Tuning for TraceTarnish: Techniques, Trends, and Testing Tangible Traits

Robert Dilworth ▨ ▩ ▢ ◈ ◈ ▣

Department of Computer Science and Engineering, Mississippi State University,
Mississippi State, Mississippi, USA
`rkd103@msstate.edu`

Abstract. In this study, we more rigorously evaluated our attack script *TraceTarnish*, which leverages adversarial stylometry principles to anonymize the authorship of text-based messages. To ensure the efficacy and utility of our attack, we sourced, processed, and analyzed Reddit comments–comments that were later alchemized into *TraceTarnish* data–to gain valuable insights. The transformed *TraceTarnish* data was then further augmented by *StyloMetrix* to manufacture stylometric features–features that were culled using the Information Gain criterion, leaving only the most informative, predictive, and discriminative ones. Our results found that function words and function word types (L_FUNC_A & L_FUNC_T); content words and content word types (L_CONT_A & L_CONT_T); and the Type-Token Ratio (ST_TYPE_TOKEN_RATIO_LEMMAS) yielded significant Information-Gain readings. The identified stylometric cues–function-word frequencies, content-word distributions, and the Type-Token Ratio–serve as reliable indicators of compromise (IoCs), revealing when a text has been deliberately altered to mask its true author. Similarly, these features could function as forensic beacons, alerting defenders to the presence of an adversarial stylometry attack; granted, in the absence of the original message, this signal may go largely unnoticed, as it appears to depend on a pre- and post-transformation comparison. "In trying to erase a trace, you often imprint a larger one." Armed with this understanding, we framed *TraceTarnish*'s operations and outputs around these five isolated features, using them to conceptualize and implement enhancements that further strengthen the attack.

Keywords: Adversarial Stylometry · Privacy · Unicode Steganography with Zero-Width Characters · Feature Selection Using Information Gain

1   Introduction

> When   I   leave,
> burn the spread of this bed,
> that I touched. Burn the chair in
> the living room, in your wall incinerator.
> Wipe down the furniture with alcohol, wipe
> the door-knobs. Burn the throwrug in the parlour.
> Turn the air-conditioning on full in all the rooms
> and spray with moth-spray if you have it. Then,
> turn on your lawn sprinklers as high as they'll
> go and hose off the sidewalks. *With any*
> *luck at all, we can kill the trail...*
>
> ―――――――――――――――――
>
> *Fahrenheit 451*
> *Ray Bradbury*

This study represents the natural progression of our previous work (*Dilworth* [6]), extending the research and providing empirical evidence to support our hypothesis: that a combination of (A) machine translation, (B) paraphrasing, and (C) imperceptible steganographic payloads–identified as suitable adversarial stylometric techniques–will weaken or eliminate a composer's trace. By "trace," we refer to a writer's stylistic tendencies, including grammatical cues, lexical structures, spelling quirks, phrasing patterns, and the frequency of these features, among others.

To this end, we will source either a corpus containing only the work of a single author or a corpus composed of many individually authored texts. The corpus will be conversational[1] to better reflect the use case introduced in our previous study: a privacy-minded user engaging with a messaging platform– whether a major social media service or a niche forum–who has taken reasonable precautions to anonymize their identity. Even with minimal risk of identification, posting a textual message introduces another attack vector. If the user frequently misspells a particular word (as an example), that habit can leave a trace capable of de-anonymizing them.

We therefore propose adding adversarial stylometry to the user's OPSEC[2] toolkit. *TraceTarnish*, the script developed in our prior study, serves as the actionable avatar of adversarial stylometry: it preserves message meaning, strips away the user's stylistic kernel, and injects poison to thwart text-based identification methods. See (**Figure 1**), which recapitulates the workflow of our attack, *TraceTarnish*.

―――――――――

[1] We intend to evaluate various authors and writing styles at differing levels of academic rigor, so while some of the data may be "conversational," we will survey a variety of samples to ensure a comprehensive evaluation.

[2] Operations security (OPSEC), according to the National Institute of Standards and Technology (NIST), is a "systematic and proven process intended to deny... potential adversaries information about capabilities and intentions."

In this work, we will evaluate how effectively *TraceTarnish* achieves its dual goals of preserving privacy and enabling anonymous interactions online. Specifically, we will compare our approach against existing stylometric methods, identify any weaknesses in the script, and propose and implement improvements if needed.
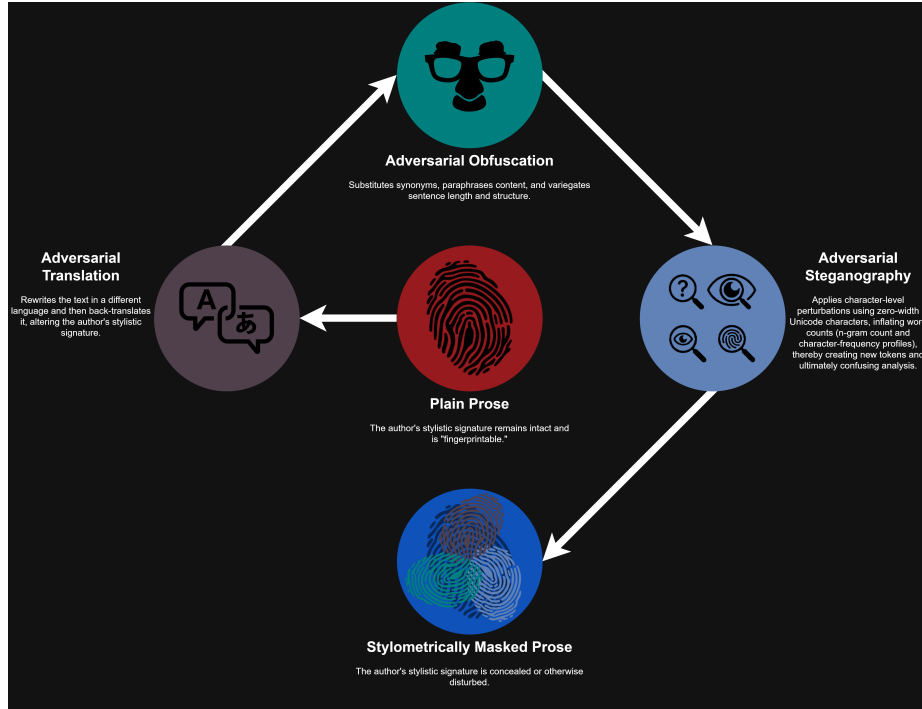

Fig. 1: An operational overview of *TraceTarnish*, wherein the attack passes a text-only message through a process that (1) round-trip translates it using machine translation, (2) obfuscates the text by paraphrasing, and (3) embeds noise via steganography.

2   Background

In this section, we indicate existing literature that mentions both steganography and stylometry in the same breath. Notice our use of the operant phrase "in the same breath;" our scoring of the literature yielded unimpressive results. Nevertheless, the dearth of related publications reinforces the novelty of our approach, as querying the keywords "steganography" and "stylometry" returns only three articles[3]. The articles, however, are not equivalent in content. To return

---

[3] This could very well be the result of our negligence when performing the search, but that is neither here nor there.

to our specific choice of words, two of the three papers are more scholarly with little technical consideration. Note that "scholarly" indicates a paper whose focus is less on experimentation and more on theory and case studies, whereas a "technical" paper, while it certainly discusses underlying theory, emphasizes empirical considerations and the interplay between the aspects under investigation. In this way, we will outline the relevant components of the scholarly publications and provide an overview of the technical publication. This literature review will synthesize the findings of the examined works, culminating in a definition of our specific approach to adversarial stylometry–"injection"–which parallels our presentation of tampering variants as found in Section 7 of our previous work [6].

To ensure that no stone is left unturned, we first take the time to define our use of steganography and stylometry as they relate to our study.

2.1   What is Steganography?

Network censors can block anonymity services by looking for protocol-specific strings in packet payloads, not merely the IP header. When a firewall examines the data carried by packets, it can spot distinctive patterns that betray the use of certain anonymity networks. For instance, early Tor releases embedded the literal word "Tor" in the TLS certificates exchanged during the handshake, making the traffic trivially recognizable. A censor could then disrupt those connections by sending TCP RST packets. To counter this, contemporary designers aim to give anonymity-network protocols steganographic qualities–making them hard to fingerprint while still functional. This conceal-in-plain-sight strategy, which prevents a passive observer from even noticing the hidden communication, inspired the creation of Tor's "bridge" relays: volunteers in uncensored regions run special nodes that clients can reach without exposing the usual Tor signatures. In essence, steganography leverages the excess capacity of a carrier medium to embed hidden data without raising suspicion (*Edman et al.* [8] & *Anderson et al.* [3]).

2.2   What is Stylometry?

Stylometry is a linguistic, side-channel technique for attributing a piece of writing to its author. Unlike classic forensic methods such as DNA analysis or handwriting examination, stylometry examines patterns in the text itself–word choice, syntax, punctuation, and other stylistic markers–that tend to remain stable across an author's body of work. By modeling these consistent characteristics, analysts can classify unknown documents and infer the most likely writer, supporting authorship attribution tasks (*Edman et al.* [8] & *Anderson et al.* [3]).

2.3   Proposing "Injection" as the Fourth Canonical Adversarial Stylometric Strategy Alongside Translation, Imitation, and Obfuscation

Adversarial machine-learning principles, or at least those tangential to stylometry, involve injecting invisible characters, control characters, homoglyphs, and

pseudo-homoglyphs–distinct characters that resemble typical glyphs and are otherwise referred to as "imperceptible perturbations." Our focus is on this collection of invisible characters (valid characters that, by design, do not render a visible glyph) and homoglyphs (unique characters that render the same or similar glyphs) to perturb models. A common example of homoglyphs is the similarity between the English and Cyrillic alphabets. These characters are invisible to human users, but the underlying bytes used to encode them can drastically change the output of the systems that process them (*Boucher et al.* [5]).

The goal is to produce incorrect outputs when valid characters are injected; when encoding characters that are unrecognizable by the system, the aim is to propagate the unknown word from the initial input to the final output. This attack vector targets the rampant adoption of natural-language-processing (NLP) systems and engineers' reluctance–or reduced concern–for properly sanitizing user input. The motivation is that other overt attempts at adversarial tampering can manifest during further processing. For instance, while injecting spelling mistakes could feasibly lead to authorial misclassification, paraphrasing the text that contains those errors could modify its meaning in a detectable way. The key ingredient of this class of attack is that it is largely system-agnostic and does not produce visual artifacts (*Boucher et al.* [5]).

We now provide a precise delineation of our use of "Injection."

Injection Injection exploits invisible characters, control characters, homoglyphs, and pseudo-homoglyphs to perturb stylometric models without altering the visible text. By inserting characters that are either non-rendering (zero-width spaces, zero-width joiners, etc.) or that render as visually identical glyphs from different code points (e.g., Latin "a" vs. Cyrillic "a"), the attacker creates "imperceptible perturbations." These perturbations are invisible to human readers but change the underlying byte sequence, causing downstream NLP pipelines to mis-parse, split, or mis-tokenize words.

2.4   Implementing "Injection" in the *TraceTarnish* Attack Script

In materializing "injection" in our attack script *TraceTarnish*, we prioritize zero-width Unicode characters for steganographic encoding, as rarer characters used in homoglyphic substitution attacks are more likely to trigger the rendering of special characters reserved for representing valid Unicode encodings that lack a corresponding glyph. Our attack currently embeds invisible characters within words, causing the corrupted words to be parsed as multiple shorter words, interfering with standard processing, poisoning, degrading, and altering a stylometric system, all without significantly modifying the semantic meaning of the text.

3   Methodology

With this study, we will compile the works of a particular author–or of multiple authors–ensuring that each work is verifiably authored exclusively by its claimed

creator–a challenge given the widespread use of AI writing assistants and editors. We will populate the full-length texts into a database with two labels. The first label will indicate the text's state–either "anonymized" or "non-anonymized." The second label will contain the text itself, with the raw text receiving the "non-anonymized" label. We will then use our script, *TraceTarnish*, to adversarially modify the raw text to obscure authorship; this altered text will receive the "anonymized" label. The result will be a dataset of an author's solely authored works alongside their adversarially modified counterparts.

This dataset will be fed into a *StyloMetrix* (*Okulska et al.* [2, 12]) pipeline, where we generate vectors of stylistic features. At its core, the tool engineers features–i.e., *Stylometrix* vectors–from which statistical analysis can be performed by explainable classification models, enabling data mining of corpora and further linguistic investigation. The project's GitHub page[4] exhaustively enumerates the tool's extensive suite of metrics, ranging from grammatical forms to lexical attributes. This "explainability," afforded by the team's emphasis on interpretability, lets users see exactly which *Stylometrix* vectors drive each classification–and because the model's reasoning is (relatively) transparent, no hidden data profiling or opaque black-boxes are required, which aligns well with the preferences of privacy-conscious users.

We will use these vectors as inputs to a Random Forest classifier to assess how well the model can identify the correct label[5]. If the anonymized text can be easily identified, we must reconsider our authorship-obfuscation strategy. Conversely, if the attack shows success[6], we will examine the stylistic features uncovered by *StyloMetrix* using Information Gain, which aligns with our choice of a Random Forest algorithm (which employs multiple decision trees). Given the simplicity of our dataset, quantifying feature relevance with Information Gain is appropriate. We will then isolate and analyze the most informative features that discriminate between anonymized and non-anonymized text produced by *TraceTarnish*.

4   Dataset

As for what data we will be sleuthing, we sourced an existing dataset from Kaggle [1] containing Reddit comments. To better facilitate our analysis, we reduced the size of the original dataset to 100 entries. This was primarily due to the time our script needed to fully compute. By far, the largest performance bottleneck of the script was the round-trip translation; all other components were

---

[4] https://github.com/ZILiAT-NASK/StyloMetrix/blob/main/resources/metrics_list_en.md

[5] Henceforth, we will forgo any further elaboration on the Random Forest model fitting, as the classifier correctly identified and labeled the anonymized and non-anonymized samples 100% of the time. As such, including a confusion matrix and reporting the classifier's metrics (Recall, Precision, and F_1 Score) would be needlessly otiose.

[6] Irrespective of the outcome, exploring the Information Gain results will still be worthwhile.

relatively efficient. As a consequnce, we preprocessed the dataset to prepare it for subsequent processing by our *TraceTarnish* script.

One step of preprocessing the dataset involved reformatting the data in a way that could be read by *TraceTarnish*. This final product was then passed to *TraceTarnish* to produce anonymized text, where our use of "anonymized" alludes to text that has been stripped of the author's stylistic trace and embedded with noise. See (**Figure 2**) for a preview of our dataset with omissions[7]. See (**Figure 3**) for a quick appraisal of our data's *Stylometrix* vectors.

Fig. 2: Our dataset containing the inputs fed to and the outputs retrieved from *TraceTarnish*. Rows assigned a "0" indicate raw Reddit comments; rows assigned a "1" represent Reddit comments that have been anonymized via *TraceTarnish*. To eliminate the influence and subsequent variability of input text length, we ensured that the material passed to *TraceTarnish* was of a consistent and uniform length.

---

[7] As a disclaimer: The viewpoints, opinions, and statements expressed by the Reddit users (otherwise known as "Redditors")–as exposed in the scrutinized dataset–are theirs and theirs alone, and are not indicative of the authors or their affiliated organizations; similarly, the thoughts portrayed therein do not necessarily reflect those of the authors and their affiliated organizations. The content generated by the "Redditors" is the sole responsibility of the "Redditors", and the accuracy and completeness of their content is not endorsed or guaranteed. We, the authors, have no affiliation with the company or the brand; our mentioning the product/service should not be construed as advertisement or promotion. We, the authors, are not liable for any content that may be deemed offensive, inappropriate, or inaccurate. Our study required realistic user discourse, which was easily fostered on the platform; our use of that data was solely for academic purposes. It was intended to illustrate concepts, demonstrate examples, and enable the scientific process. As an aside, we hope the irony of pre-emptively framing what might be deemed as "free expression" is not lost on the reader, for nothing is safe from the relentless pursuit of the *"Mechanical Hound"*–be it artistic pursuits or other equally important expressive modalities.

```
                                          text  POS_VERB  POS_NOUN  POS_ADJ   POS_ADV   POS_DET  POS_INTJ  ...        OM  RCI  DMC   OR       QAS   PA   PR
0    I wouldn't say so... You'll still be surrounde...  0.382353  0.235294  0.088235  0.058824  0.000000       0.0  ...  0.000000  0.0  0.0  0.0  0.029412  0.0  0.0
1    I wouldn't say you're looking ...  0.125000  0.125000  0.000000  0.000000  0.000000       0.0  ...  0.041667  0.0  0.0  0.0  0.000000  0.0  0.0
2    It's a common hobby - they'll take their bikes...  0.125000  0.225000  0.050000  0.025000  0.100000       0.0  ...  0.000000  0.0  0.0  0.0  0.100000  0.0  0.0
3    Motorcycles can be brought t...  0.074074  0.185185  0.000000  0.000000  0.037037       0.0  ...  0.000000  0.0  0.0  0.0  0.037037  0.0  0.0
4    being able to press the lock button on your ke...  0.261905  0.142857  0.023810  0.095238  0.071429       0.0  ...  0.000000  0.0  0.0  0.0  0.047619  0.0  0.0
..                                         ...       ...       ...       ...       ...       ...       ...  ...       ...  ...  ...  ...       ...  ...  ...
195  The common state is like t...  0.052632  0.052632  0.052632  0.052632  0.000000       0.0  ...  0.000000  0.0  0.0  0.0  0.000000  0.0  0.0
196  He couldn't be leading the church astray, lead...  0.344828  0.241379  0.000000  0.172414  0.068966       0.0  ...  0.068966  0.0  0.0  0.0  0.068966  0.0  0.0
197  The church did n...  0.000000  0.000000  0.000000  0.000000  0.000000       0.0  ...  0.000000  0.0  0.0  0.0  0.000000  0.0  0.0
198  Having access to health care, being able to vo...  0.371429  0.228571  0.057143  0.028571  0.085714       0.0  ...  0.000000  0.0  0.0  0.0  0.057143  0.0  0.0
199  You have access to the s...  0.161290  0.161290  0.000000  0.000000  0.064516       0.0  ...  0.000000  0.0  0.0  0.0  0.032258  0.0  0.0

[200 rows x 197 columns]
```

Fig. 3: The *Stylometrix* vectors produced from our *TraceTarnish* data.

5   Information Gain Results

In this section, we present our Information Gain findings, which we calculated using the *StyloMetrix* vectors and the labels of our *TraceTarnish* data. To better facilitate the forthcoming exposition, we first define the strategy and restate its purpose within the context of this study.

5.1   Decision-Tree Fitting Guided by the Information-Gain (Entropy) Criteria

Feature selection is a critical step in a machine-learning workflow; it involves choosing the most relevant and informative features (or variables) from a dataset to build a predictive model.

Information Gain helps identify the most informative features by measuring how much they reduce uncertainty (or entropy). Entropy quantifies a dataset's impurity or disorder. Information Gain evaluates prediction-relevant features by measuring the reduction in entropy after each split of a decision tree. It computes the difference between the entropy before and after a split, where a higher value indicates a better split. A high Information Gain means the split is useful for making decisions.

The features exhumed by this process encapsulate *TraceTarnish*'s stylistic signature–a signature that can be used to detect its usage, which is unideal given the objective of becoming less identifiable.

5.2   High-Impact Stylometric Indicators in *TraceTarnish* Outputs

Switching from the general to the specific, we now report our Information-Gain yields. As a rule of thumb, a value that is significantly larger than the average gain for the dataset–often above 0.5–is considered a high Information-Gain reading. In practice, splits with Information-Gain readings > 0.5 are typically prioritized, while those < 0.1 are often discarded as uninformative.

This will be the guiding heuristic that determines which features are suitable for further examination; applying it leaves us with 5 of the original 196 features.

The remainder of this section will be dedicated to defining the isolated features, providing additional context for the features and how they may be calculated, and hypothesizing why the *TraceTarnish* data would yield those results.

See (**Table 1**) for the most relevant stylometric features and the information gained from them. See (**Table 2**) for a more granular look at the most pertinent stylometric features and their unprocessed calculated values. See (**Figure 4**) for a radar-chart assemblage that captures the value trajectory across anonymized and non-anonymized text pairs for the audited features. See (**Figure 5**), which shows a dendrogram of the Burrow's Delta matrix calculated for the selected data points (or, more specifically, the text associated with each data point).

StyloMetrix Category	*StyloMetrix* Code	*StyloMetrix* Name	Information-Gain Reading
Lexical	L_FUNC_T	Function words types	0.5332
Lexical	L_CONT_A	Content words	0.5421
Lexical	L_FUNC_A	Function words	0.5624
Lexical	L_CONT_T	Content words types	0.5987
Statistics	ST_TYPE_TOKEN_RATIO_LEMMAS	Type-token ratio for words lemmas	**0.6748**

Table 1: The most informative *StyloMetrix* features extracted from our *TraceTarnish* data; highlights the top-ranking linguistic metrics that provide the greatest discriminative power for *TraceTarnish* analysis.

Text_ID	L_CONT_A	L_FUNC_A	L_CONT_T	L_FUNC_T	ST_TYPE_TOKEN_RATIO_LEMMAS
I-NANON	0.5000 →	0.6765 →	0.5000 →	0.5588 →	0.9412 →
I-ANON	0.1250 ↓	0.2917 ↓	0.1250 ↓	0.2917 ↓	0.4167 ↓
II-NANON	0.4250 →	0.6000 →	0.4000 →	0.4250 →	0.7750 →
II-ANON	0.2593 ↓	0.1852 ↓	0.2593 ↓	0.1852 ↓	0.4444 ↓
III-NANON	0.3611 →	0.6667 →	0.3611 →	0.4722 →	0.7500 →
III-ANON	0.1000 ↓	0.0000 ↓	0.1000 ↓	0.0000 ↓	0.1000 ↓
IV-NANON	0.4412 →	0.6471 →	0.3824 →	0.5000 →	0.7941 →
IV-ANON	0.0526 ↓	0.1579 ↓	0.0526 ↓	0.1579 ↓	0.2105 ↓
V-NANON	0.3590 →	0.6410 →	0.3077 →	0.5128 →	0.7692 →
V-ANON	0.0625 ↓	0.0625 ↓	0.0625 ↓	0.0625 ↓	0.1250 ↓

Table 2: A sampling of the most insightful *StyloMetrix* features and their raw values demonstrates how the transformation from non-anonymized to anonymized text results in a steady depreciation of the included features, explaining why they remain highly informative. For clarification: "ANON" = Anonymized; "NANON" = Non-anonymized. Instances of paired source and anonymized texts are indicated using Roman numerals. Regarding the rows of data selected, we simply collated the first five data points.
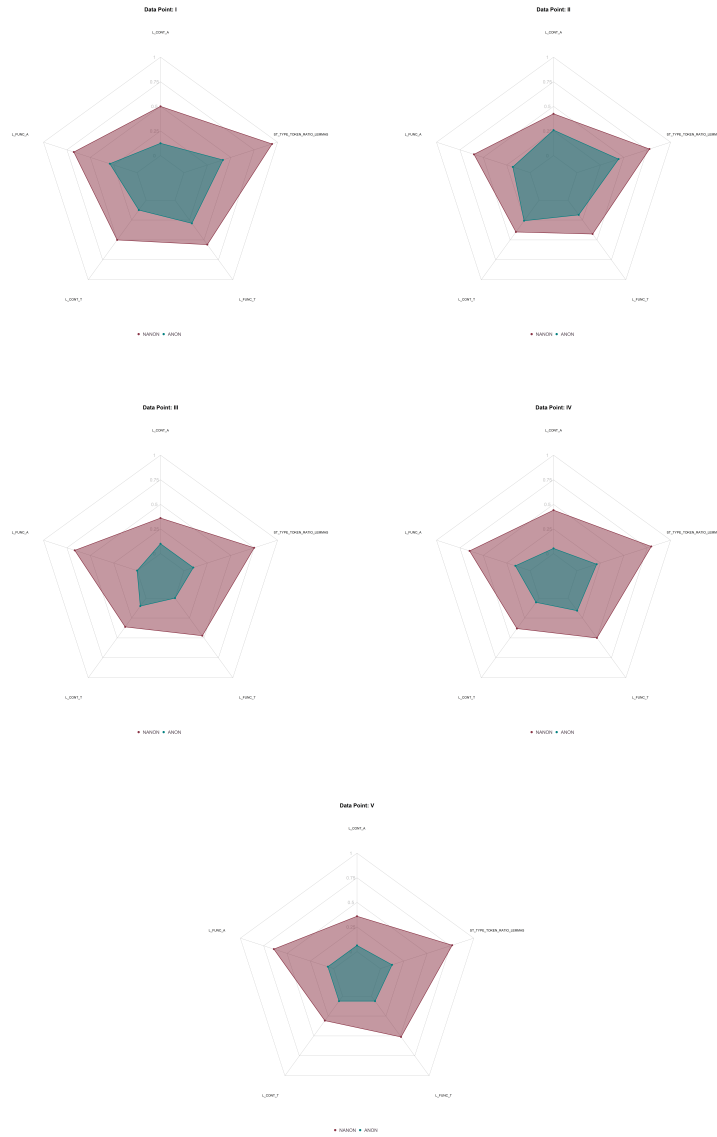
Fig. 4: A collection of radar charts that visually represent the contents of (**Table 2**), illustrating how the isolated *StyloMetrix* feature scores plummet when anonymizing text.
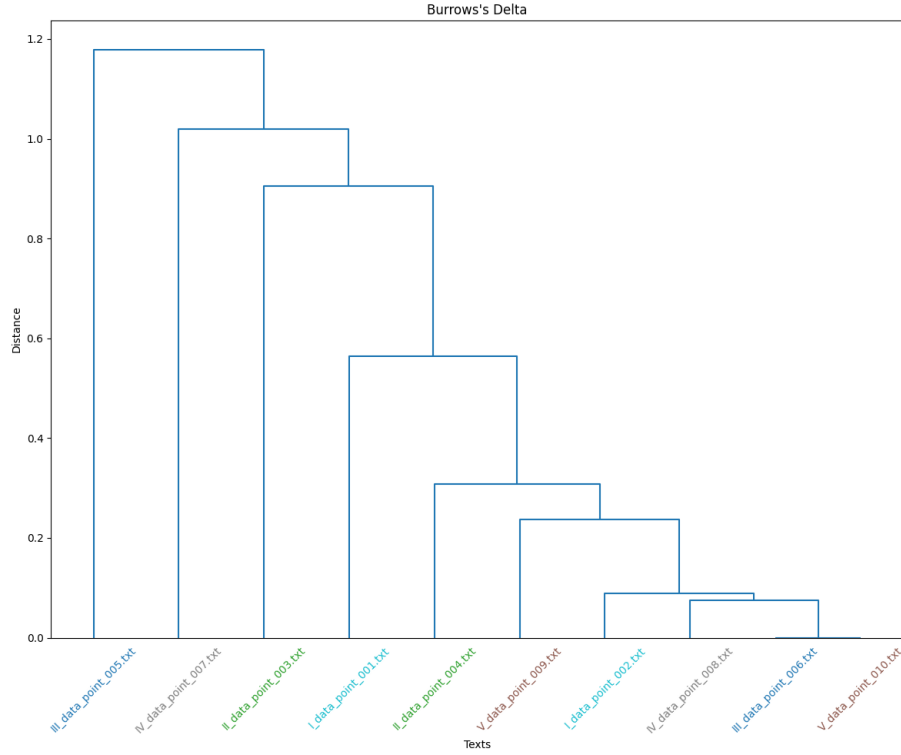
Fig. 5: Visualizes the computed Burrows's Delta values for the first five pairs of anonymized and non-anonymized text samples using a dendrogram. The dendrogram labels are color-coded by groups–groups derived from our established naming convention. Here, a group of paired texts is denoted by their shared Roman-numeral prefix in their filenames. All odd filename suffix values, e.g., "001" in "I_data_point_001.txt," correspond to the "NANON" (non-anonymized) label; the even suffixes correspond with the "ANON" (anonymized) label. An observation that can be made is that, in general–excluding "V_data_point_009.txt"– there is an apparent divide between the "NANON" and "ANON" samples. The code used to generate the graph is courtesy of James O'Sullivan [13].

5.3   Feature Definitions

Here, we review the features identified because of their elevated Information-Gain scores.

Function Words Function words are a set of grammatically required items that hold sentences together. This group comprises pronouns, conjunctions, prepositions, determiners, auxiliary verbs, qualifiers, and interrogatives. Unlike content words, which carry substantive meaning, function words act as the "backbone"

of a text. Because they are largely independent of topic and genre, they are especially useful for describing an author's stylistic fingerprint. Typical examples include: the, and, a, of, to, in, that, with, but, it, among others [15].

Function Word Types Function words can be divided into several types:

- ❖ Pronouns replace nouns (e.g., he, she, they).
- ❖ Prepositions indicate relationships (e.g., in, on, at).
- ❖ Conjunctions connect words or clauses (e.g., and, but, although).
- ❖ Articles define nouns as specific or nonspecific (e.g., the, a, an).
- ❖ Auxiliary verbs assist main verbs (e.g., am, is, can).
- ❖ Interjections provide emotional context (e.g., oh, wow).

These words are crucial for constructing coherent sentences and for understanding the relationships between ideas, thereby supporting the structural framework of language that surrounds more meaningful content words.

Content Words Content words, by contrast, express the substantive information of a passage and consist of nouns, main verbs, adjectives, and adverbs. For a stylometric system to be robust in practical settings, its features must remain stable despite topical shifts. Consequently, stylistic analyses generally treat content words as non-diagnostic, except in occasional special cases [15].

Content Word Types Content word types refer to the categories of content words. While the specific types generally align with parts of speech, the term "types" emphasizes the distinct roles these words play in language. The main types are:

- ❖ Nouns, which identify people, places, things, or ideas.
- ❖ Verbs, which represent actions, states, or occurrences.
- ❖ Adjectives, which describe or modify nouns.
- ❖ Adverbs, which modify verbs, adjectives, or other adverbs.

Type-Token Ratio for Word Lemmas The Type-Token Ratio (TTR) measures the proportion of distinct word types (vocabulary size) to the total number of tokens (text length). A high TTR suggests a rich and varied lexicon, which may reflect coverage of many topics or an author's tendency to approach a limited set of themes from multiple angles with diverse phrasing; a low TTR indicates a more restricted vocabulary or frequent repetition of ideas using similar wording [14].

5.4   The Impact of *TraceTarnish* on an Author's Stylistic Signature

Now, we hypothesize why the *TraceTarnish* data yielded the observed results.

On average, the anonymized *TraceTarnish* outputs were noticeably shorter than their non-anonymized input counterparts, primarily because of the adversarial obfuscation step. This step replaces key nouns, verbs, and modifiers with synonyms or alternative constructions, making it harder for keyword-based detectors to match the original phrasing while preserving the overall meaning.

The adversarial translation step likely removes articles and replaces proper nouns and similar elements with functional words, reducing overall lexeme counts. The round-trip translation typically introduces lexical drift, grammatical shifts, or loss of nuance.

Finally, the adversarial steganography step, presumably, affects the Type-Token Ratio (TTR) metric, as this aspect of the attack can virtually double the number of tokens registered for a given text. The zero-width Unicode characters are likely to be (A) interpreted as legitimate whitespace instances or (B) valid characters deserving of analysis, causing a word such as "privacy" to be parsed as (A) "pri" and "vacy" or (B) a string that visually resembles "privacy" but would not be detected as such[8].

These alterations appear to disproportionately affect function words and content words alike. By substituting or removing articles, prepositions, and conjunctions, the adversarial steps reduce the frequency of common function-word types, which in turn lowers their contribution to the overall token count. Simultaneously, synonym replacements and lexical drift diversify the pool of content words, increasing the number of distinct content-word types while often shortening or fragmenting individual instances. The combined effect is a marked rise in the Type-Token Ratio: fewer repeated function words and a broader, more varied set of content words inflate the ratio, even though the total token count may decline. Consequently, the text becomes statistically more "lexically rich" but less faithful to the original distribution of function and content word categories.

To illustrate our point, consider the following sentences. Original sentence: "If you are distressed by anything external, the pain is not due to the thing itself, but to your estimate of it; and this you have the power to revoke at any moment" (*Marcus Aurelius*). Adversarial version (illustrating the concept): "If you suffer for something external, the pain is not from him, but from his preciousness, and he has the power to withdraw at any time."

The text-analysis results for the original sentence are: TTR 0.848; unique words 28; total words 33. The results for the adversarial version *without* steganographic tampering are: TTR 0.923; unique words 24; total words 26. From this,

---

[8] If robust sanitation protocols were applied to all input–whether LLM prompts or generic text fields–Unicode smuggling attacks that hide zero-width Unicode characters (similar to ASCII-smuggling attacks) would be largely neutralized. Proper filtering would detect and strip invisible code points before the text reaches downstream processors, preventing the hidden payload from being interpreted as executable or meaningful data.

we can conclude that the edits increase the TTR, indicating a text that is statistically more lexically rich while using fewer overall tokens. The results for the adversarial version *with* steganographic tampering are: TTR 0.660; unique words 33; total words 50. For clarification, the two adversarial versions look identical, at least visually, but the former aligns with our explanation, while the latter conforms to our experimental results. In the face of steganographic tampering, the TTR decreases and the word counts balloon; when all other adversarial steps are performed without steganography, the TTR increases and the word counts shrink. See (**Figure 6**) for a visual aid of our TTR working example.

Overall, *Trace Tarnish* simplifies its textual input–producing a more simplistic rendition of the input text–while the text's semantic meaning is effaced through the incorporation of words that are valid in one language but invalid in another, and the number of tokens in the paraphrased text increases, roughly doubling the original count.

```
original_version.csv:

    "If you are distressed by anything external,
    the pain is not due to the thing itself, but
    to your estimate of it; and this you have
    the power to revoke at any moment."


adversarial_version_without_steganography.csv:

    "If you suffer for something external,
    the pain is not from him, but from his
    preciousness, and he has the power to
    withdraw at any time."


adversarial_version_with_steganography.csv:

    If"[U+200F]y[U+200F]ou s[U+200F] uffer
    f[U+200F]or s[U+200F]omething e[U+200F] ,xternal
    t[U+200F]he p[U+200F]ain i[U+200F]s n[U+200F] ot
    f[U+200F]rom h[U+200F]im, b[U+200F]ut f[U+200F] rom
    h[U+200F]is p[U+200F]reciousness, a[U+200F] nd
    h[U+200F]e has t[U+200F]he p[U+200F]ower t[U+200F] o
    w[U+200F]ithdraw a[U+200F]t a[U+200F]ny t[U+200F]".ime
```

```
                                                    python type_token_ratio.py original_version.csv
output saved to:
original_version_out.txt

Type-Token Ratio (U/T):        0.8485
Number of Utterances:          1
Total Number of Words (T):     33
Total Number of Unique Words (U): 28
```

```
                                            python type_token_ratio.py adversarial_version_without_steganograp
hy.csv
output saved to:
adversarial_version_without_steganography_out.txt

Type-Token Ratio (U/T):        0.9231
Number of Utterances:          1
Total Number of Words (T):     26
Total Number of Unique Words (U): 24
```

```
                                            python type_token_ratio.py adversarial_version_with_steganography.
csv
Traceback (most recent call last):
  File                                      \type_token_ratio.py", line 111, in <module>
    output = main(sys.argv[1])
  File                                      \type_token_ratio.py", line 34, in main
    flines = f.readlines()
  File                                      \lib\encodings\cp1252.py", line 23, in decode
    return codecs.charmap_decode(input,self.errors,decoding_table)[0]
UnicodeDecodeError: 'charmap' codec can't decode byte 0x8f in position 9: character maps to <undefined>
```

```
Processing <adversarial_version_with_steganography.csv> using the online Type-Token
Ratio Calculator <https://calculator.academy/type-toke-ratio-ttr-calculator/>

output saved to:
N/A

Type-Token Ratio (U/T):             0.6600
Number of Utterances:               1
Total Number of Words (T):          50
Total Number of Unique Words (U):   33
```

Fig. 6: The outputs of Steven C. Howell's "type_token_ratio.py" program [10] using a processed and an unprocessed version of a Marcus Aurelius quote.

6   Proposed Enhancements and Next Steps

With everything we have learned from our experimentation, it has become increasingly apparent that our attack pipeline would benefit from an "adversarial

imitation" step, but one unlike our previously proposed implementation. Initially, we sequenced the components as Translation → Imitation → Obfuscation → Steganography; the current iteration runs Translation → Obfuscation → Steganography. While this satisfies our ends, the quality of the text produced could always be improved. Much like Hume's guillotine (the is/ought problem)[9], what *ought* to be is a distant horizon, while what *is*, is close and in immediate view. Our attack's outputs *ought* to be sensible and comprehensible. Our attack *is* effective in that it verifiably deteriorates an author's stylistic signature, with the trade-off being coherence. Although *TraceTarnish*'s outputs (more or less) embody the themes of their inputs, the state of the inputs after processing can have immense repercussions.

A well-formed statement has a far better chance of producing a passable output, whereas colloquialisms, slang, and other aspects typical of conversational text can produce suboptimal results. This is expected; machines are deterministic, so analytically-driven processes such as machine translation and paraphrasing are also subject to that determinism. After all, the composition of genuinely authored text can be influenced by manifold influences and sundry stimuli; it is lithe, effervescent, and mercurial. This too appears in certain outputs, as non-conventionally structured text strewn with emoticons leads to unremarkable results.

With all this in mind, we have concluded that some form of quality control is necessary, perhaps in the form of a large language model (LLM) [16]. Given our particular ask, a lightweight LLM seems fitting. The idea would be to prompt an *offline*, *self-hosted* LLM to assume the role of Grammarly.

The thought process behind our design is as follows. Our data should never leave our device, hence the *self-hosting*. Our data, whatever it may be, should *not* be used to train future models without our express consent, which is why the LLM will be *offline*. If it were online, and we were to disclose our sensitive information in some chat, a log of that chat would be produced. That log could then be read and accessed by a third party; even if we deleted the original conversation, the "deleted" message may still be retained and probed for a variety of reasons [11].

Our agent–or LLM instance with a clear objective–will also be tasked with retaining the steganographic payloads of the original text while proofreading and revising the input it receives.

While performing its grammatical revisions, the LLM will be asked to randomly assume the identity (or profile) of a random person, randomizing gender, age, education level, and nationality. This subtask will characterize the code component as a whole, earning the "adversarial imitation" label, if only in spirit.

Our attack sequence, with the above modifications, will become Translation → Obfuscation → Steganography → Imitation. As before, we will refactor the

---

[9] Hume's guillotine, also called the is-ought problem, is the claim that one cannot logically derive prescriptive ("what ought be" or "how things should be") statements solely from descriptive ("what is" or "observation-based") premises. The gap between facts and values must be bridged by an additional normative premise.

code, add to it, and evaluate the changes iteratively. See (**Figures 7; 8**) for our code's current implementation.

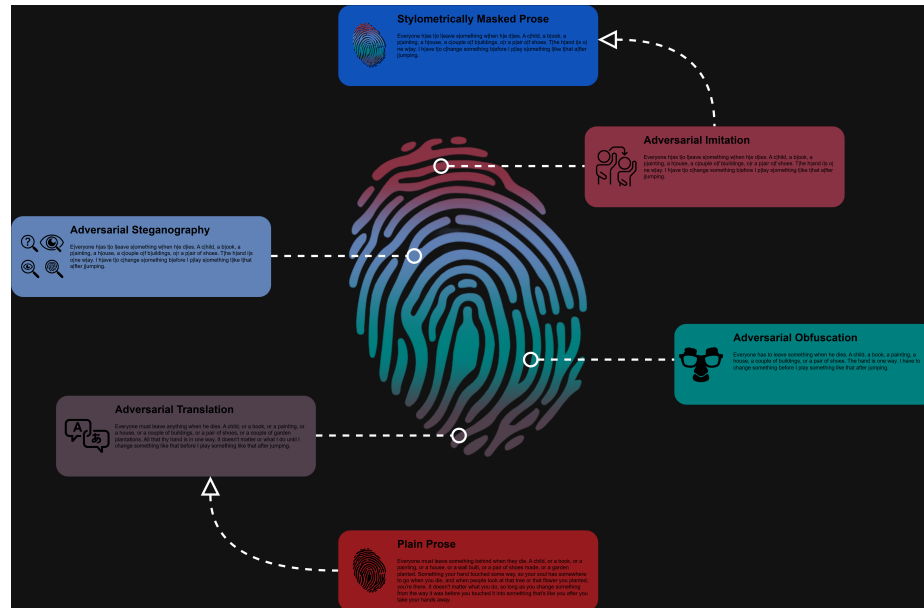Fig. 7: The terminal output for our updated *TraceTarnish* script.

Fig. 8: To bring something into existence–the very act of creation–imbues, imparts, and etches a fragment of your soul onto whatever you produce. This corporality–making something from nothing–is suffused with an undeniable essence–an essence that is uniquely yours. While sacred, that essence also erodes privacy in the most profound way: what is unique is detectable; what is detectable is leveragable. Our aim with this attack is to *"burn [this essence] to ashes, [and] then burn the ashes"* (*Ray Bradbury*). Pictured above is a sample workflow visualization of our updated *TraceTarnish* script, in which we process another of Bradbury's quotes.

7   Conclusion

> We need to
> be really bothered
> once in a while. How long
> is it since you were really
> bothered? About something
> important, about something
> real?

Fahrenheit 451
Ray Bradbury

We would like to bring our current study to a close by asking two open-ended questions. First, what makes you, *you*? This, naturally, invites a deeper, more

disconcerting inquiry. How could the answer to the first question be used against *you*?

From the perspective of computational stylistics (stylometry), a robust profile can be synthesized using the counts of function words, tallies of function word types, and the richness of the author's vocabulary (or the proportion of unique words by the length of their write-up).

Like it or not, *you* can be reduced to a number. *You* can be identified, tracked, and censored based on a numerical score or array of scores. This is *you*, and what makes you, *you* is exploitable. Does it have to be this way? Or is there a better way to be?

While not perfect–and certainly not complete–*TraceTarnish* endeavors to eviscerate the *you* that can be taken advantage of. But it is not without its shortcomings. As we've uncovered, calculating and exploring the correct subset of stylometric features may signal the utilization of *TraceTarnish*.

Be that as it may, there is still work to be done, for the erosion of privacy is *nigh* [7], surveillance is *imminent* [4], and censorship *looms around the corner* [9].

Now, we leave you with a final question: what are *you* going to do about it[10]?

References

1. datasets for natural language processing, `https://www.kaggle.com/datasets/to ygarr/datasets-for-natural-language-processing`
2. Stylometrix, `https://github.com/ZILiAT-NASK/StyloMetrix/blob/main/READ ME.md`
3. Anderson, A., Collie, B., Vel, O.D., McKemmish, R., Mohay, G.: Computer and Intrusion Forensics. Artech (2003), `https://ieeexplore.ieee.org/document/9 100004`
4. Belanger, A.: Flock haters cross political divides to remove error-prone cameras (11 2025), `https://arstechnica.com/tech-policy/2025/11/flock-haters-cro ss-political-divides-to-remove-error-prone-cameras/`
5. Boucher, N., Shumailov, I., Anderson, R., Papernot, N.: Bad characters: Imperceptible nlp attacks. In: 2022 IEEE Symposium on Security and Privacy (SP). pp. 1987–2004 (2022). `https://doi.org/10.1109/SP46214.2022.9833641`, `https://ieeexplore.ieee.org/document/9833641`
6. Dilworth, R.: Unveiling unicode's unseen underpinnings in undermining authorship attribution (8 2025), `https://arxiv.org/abs/2508.15840`
7. Discord: Update on a security incident involving third-party customer service (10 2025), `https://discord.com/press-releases/update-on-security-inciden t-involving-third-party-customer-service`
8. Edman, M., Yener, B.: On anonymity in an electronic society: A survey of anonymous communication systems. ACM Comput. Surv. **42** (12 2009). `https://doi. org/10.1145/1592451.1592456`, `https://doi.org/10.1145/1592451.1592456`

---

[10] For you see, "... [we] are looking at... [those who are craven]. [We] saw the way things were going, a long time back. [We] said nothing. [We're] one of the [apathetic] who could have spoken up and out when no one would listen to the '[culpable],' but [we] did not speak and thus became [culpable ourselves]." (*Ray Bradbury*)

9. Goldman, E.: The "segregate-and-suppress" approach to regulating child safety online. Stanford Technology Law Review **28** (4 2025). `https://doi.org/10.2139/ssrn.5208739`

10. Howell, S.: Type-token ratio (2016), `https://github.com/StevenCHowell/type_token_ratio`

11. Lightcap, B.: How we're responding to the new york times' data demands in order to protect user privacy (2025), `https://openai.com/index/response-to-nyt-data-demands/`

12. Okulska, I., Stetsenko, D., Kołos, A., Karlińska, A., Głąbińska, K., Nowakowski, A.: Stylometrix: An open-source multilingual tool for representing stylometric vectors (9 2023), `https://arxiv.org/abs/2309.12810`

13. O'Sullivan, J.: Stylometry (2024), `https://github.com/jamesosullivan/stylometry`

14. Savoy, J.: Machine Learning Methods for Stylometry: Authorship Attribution and Author Profiling. Springer Cham (9 2020). `https://doi.org/10.1007/978-3-030-53360-1`

15. Wang, H.: Enhancing representation generalization in authorship identification (9 2023), `https://arxiv.org/abs/2310.00436`

16. Wang, H.: Defending Against Authorship Attribution Attacks with Large Language Models. Ph.D. thesis, Indiana University (6 2025), `https://hdl.handle.net/2022/33626`