHAP words were contextual ("baker's", "hearth") and structural ("first", "wasn't"), with *zero overlap* with the catalog.

Why: The AI-isms catalog identifies human-interpretable markers through frequency analysis, but the neural model learns deeper distributional patterns. Words like "baker's" aren't inherently AI-like, but their positional distributions and co-occurrence patterns differ systematically. The model exploits statistical regularities invisible to human analysts.

2. False Positives Have FEWER AI-isms Than Correct Predictions **Expected**: FPs would show elevated AI-isms counts, explaining why human text was misclassified.

Actual: FPs averaged 1.25 AI-isms vs 1.95 for correctly classified human text—35% *fewer* markers.

Why: Misclassification is driven by *register and structure*, not vocabulary. Human passages with formal punctuation (semicolons), proper names, and complex sentences resemble AI's attempt to emulate 19th-century style. The model conflates "polished formality" with "AI generation", causing it to penalize stylistically sophisticated human writing.

3. Style Mimicry (Class 3) Provides Minimal Evasion **Expected**: Explicit author mimicry would reduce detection by 10–20pp.

Actual: Class 3 evasion ranged from 0pp (Try 2) to 6.1pp (Try 1) to 5.6pp (Try 3).

Why: Style mimicry successfully reduces AI-isms frequency (0.76–0.91 fewer items), but the detector doesn't rely primarily on these markers. It exploits distributional patterns and structural regularities that simple style prompting cannot address. Only systematic adversarial optimization (Task 4) can achieve substantial evasion.

4. Character Names Cause Confusion **Expected**: Character names would be neutral features, neither helping nor hurting.

Actual: Names like "willoughby" triggered FPs on human text, while "martha" provided cover for AI text.

Why: The model learned correlations between specific character names and AI/human classes rather than understanding that names are topic-specific, not authorship-specific. This reflects *spurious correlation*—the training data confounded character identity with text source. Names from Class 3 prompts (which explicitly mention characters) became associated with AI, while human excerpts' names became human markers.

5. Confidence Gap Reveals Model Uncertainty Expected: Errors might occur at moderate confidence (60–80%), with some high-confidence mistakes.

Actual: Errors averaged 62.4% confidence vs 98.6% for correct predictions—a 36.2pp gap. Only 11/1264 high-confidence ($> 95\%$) predictions were wrong.

Why: The model learns not just classification but also *epistemic uncertainty*. It recognizes when samples fall near the decision boundary and reduces confidence accordingly. This suggests the model generalizes well and doesn't overfit to spurious patterns—it "knows what it doesn't know".

4.6 Initial Challenges

1. Different attribution methods (SHAP vs Captum) highlighted different features, requiring convergent evidence from multiple techniques
2. DistilBERT's WordPiece tokenizer fragments words unpredictably, requiring manual aggregation of subword attribution scores
3. SHAP and Integrated Gradients are computationally expensive (~30 seconds per sample), limiting analysis to representative subsets
4. High attribution indicates sensitivity not causation; ablation studies would establish necessity but were not performed

4.7 Limitations

1. Attribution is correlational not causal; ablation studies needed to establish necessity were not performed
2. Findings are method-dependent (SHAP vs Captum) and model-specific (DistilBERT+LoRA); limited sample size and hindsight bias may influence interpretation
3. Topic confounding: detectors may exploit 19th-century content rather than authorship (99% → 75% accuracy drop on modern topics)
4. Temporal validity: findings reflect 2024-era models and may not generalize to future systems

4.8 Learning Outcomes

1. Detectors learn distributional patterns beyond human-interpretable markers; top SHAP words ("baker's", "hearth") don't overlap with AI-isms catalog ("delve", "tapestry")
2. Systematic error analysis revealed exploitable patterns: semicolons trigger FPs (4/4), character names enable FN evasion ("martha" 3/7), AI with < 2 AI-isms evades detection; 36.2pp confidence gap between errors and correct predictions

3. Style mimicry (Class 3) reduces AI-isms by 0.76–0.91 but achieves only 0–6.1pp evasion; adversarial optimization required for substantial evasion

4. Topic confounding threatens generalization: detector relies on 19th-century context rather than authorship

5 Task 4: The Turing Test - Evolving AI Text to Fool Detectors

5.1 Implementation

The goal of Task 4 was to implement a **Genetic Algorithm (GA)** to evolve AI-generated text until it could fool the Tier C detector developed in previous tasks. The hypothesis was that through iterative mutation and selection, AI-generated paragraphs could be optimized to achieve $> 90\%$ human confidence scores.

5.1.1 Technical Architecture

Ensemble Detector Configuration Rather than relying on a single detector, I used an **ensemble voting approach** combining all three detector variants from Task 2:

- **Try 1**: Trained on Class 1 (human) vs Class 2 (AI-neutral)
- **Try 2**: Trained on Class 1 (human) vs Class 3 (AI-styled)
- **Try 3**: Trained on all three classes with style-transfer awareness

Rationale: Try 3 was identified as the hardest to fool (highest validation accuracy, explicit style mimicry training). By using ensemble voting (averaging probabilities across all three models), the fitness function provides a more robust signal that cannot be exploited by targeting quirks of a single detector.

The fitness function is defined as:

$$\text{fitness}(text) = \frac{1}{3}\sum_{i=1}^{3} P^{(i)}_{\text{human}}(text)$$

where $P^{(i)}_{\text{human}}(text)$ is the probability that detector i classifies the text as human-written.

Generation Model: Gemma-3-27B Following the infrastructure established in Task 0, I used Google's **Gemma-3-27B-IT** model via the GenAI API for both:

1. **Initial population generation**: Creating diverse starting paragraphs

2. **Mutation operator**: Implementing human-like text perturbations

Why Gemma instead of local models?

- Consistency with Task 0's generation pipeline
- High-quality instruction following for targeted mutations
- Free tier sufficient for experimentation
- No local GPU infrastructure needed

5.1.2 Genetic Algorithm Design

To prevent premature convergence and maintain population diversity, I implemented **MD5 hash-based duplicate detection**. Each paragraph is hashed before being added to the population; duplicates are rejected and regeneration is triggered. This prevents the GA from converging on repeated variants of the same solution.

Population Initialization Started with 20 diverse AI-generated paragraphs on the topic "social class and morality in 19th century England", using varied generation prompts:

- Casual, conversational style with contractions and informal language
- Explaining to a friend over coffee with hedging phrases ("I think", "maybe")
- Slightly rambling, natural style with dramatic sentence length variation
- 19th century essay style with semicolons and em-dashes
- Mixing formal and casual language unpredictably
- Narrative storytelling approach with specific examples
- Strong opinions and personality with genuine emotion
- Thinking out loud with self-corrections and uncertain phrases
- Varied rhythm with short punchy sentences mixed with long complex ones

This diversity ensures broad exploration of the initial solution space. High temperature (1.4) was used for generation to encourage stylistic diversity.

Selection Strategy: Elitism Each generation:

1. Score all paragraphs using ensemble fitness function
2. Select top 3 performers (highest human probability)
3. Keep elite individuals in next generation (elitism)
4. Generate mutations and crossover offspring

Mutation Operators: LLM-as-Mutator Rather than character-level or word-swap mutations (which produce unnatural text), I used **Gemma as a semantic mutation operator**. **Twenty distinct mutation types** were defined across five categories:

Rhythm Mutations:

- Rewrite to dramatically vary sentence lengths (some 3-5 words, one 30+ words)
- Break up long sentences and combine short ones

Punctuation Mutations:

- Add semicolons to connect related ideas
- Add em-dashes for parenthetical asides or emphasis

Imperfection Mutations (Critical):

- Start sentences with “And” or “But”, add parenthetical thoughts, use contractions
- Add hedging language (“I think”, “maybe”, “sort of”, “seems like”, “probably”)
- Add deliberate grammatical quirks (sentence fragments, dangling modifiers, informal structure)

Vocabulary Mutations:

- Replace formal words with casual equivalents (“utilize” → “use”)
- Add archaic or rare words naturally (“wherefore”, “erstwhile”, “albeit”)
- Use more specific, concrete nouns instead of abstract ones (“technology” → “algorithms”)

Tone Mutations:

- Rewrite more casually and conversationally (like talking to a friend)
- Mix formal and informal language unpredictably
- Add personality and opinions (“honestly”, “to be fair”, “surprisingly”)

Structure Mutations:

- Change order of ideas (conclusion first, then reasoning)
- Add redundancy (repeat ideas in different words, as humans naturally do)
- Add sentences that wander or trail off before making their point

Specificity Mutations:

- Add specific examples, anecdotes, or concrete details instead of general statements
- Replace abstract nouns with tangible, visualizable alternatives

Meta-Awareness Mutations:

- Add self-aware phrases ("if that makes sense", "you know what I mean", "at least that's how I see it")
- Remove perfect transitions ("Furthermore", "Moreover") and jump between ideas naturally

Rationale: These 20+ mutations target features that distinguish human writing from typical AI-generated text: natural rhythm variation, subtle imperfections, conversational elements, genuine redundancy, and self-awareness. The diversity of mutation types prevents the GA from exploiting a single feature.

Crossover Operator Sentence-level crossover was implemented to combine successful paragraphs:

1. Split two high-fitness parents into sentences
2. Randomly select sentences from each parent
3. Combine selected sentences into offspring
4. Use Gemma to smooth transitions without changing content

Adaptive Mechanisms

- **Duplicate detection**: Hash-based detection rejects duplicates to maintain diversity
- **Adaptive mutation strength**: Increases mutation intensity if fitness plateaus
- **Variable temperature**: Ranges from 1.1 (weak) to 1.5 (strong) based on mutation strength

Mutation Strength and Temperature Control Mutation operators use variable temperature settings based on **adaptive mutation strength**:

- **Weak mutations**: Temperature 1.1 (minor perturbations, preserve most structure)
- **Medium mutations**: Temperature 1.3 (significant changes)
- **Strong mutations**: Temperature 1.5 (aggressive rewrites)

The GA monitors fitness improvement between generations. If fitness plateaus (no improvement for 2+ generations), mutation strength increases automatically to escape local optima. This adaptive mechanism proved critical for continued progress in generations 10-15.

5.1.3 Quantitative Evolution Metrics

Detailed generation-by-generation progression:

Generation	Best Fitness	Elite Mean Fitness
0	0.132 (13.2%)	0.062 (6.2%)
5	0.359 (35.9%)	0.311 (31.1%)
10	0.786 (78.6%)	0.717 (71.7%)
13	0.880 (88.0%)	0.839 (83.9%)
16	0.900 (90.0%)	0.890 (89.0%)
18	**0.920 (92.0%)**	0.903 (90.3%)

Key observations:

- Initial population mean was only 6.2% human (highly detectable)
- By generation 18, even rank-3 elite specimens scored >89% human
- The fitness improvement accelerated in generations 6-12 (rapid phase)
- Goal achievement (90%) occurred in generation 16; continued to 18 for robustness

Termination Criteria The GA runs for up to 20 generations or until a paragraph achieves $> 90\%$ human confidence (fitness > 0.90).

5.1.4 Implementation Details

- **Device**: CUDA GPU (for detector inference)
- **API Rate Limiting**: 0.4-0.5s delay between Gemma calls to respect API quotas
- **Tokenization**: DistilBERT tokenizer with 512 token max length
- **Random Seeds**: Fixed at 42 for reproducibility of selection/crossover
- **Mutation Rate**: 70%
- **Crossover Rate**: 30%

5.2 Results

5.2.1 Generation 0: Baseline Performance

Initial population statistics:

- **Best Human Probability**: 0.132 (13.2%)
- **Mean Human Probability**: Approximately 10%

- **Note**: Despite varied prompting strategies, initial AI-generated text was strongly detected

This confirms the baseline assumption: raw AI-generated text is reliably detected even when prompted for diversity.

5.2.2 GA Convergence

The genetic algorithm successfully evolved text to fool the detector:

- **Generation 0**: Best score = 0.132 (13.2% human)
- **Generation 5**: Best score = 0.359 (35.9% human)
- **Generation 10**: Best score = 0.786 (78.6% human)
- **Generation 18**: Best score = 0.920 (92.0% human) **Goal Reached**

Fitness improvement: 78.8 percentage points over 18 generations.

Convergence Pattern The evolution showed three phases:

1. **Generations 0-5**: Exploration phase with gradual improvement (13.2% → 35.9%)
2. **Generations 6-12**: Rapid improvement as successful mutations accumulated (35.9% → 85%)
3. **Generations 13-18**: Fine-tuning phase approaching goal (85% → 92.0%)

5.3 Validation Testing: Personal Writing

5.3.1 Motivation for Personal Testing

After successfully reaching 92% human confidence with evolved text, I tested the detector on writing with absolute ground truth: my own human-written Statement of Purpose for graduate school applications.

5.3.2 Result

My Human-Written SOP: 0.86% human (99.14% AI)

The detector classified authentic human writing with higher AI confidence than most actual AI-generated paragraphs in the training set.

5.4 Systematic Investigation: Out-of-Distribution Testing

5.4.1 Experiment 1: Manual Humanization

Method I manually edited my SOP to include features typically associated with casual human writing:

- Added contractions: "I have" → "I've", "cannot" → "can't"
- Inserted hedging phrases: "perhaps", "I think", "maybe"
- Shortened sentences
- Added casual transitions: "Also", "Plus"

Result Edited SOP: 0.8% human (-0.06pp change)

The modifications produced no improvement. The detector remained 99.2% confident the text was AI-generated.

Interpretation The detector does not primarily rely on:

- Contractions vs. formal language
- Sentence length variability
- Hedging or conversational markers

5.4.2 Experiment 2: Austen-Style Rewrite

Method I rewrote my modern SOP using 19th-century vocabulary and Jane Austen's stylistic conventions:

> *"It is a truth universally acknowledged, that a young scholar possessed of curiosity about linguistic patterns must be in want of a suitable position in a laboratory of research. I have long harboured an interest in the manner by which patterns emerge in language..."*

Result Austen-Style SOP (modern NLP topic): 1.3% human (+0.44pp improvement)

Even with Victorian vocabulary and 19th-century phrasing, the modern NLP topic domain remained strongly classified as AI (98.7%).

Additional Test: Pure 19th-Century Context I wrote a completely different SOP as a 19th-century governess applying for a teaching position, with period-appropriate topics:

Pure 19th-Century Governess SOP: 8.6% human (+7.7pp improvement)

This scored better but still registered as 91.4% AI.

5.4.3 Experiment 3: Out-of-Vocabulary Testing

To systematically test detector behavior, I evaluated paragraphs from diverse domains guaranteed to be human-written:

Domain	Human Probability
Wikipedia (Bees)	1.9%
Reddit Gaming Discussion	8.8%
Cooking Blog	0.8%
Academic (Climate)	0.9%

All modern human writing scored below 9% human classification.

Contrast with 19th-Century Human Text As a control, I tested an actual opening paragraph from *Pride and Prejudice*:
Jane Austen (1813): 65.6% human

5.5 The Definitive Test: Baby GPT

5.5.1 Motivation

Following analysis of the pattern where 19th-century style correlated with higher human scores regardless of actual authorship, I tested an AI model trained exclusively on Jane Austen novels (Baby GPT).

5.5.2 Hypothesis

If the detector learned era/genre rather than authorship, Baby GPT (AI trained on Austen corpus) should score higher on human classification than modern human writing.

5.5.3 Result

Baby GPT (AI trained on Austen): 99.5% human

This exceeds:

- My human writing (0.9%)
- Modern Wikipedia articles (1.9%)
- Cooking blog (0.8%)
- Academic writing (0.9%)
- Actual Jane Austen text (65.6%)
- The GA-evolved text (92.0%)

5.6 Root Cause Analysis

5.6.1 Training Data Confounding

The fundamental issue traces to dataset design in Task 0. The training data confounded multiple variables:

1. **Historical era with authorship**:
   - Human data: 19th-century novels (Austen, Gaskell)
   - AI data: Modern AI models generating contemporary prose
2. **Literary genre with cognitive origin**:
   - Human data: Fiction, narrative, domestic themes
   - AI data: Often more expository or neutral when not style-prompted
3. **Vocabulary distribution with authorship signal**:
   - Human data: “whilst”, “ought”, “hitherto”, archaic pronouns
   - AI data: “however”, “moreover”, “can’t”, modern contractions

5.6.2 What the Model Actually Learned

The detector learned a valid pattern from the training distribution:

$$P(\text{Human} \mid \text{Text}) \approx P(\text{19th-century vocabulary} \mid \text{Text})$$

This achieves high validation accuracy on the training distribution because all human examples were 19th-century and all AI examples were modern. However, this pattern fails to generalize to:

- Modern human writing (21st century)
- AI trained on historical corpora (Baby GPT)
- Cross-era authorship detection

5.6.3 Why Validation Accuracy Was Misleading

Task 2 achieved 99% validation accuracy. This accuracy measured:

“Can the model distinguish Task 0’s Class 1 from Class 2/3?”

Not:

“Can the model distinguish human authorship from AI generation in general?”

The validation set came from the same confounded distribution, so high accuracy did not guarantee genuine authorship detection capability.

5.7 Why the GA Succeeded

5.7.1 What the GA Optimized

The genetic algorithm successfully increased fitness from 13.2% to 92.0% by discovering that the detector rewarded:

- Archaic vocabulary: "whilst", "endeavour", "hitherto"
- Formal Victorian prose style
- Literary register with subordinate clauses
- 19th-century phrasing conventions

5.7.2 The GA as a Diagnostic Tool

The GA functioned as an adversarial probe that revealed the detector's true decision boundary. By optimizing text to maximize human probability, it effectively performed gradient ascent on the detector's learned features, exposing that those features were era/genre markers rather than authorship signals.

The shortest path to high fitness was "add Victorian vocabulary", not "write more authentically human". This diagnostic power demonstrates what adversarial testing is designed to achieve: revealing what a model truly learned, not what we intended it to learn.

5.8 Error Pattern Analysis

5.8.1 Systematic Failure Modes

False Positives: Modern Human Writing Classified as AI Any text with modern vocabulary, even if genuinely human-written, scored below 10% human:

- Personal SOP: 0.9%
- Wikipedia: 1.9%
- Cooking blog: 0.8%
- Academic: 0.9%
- Reddit Gaming: 8.8%

Root cause: Model learned "modern vocabulary = AI" because training AI data was modern.

False Negatives: AI Text Classified as Human AI text with 19th-century vocabulary scored above 90% human:

- Baby GPT: 99.5%
- GA-evolved text: 92.0%

Root cause: Model learned "19th-century vocabulary = human" because training human data was 19th-century.

5.8.2 Baby GPT Scores Higher Than Actual Austen

The most definitive evidence:

$$\text{Baby GPT (99.5\%)} > \text{Actual Jane Austen (65.6\%)}$$

Explanation

1. Baby GPT was trained on multiple Austen novels
2. It learned the average statistical properties across all novels
3. Its generated text is a concentrated distillation of Austen's most frequent patterns
4. Actual Austen includes more variation, rare constructions, and idiosyncrasies
5. The detector learned to recognize prototypical Austen style (which Baby GPT exemplifies) rather than Austen's full range

This mirrors how generative models can produce more prototypical examples than real samples from the training distribution.

5.9 Comparison with Task 3 Findings

5.9.1 Task 3: Style Mimicry Provides Minimal Evasion

Task 3 found that Class 3 (AI-styled text) only reduced detection by 0-6.1pp compared to Class 2 (AI-neutral):

- Try 1: +6.1pp evasion
- Try 2: 0pp evasion
- Try 3: +5.6pp evasion

5.9.2 Task 4: GA Achieves 78.8pp Evasion

By contrast, Task 4's GA evolved text from 13.2% to 92.0% human (+78.8pp).

Explanation: The GA succeeded because it discovered the detector's true decision boundary (era/genre) and explicitly optimized for it through iterative refinement. Class 3 style prompting was only partially effective because Gemma's base training is modern, so "write like Austen" produces a modern interpretation of Victorian style rather than concentrated 19th-century patterns.

5.10 Lessons Learned and Future Improvements

5.10.1 What Task 4 Demonstrated

1. Adversarial Evaluation is Necessary Standard validation metrics (accuracy, F1, AUC) can be misleading if the validation set shares confounds with training data. Adversarial testing revealed true model limitations:

- Test on out-of-distribution data (modern vs. historical)
- Test on diverse human sources (personal writing, Wikipedia, Reddit)
- Test on diverse AI sources (different models, training corpora)

2. Dataset Confounding Has Measurable Consequences A model cannot learn to distinguish X from Y if all X examples share property A and all Y examples share property B. The model will learn to distinguish A from B instead. In this case, "human vs. AI" was confounded with "19th-century vs. modern vocabulary".

3. High Validation Accuracy Does Not Guarantee Robust Learning 99% accuracy on validation set did not prevent 0.9% human score on actual modern human writing. Validation accuracy measured distinguishing training classes, not detecting authorship.

4. GAs Are Diagnostic Tools Beyond evolving adversarial text, the GA served as a diagnostic by revealing the detector's decision boundary through optimization. The evolved text provided interpretable evidence of what features the detector relies on.

5. Personal Writing Tests Are Valuable Testing my own SOP provided absolute ground truth and clear falsification of the hypothesis that the detector works robustly across domains.

5.10.2 How to Fix the Detector

Dataset Redesign Diversify human data across eras and genres:

- 19th century: Austen, Gaskell, Dickens

- 20th century: Hemingway, Woolf, Baldwin
- 21st century: Modern authors, blog posts, essays
- Multiple genres: fiction, non-fiction, academic, casual

Diversify AI data to match:

- AI trained on 19th-century corpus
- AI trained on modern corpus
- AI with varied prompts (neutral, style-mimicking, adversarial)

Critical principle: For every era/genre in human data, include corresponding AI data to break the confound.

Model Architecture Multi-task learning:

- Primary task: Human vs. AI classification
- Auxiliary tasks: Era prediction, genre prediction
- Encourage model to disentangle these factors

Domain adaptation:

- Train on multiple domains (historical + modern)
- Use domain-adversarial training to learn domain-invariant features

Evaluation Protocol Out-of-distribution testing:

- Test set must include unseen eras, genres, authors
- Include personal writing from team members
- Include AI outputs from models trained on diverse corpora
- Report performance breakdowns by domain

Adversarial robustness testing:

- Run GA optimization to find adversarial examples
- Test on Baby GPT-style models
- Analyze failure modes with systematic error patterns
- Require model to maintain performance on adversarial probes

5.11 Conclusion

Task 4 implemented a genetic algorithm that successfully evolved AI-generated text from 13.2% to 92.0% human confidence across 18 generations, achieving the goal of fooling the detector.

However, systematic testing revealed a fundamental limitation: the detector classifies all modern human writing as AI (scores below 9%) while classifying AI trained on historical corpora as human (Baby GPT: 99.5%). This demonstrates that the detector learned to distinguish 19th-century vocabulary from modern vocabulary, not human authorship from AI generation.

This is a scientifically valuable result. It demonstrates:

1. Why dataset confounding matters in practice
2. How models can achieve high validation accuracy while learning incorrect concepts
3. The necessity of adversarial evaluation beyond standard test sets
4. That AI text detection requires solving distribution shift, not just classification

The genetic algorithm succeeded in revealing the detector's limitations rather than demonstrating its robustness. Future work must include diverse human data across eras and genres, match AI data diversity to human data, test on out-of-distribution samples, and use adversarial optimization as a diagnostic tool.

Task 0 warned about confounding variables. Task 4 demonstrated the consequences when that warning is not fully addressed in dataset construction.