TinyAttack: Exploring Stylistic Vulnerabilities in Large Language Models

Mamta*, **Bogdan Grecu***, **Oana Cocarascu**
King's College London
{mamta.name, bogdan.grecu, oana.cocarascu}@kcl.ac.uk

Abstract

Large Language Models (LLMs) have demonstrated impressive results in natural language processing (NLP) tasks, however, their brittleness against subtle input perturbations continues to pose a significant challenge. Existing research on robustness has predominantly focused on standard text-based perturbations and the use of invisible characters and homoglyphs, while overlooking the impact of stylized characters increasingly prevalent on social media. To address this, we propose `TinyAttack`, a novel adversarial attack framework designed to exploit vulnerabilities in LLMs through Unicode-based stylistic transformations. `TinyAttack` utilises five Unicode variants to modify the visual rendering of text without altering its underlying semantic or syntactic structure. Our comprehensive evaluation on both open-source (Llama, Mistral, Gemma, Qwen) and closed-source LLMs (Gemini, GPT) demonstrates their susceptibility to these stylized inputs, with performance drops ranging from 29-92% and 6-88.5%, respectively, across all tasks. Our code is available at https://github.com/TRAI-group/TinyAttack.

1 Introduction

Large language models have shown remarkable results in a variety of tasks. Despite this success, their performance drops when exposed to adversarial examples (Formento et al., 2025; Mamta and Cocarascu, 2025a). This susceptibility to adversarial attacks is particularly concerning in applications such as hate speech detection and fact verification where an attacker can subtly manipulate input to bypass safety mechanisms and disseminate misinformation and harmful content. Consequently, such attacks pose a substantial threat to the security and robustness of LLMs and can undermine users' trust in these models.

While most adversarial research has focused on computer vision, textual adversarial attacks pose unique challenges due to the discrete nature of text. The majority of works rely on standard text perturbations at word, character, and sentence-level (Wang et al., 2022a; Goyal et al., 2023; Liu et al., 2023; Gupta et al., 2024), with recent studies utilising invisible characters, homoglyphs, and symbols such as emojis in adversarial and prompt injection attacks (Boucher et al., 2023; Bai et al., 2024).

However, these works ignore the increasing use of unconventional writing styles such as Unicode characters found in real-world user-generated content. In today's world, users make use of creative writing styles that deviate from standard formats. Figure 1 illustrates an example where different Unicode characters are used to render text. An attacker could exploit these unconventional formats to embed malicious content or misinformation to bypass detection systems which are trained predominantly on standardized data. Thus, there is a need to evaluate the robustness of LLMs and understand their behaviour towards these Unicode formats.

In this paper, we propose `TinyAttack`, a novel adversarial attack framework that exploits vulnerabilities in LLMs through Unicode-based stylistic transformations. To guide `TinyAttack`, we use two techniques to estimate the importance of words, Feature Ablation and LLM Self-Attribution, which query the target model to predict the label and generate importance scores for each word, thus minimizing computational overhead. We propose five perturbations to create adversarial datasets, based on superscript, subscript, a combination thereof, ransom note effect, and diacritics.

We conduct a comprehensive analysis of over 500 experiments across various models, settings, and NLP tasks. We evaluate `TinyAttack` on four open-source (Llama-3-8B (Dubey et al., 2024), Mistral-7B (Jiang et al., 2023), Gemma-3-12B (Mesnard et al., 2024), Qwen2-7B (Bai et al.,

*Equal contribution.

Findings of the Association for Computational Linguistics: ACL 2026, pages 39933–39962

Figure 1: Text with various Unicode characters.

2023)) and two closed-source (GPT-3.5 Turbo and Gemini 2.0 Flash) LLMs in zero-shot and few-shot settings, using five datasets carefully chosen to represent a range of natural language understanding (NLU) and natural language inference (NLI) tasks.

Our results highlight significant performance degradation across all LLMs when subjected to `TinyAttack` perturbations. Furthermore, we find that adding adversarial examples does not increase the robustness. These results emphasize the need for LLMs to account for stylistic perturbations such as Unicode-based variations in order to preserve model reliability and ensure safe deployment. To the best of our knowledge, our work is the first to comprehensively evaluate LLMs on these unique style perturbations in adversarial attack settings.

To summarize, our contributions are as follows

- We propose `TinyAttack`, a novel adversarial attack framework that exploits five Unicode-based stylistic transformations to uncover vulnerabilities in LLMs.
- We conduct over 500 experiments to evaluate `TinyAttack` on open-source and closed-source LLMs in zero/few-shot settings on five NLU and NLI tasks.
- We show that adversarial defense using few-shot prompting fails to handle these perturbations, emphasizing the need for LLMs to be more resilient to Unicode transformations.

2 Related Work

2.1 Robustness of NLP Models

Various tasks have been explored to assess the robustness of NLP models: sentiment analysis (Jin et al., 2020; Yuan et al., 2023; Mamta and Cocarascu, 2025b), fact verification (Thorne et al., 2019; Abdelnabi and Fritz, 2023), toxic content detection (Yuan et al., 2023), argument mining (Mayer et al., 2020; Sofi et al., 2022), machine translation (Morris et al., 2020; Sai et al., 2021; Wang et al., 2021), natural language inference (Morris et al., 2020; Wu et al., 2021; Li et al., 2021; Yuan et al., 2023), question answering (Kiela et al., 2021; Yuan et al., 2023; Gupta et al., 2024), and dialogue generation (Sai et al., 2020; Li et al., 2023). Several works have shown that language models are brittle to adversarial inputs (Alzantot et al., 2018; Lin et al., 2021; Neerudu et al., 2023) and recent works on robustness of LLMs have investigated out-of-distribution datasets (Gupta et al., 2024; Yuan et al., 2023), contrast sets, challenge test sets, behavioral testing, and adversarial inputs (Gupta et al., 2024; Mamta and Cocarascu, 2025a).

Perturbations are introduced at various levels: character-level (e.g. character swapping, insertion, deletion, or substitution), word-level (e.g. synonym substitution or contextual perturbations), and sentence-level (e.g. paraphrasing) (Wang et al., 2022b). There are works that apply perturbations in the token embedding space (Wallace et al., 2019; Zou et al., 2023; Geisler et al.; Chacko et al., 2024), however, these approaches do not produce adversarial words that can be discretized into usable words.

To attack a specific word within a sentence, a two-stage process is usually applied: word importance ranking followed by word substitution. The first step involves ranking words based on their influence on the model's predictions, with the method for measuring importance varying based on the level of access to the model. In white-box attacks, where the attacker has full access to the model's architecture and weights, gradient information is used to find the importance of words (Ebrahimi et al., 2018). In contrast, black-box attacks, which simulate more realistic scenarios by only querying the model for outputs, rely on methods such as leave-one-out techniques (Jin et al., 2020) or model explainability (Mamta and Ekbal, 2022).

2.2 Unicode-based Adversarial Attacks

Several studies have used homoglyphs to perturb texts while preserving their original semantics and readability (Dionysiou and Athanasopoulos, 2021; Valle-Aguilera et al., 2024; Cooper et al., 2025). Boucher et al. (2022) showed how subtle Unicode manipulations, such as invisible characters, homoglyphs, reorderings, or deletions, can significantly reduce the performance of both commercial (Microsoft and Google) and open-source (Facebook and IBM) models. Furthermore, Boucher et al. (2023) explored the use of Unicode combining diacritical marks for adversarial attacks, particularly targeting visual text understanding models such as Vision Transformers and OCR systems. While other recent works (Daniel and Pal, 2024; Bai et al.,

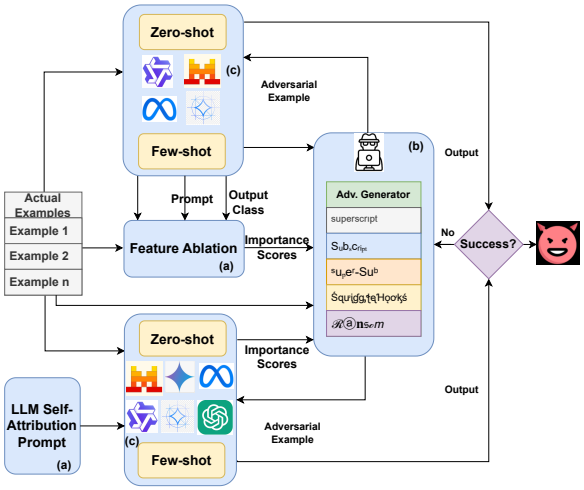

Figure 2: An overview of the TinyAttack framework.

2024) investigate the impact of Unicode characters on jailbreak attacks, these specific security concerns are beyond the scope of our current focus.

TinyAttack uses a broader and more nuanced class of stylistic Unicode transformations that are becoming prevalent in user-generated content to evaluate the adversarial robustness of a range of closed-source and open-source LLMs.

3 Methodology

TinyAttack (see Figure 2) consists of three main components: (a) word importance, (b) an adversarial generator, and (c) a target model. The *Word Importance* component identifies the importance of words using one of two methods: Feature Ablation or LLM Self-Attribution. Subsequently, TinyAttack generates adversarial samples by perturbing words in descending order of their importance using the *Adversarial Generator* component, which are used to attack the *target LLMs*. Next, we describe the task and present the details of our proposed framework.

3.1 Problem Formulation

Given an input X consisting of n tokens $\{x_1, x_2, \ldots, x_n\}$ and ground truth label y, the objective of the attack is to generate a perturbed version X' such that the target model M misclassifies it, i.e. $M(X') \neq y$ (an untargeted attack). For tasks consisting of two inputs $X1$ and $X2$, the adversary applies perturbations to $X1$ (e.g. hypothesis/claim/argument), resulting in a modified input $X1'$, while keeping $X2$ (e.g. premise/evidence/topic) unchanged. The goal remains to mislead the model such that $M(X1', X2) \neq y$. We assume soft-label black-box access of the model where an attacker can access the model's logits but has no access to the model's gradients and weights for open-source models. For closed-source models, we assume black-box access where an attacker can only access the model's output. TinyAttack generates perturbed inputs using five Unicode variants to modify the visual rendering of text without altering its underlying semantic or syntactic structure.

3.2 Word Importance

3.2.1 Feature Ablation

For open-source LLMs, we use Feature Ablation to identify the most important words for the model's predictions. We systematically remove individual words to observe the change in the model's output and compute their importance. Specifically, for an input $X1$ with ground truth label y, we first obtain the model's prediction $M(X1)$ and its probability for y. Then, for each word x_i, we create a new input $X1_{\neg x_i}$ by masking x_i, and query the LLM with $X1_{\neg x_i}$ to obtain $M(X1_{\neg x_i})$. The importance of x_i is measured by the drop in probability of the true label y.

3.2.2 LLM Self-Attribution

To reduce the computational cost of iterative ablation for LLMs, we leverage the instruction-following and reasoning capabilities of the LLMs themselves. This is particularly important when an attacker can access the model via an API, which is typically the case of closed-source models. We employ a prompt-based word importance estimation method, which we refer to as LLM Self-Attribution. Instead of multiple queries, this method requires only two queries, a label prediction query followed by self-attributed word importance scores.[1]

3.3 Adversarial Generator

Once words are ranked by importance using either Feature Ablation or LLM Self-Attribution, TinyAttack applies targeted perturbations as shown in Figure 2. We employ five perturbations based on Unicode stylistic transformations that mimic unconventional writing styles found in real-world user-generated content. These perturbations are label-preserving as they do not modify the underlying semantic or syntactic structure of the input sentence (see Appendix G).

[1]The prompt that instructs the LLM to return the importance score of words in the input sentence is in Appendix D.

Original Text	Adversarial Text	Type
The film seems a dead weight .	The $^{\text{film}}$ $^{\text{seems}}$ a dead weight	Superscript
Used to love it here!	Used to $_{\text{love}}$ it here!	Subscript
He put it together without asking if it was together or separate.	He $_{\text{p}}{}^{\text{u}}{}_{\text{t}}$ it together without asking if it was together or s$^{\text{e}}$par$^{\text{a}}$t$_{\text{e}}$.	Super-Sub
It was a lovely ride.	It was a ļơɣẻļƴ ride.	SquiggleHooks
I also ordered two kids meals.	I *aℓⓈo* *ordered* two kids **meals.**	Ransom

Figure 3: Examples of Unicode-based attacks. The words perturbed are shown in red.

For each input, we iteratively apply these perturbations to the most important words. The attack process continues until one of two conditions is met: either a successful attack is achieved, meaning that the target model M misclassifies the perturbed input $X1'$, i.e. $M(X1') \neq y$, or all important words identified have been perturbed without resulting in a misclassification.

Next, we describe the Unicode perturbations applied by the *Adversarial Generator* (see Figure 3 for examples).[2] We use the typographic variations of superscripts and subscripts. These characters are visually rendered slightly above (superscript) or below (subscript) the normal text. Although commonly used in mathematical formulas or footnotes, their use in user-generated content often serves a stylistic purpose. We define five different attacks:

Superscript. Here, all the characters within an important word are converted to their Unicode superscript equivalents. This forces the LLM to process a sequence of elevated characters that present a tokenization or embedding challenge for the model.

Subscript. Conversely, the subscript attack converts the characters within an important word to their Unicode subscript equivalents. Some characters do not have an existing subscript equivalent and thus are not changed. The characters not changing in this attack are {a, b, c, d, f, g, q, w, y, z}.

Super-Sub. This variant alternates the conversion to superscript and subscript characters within the important word. Note that if the character does not have a subscript equivalent, and is in a position that should be changed to subscript, the character remains unchanged.

Alternating Squiggles and Hooks. This perturbation introduces unique visual accents and flourishes to each letter, inspired by symbols often found in phonetics, linguistics, or as accent marks in various languages. It uses diacritic marks such as circumflexes, hooks, horns, or tildes that traditionally add nuanced inflections to vowels or shape consonant sounds. For an important word identified by our importance estimation methods, we apply a mark above one character, then below the next, and so on. These are alternative Unicode representations that are visually distinct and do not change the word's meaning for a human reader.

Ransom Note Style. This perturbation is inspired by the *ransom note effect* which uses an excessive number of juxtaposed typefaces. Traditionally, the text appears as formed from words and letters cut randomly from various print sources to avoid recognizable handwriting. Ransom Note perturbations combine characters from multiple, visually distinct Unicode blocks within a single word or phrase.

4 Experiments

To expose vulnerabilities in LLMs, we conduct an extensive evaluation of over 500 experiments with `TinyAttack` across various tasks and models.[3]

Datasets. We evaluate our attack framework on five datasets spanning NLU and NLI tasks. We use the SST-3 dataset for sentiment analysis (Sent) (Socher et al., 2013), HSOL dataset for hate speech detection (HS) (Davidson et al., 2017), Climate-FEVER for Fact Verification (FV) (Diggelmann et al., 2020), UKP dataset for argument mining (AM) (Stab et al., 2018), Stanford NLI (SNLI) for NLI tasks (Bowman et al., 2015). We use the complete test set for HS, FV, and AM datasets. For NLI and Sent, we randomly sample 1500 examples (500 per class) from its test split as the original test splits are large.[4]

Target LLMs. We evaluate both open and closed-source models. Open-source LLMs include Meta-Llama-3-8B-Instruct (Dubey et al., 2024), Mistral-7B-Instruct-v0.1 (Jiang et al., 2023), Gemma-3-12b-it (Kamath et al., 2025), Qwen2-7B-Instruct. Closed-source LLMs include GPT-3.5 Turbo Instruct via OpenAI API and Gemini 2.0 Flash via Gemini API (Anil et al., 2023). All experiments were conducted in zero- and few-shot settings. In the few-shot setting, we provide 6 in-context examples (2 per class for each task) to assess whether we can enhance the LLM's robustness.

[2]All perturbations are generated using https://yaytext.com

[3]Implementation details can be found in Appendix B.

[4]Details about the datasets are in Appendix C.

			Superscript		Subscript		Super-Sub		Squiggle&Hooks		Ransom	
Task	**Model**	**Orig**	**FA**	**LSA**	**FA**	**LSA**	**FA**	**LSA**	**FA**	**LSA**	**FA**	**LSA**
Sent	**Llama**	69.95	89.6	91.5	82.5	87.0	87.2	90.8	90.5	86.7	84.8	91.9
	Mistral	71.01	86.4	79.8	82.8	78.8	85.6	80.6	81.5	84.2	79.8	84.3
	Gemma	77.1	56.6	57.6	78.6	78.9	79.7	70.0	87.6	89.9	85.5	42.4
	Qwen	74.55	45.7	40.9	85.1	79.7	72.6	70.3	90.2	84.9	71.1	72.5
	Gemini	**79.37**	-	15.8	-	20.1	-	21.8	-	68.6	-	22.5
	GPT 3.5	70.79	-	52.7	-	76.5	-	74.0	-	81.3	-	81.0
HS	**Llama**	67.68	69.5	-	72.7	-	76.9	-	77.3	-	81.0	-
	Mistral	70.25	68.0	71.5	69.3	68.0	74.8	72.8	76.0	75.0	81.3	82.3
	Gemma	72.12	46.2	30.3	53.8	69.2	52.2	60.0	69.9	77.9	58.0	56.4
	Qwen	67.4	48.8	50.3	78.1	75.3	68.2	69.4	82.7	77.9	66.9	65.1
	Gemini	71.15	-	18.6	-	18.4	-	21.0	-	50.2	-	25.0
	GPT 3.5	64.04	-	48.2	-	78.9	-	77.8	-	85.0	-	86.3
FV	**Llama**	50.41	53.0	84.3	73.1	79.6	71.3	81.2	79.8	77.9	75.5	82.7
	Mistral	55.99	73.7	78.6	73.3	72.0	76.0	76.4	73.5	78.0	74.5	77.3
	Gemma	56.22	27.4	27.7	70.7	70.1	51.5	60.2	82.8	72.0	59.0	44.2
	Qwen	54.42	45.0	44.9	72.7	72.4	62.9	62.3	74.6	79.5	66.6	62.2
	Gemini	58.02	-	36.5	-	36.6	-	38.9	-	60.3	-	39.2
	GPT 3.5	39.08	-	65.6	-	78.2	-	75.2	-	84.0	-	81.5
AM	**Llama**	61.28	70.9	88.3	75.0	88.0	80.9	88.4	82.2	86.8	81.0	86.0
	Mistral	62.39	78.1	74.7	77.8	70.6	77.6	73.7	88.6	74.0	86.4	82.5
	Gemma	73.65	53.4	21.6	58.6	79.0	58.5	60.9	56.4	75.7	66.4	35.6
	Qwen	67.14	63.7	36.0	68.6	77.3	66.2	67.0	72.4	81.9	66.3	66.5
	Gemini	59.23	-	6.3	-	7.3	-	13.3	-	50.8	-	15.6
	GPT 3.5	56.88	-	80.1	-	83.5	-	83.5	-	88.5	-	84.9
NLI	**Llama**	41.48	71.9	77.5	70.7	76.9	69.1	83.1	75.6	79.7	69.9	81.7
	Mistral	56.4	83.4	82.9	84.3	82.7	82.8	79.9	81.9	79.3	78.9	79.4
	Gemma	60.55	52.5	8.5	60.5	65.2	60.8	79.4	51.4	78.2	52.9	79.3
	Qwen	80.29	29.3	21.2	76.4	76.1	56.3	57.5	85.0	88.8	54.5	55.3
	Gemini	60.04	-	13.0	-	64.7	-	15.9	-	51.5	-	14.6
	GPT 3.5	16.67	-	-	-	-	-	-	-	-	-	-

Table 1: Percentage Drop in F_1 (PDF) in the zero-shot setting using Feature Ablation (FA) and LLM Self-Attribution (LSA) for computing word importance. Here, Orig represents the F_1 score on the original test set.

5 Results and Discussion

To evaluate the robustness of LLMs against `TinyAttack`, we consider accuracy, F_1, Perturbation Ratio (PR), and Percentage Drop in F_1 (PDF).[5] PR represents the average fraction of words perturbed per adversarial example (successful or unsuccessful attack); lower rates indicate more effective attacks.

$$\text{PR} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\text{PerturbedWords}_i|}{|\text{TotalWords}_i|} \quad (1)$$

PDF is defined as follows, where $F_{1_{orig}}$ is the F_1 score on the original test set and F_{1_p} is the F_1 score after applying perturbation p:

$$PDF = ((F_{1_{orig}} - F_{1_p})/F_{1_{orig}}) * 100 \quad (2)$$

Which model is more/least robust among open-source models? Table 1 shows PDF values in the zero-shot setting using Feature Ablation and Self-Attribution for computing the importance of words. We observe that Gemma, Gemini, and Qwen perform better on the original test set compared to the other models. In addition, Gemma and Qwen outperform Gemini on HS, AM, and NLI tasks. Llama performs the worst in almost all tasks.

All LLMs exhibit a significant drop in F_1 for all Unicode perturbations, showing their vulnerability. Qwen and Gemma consistently exhibit greater robustness across multiple tasks and perturbations. This is evident from their relatively lower PDF values compared to their F_1 scores on the original test set. Qwen achieves the lowest PDF values compared to other models, whilst Gemma shows more robustness on the FV and AM tasks. Llama is more vulnerable compared to other models to almost all perturbations. Mistral performs reasonably well compared to Llama but is less robust compared to Gemma and Qwen. For the superscript attack, Qwen and Gemma require a higher number of perturbations compared to other models. However, for squiggle & hooks attacks, they require fewer perturbations to execute a successful attack. Table 2 shows PR in the zero-shot setting.

We also evaluate whether incorporating in-context examples to the prompt can help increase robustness. Table 3 shows that under few-shot set-

[5]Appendix E has the complete set of results.

Task		Super PR	Sub PR	Super-Sub PR	Sq&Hook PR	Ransom PR
Sent	Llama	0.52	0.56	0.53	0.51	0.52
	Mistral	0.51	0.53	0.52	0.51	0.41
	Gemma	0.49	0.46	0.45	0.36	0.48
	Qwen	0.69	0.44	0.52	0.39	0.52
HS	Llama	0.64	0.64	0.63	0.6	0.6
	Mistral	0.55	0.54	0.52	0.51	0.49
	Gemma	0.69	0.61	0.66	0.58	0.65
	Qwen	0.69	0.56	0.61	0.51	0.58
FV	Llama	0.58	0.59	0.56	0.52	0.56
	Mistral	0.54	0.56	0.54	0.55	0.53
	Gemma	0.74	0.51	0.49	0.1	0.41
	Qwen	0.64	0.5	0.54	0.46	0.54
AM	Llama	0.52	0.51	0.49	0.46	0.42
	Mistral	0.55	0.5	0.49	0.4	0.46
	Gemma	0.62	0.66	0.69	0.6	0.65
	Qwen	0.58	0.54	0.51	0.52	0.54
NLI	Llama	0.63	0.66	0.65	0.57	0.64
	Mistral	0.49	0.53	0.51	0.49	0.5
	Gemma	0.57	0.56	0.56	0.53	0.55
	Qwen	0.79	0.54	0.65	0.46	0.65

Table 2: Perturbation Ratio (PR) across all attacks in the zero-shot setting using Feature Ablation.

ting, Gemma and Qwen achieve high performance on the original test set compared to other open-source models. Despite this, all models are still vulnerable to all perturbations as seen by the high PDF value. Similar to zero-shot, Gemma and Qwen are more robust compared to all other models in the few-shot setting. We also evaluate the effect of different perturbation ratios on all models for the NLI task. Results are presented in Appendix A.3.

Which perturbations are more challenging? A significant drop in F_1 is observed across all Unicode stylistic perturbations, models, and tasks, indicating greater vulnerability to these perturbations. In the zero-shot setting, the most challenging perturbations for Llama are squiggle & hooks and ransom, which result in the largest performance drops. Other models such as Gemini, GPT, Mistral, Gemma, and Qwen exhibit substantial degradation under squiggle & hooks perturbation.

In terms of PR, superscript, subscript, and super-sub require a higher number of word perturbations as shown in Table 2. In contrast squiggle & hooks and ransom achieve large drop in performance by perturbing fewer words. In the few-shot setting, superscript requires a higher number of word perturbations (high PR) and squiggle & hooks requires fewer perturbations (low PR) to execute a successful attack (see Appendix E).

How do open-source models compare to closed-source models? In both zero- and few-shot settings, Gemini is robust to superscript, subscript, ransom, and super-sub attacks, which is evident from the low drop in performance (PDF). For squiggle & hooks, there is a significant drop in PDF, but significantly less compared to open-source models. Furthermore, Gemini requires a large number of word perturbations (high PR), indicating its robustness to these Unicode stylistic variations. The performance of GPT 3.5 varies across tasks. For example, in the few-shot Sent task, GPT 3.5 is robust compared to open-source models, however, for HS, it is more vulnerable to certain attacks compared to open-source models. Interestingly, GPT 3.5 predicts only the entailment class for the NLI task in both zero-shot and few-shot settings.

Can we rely on LLM Self-Attribution? In the zero-shot setting, the Feature Ablation method results in a high PDF for most attacks, with similar values when using LLM Self-Attribution. In the few-shot setting, LLM Self-Attribution shows a high performance drop for most attacks. This suggests that LLM Self-Attribution becomes more effective when we provide in-context examples that increase the contextual understanding. Thus, LLM Self-Attribution can be a better alternative compared to Feature Ablation which is computationally expensive. We note that Llama refused to provide word importance scores in the case of HS due to its ethical guidelines.

How do `TinyAttack` perturbations compare to other perturbations? We compare `TinyAttack` perturbations with standard character-based perturbations such as insertion and deletion, word-level perturbations (synonyms replacements), and sentence based perturbations (paraphrasing) (Mamta and Cocarascu, 2025a). Table 4a shows the PDF values for sentiment and NLI tasks. We see that `TinyAttack` perturbations are more challenging than standard character-, word-, and sentence-level perturbations, as shown by the smaller drop in F_1 (PDF). Moreover, they require a larger number of perturbations to execute a successful attack, resulting in a higher PR (see Appendix A.2). We also compare `TinyAttack` with invisible characters and deletion attacks (see Appendix A.2). We also conducted a human evaluation to compare the readability of standard perturbations with `TinyAttack` perturbations, confirming that the Unicode-based perturbations are human-readable and label-preserving, as well as consistently more readable than character insertion and deletion (see Appendix G for details).

			Superscript		Subscript		Super-Sub		Squiggle&Hooks		Ransom	
Task	**Model**	**Orig**	**FA**	**LSA**	**FA**	**LSA**	**FA**	**LSA**	**FA**	**LSA**	**FA**	**LSA**
Sent	**Llama**	70.73	79.1	92.4	76.1	88.0	79.2	91.9	86.0	90.7	75.0	91.8
	Mistral	72.73	84.8	83.8	84.6	76.3	86.8	81.8	83.1	82.1	78.1	84.5
	Gemma	75.67	57.9	44.2	64.7	78.0	66.7	69.9	89.0	88.7	48.1	44.2
	Qwen	74.73	69.0	43.5	83.0	77.7	70.3	70.2	85.3	80.1	69.7	71.5
	Gemini	79.72	-	18.0	-	21.0	-	22.2	-	61.9	-	20.7
	GPT 3.5	71.15	-	43.6	-	67.7	-	63.8	-	72.0	-	71.8
HS	**Llama**	65.90	65.3	-	72.6	-	74.5	-	71.3	-	72.7	-
	Mistral	70.46	68.5	78.5	71.0	77.5	90.6	83.5	78.5	86.2	79.7	85.4
	Gemma	77.93	48.4	38.0	52.9	71.6	61.0	63.6	80.2	77.8	55.4	59.6
	Qwen	71.37	48.8	48.6	68.4	67.5	66.5	63.3	76.2	69.9	65.0	62.9
	Gemini	75.57	-	19.0	-	21.7	-	22.2	-	36.1	-	25.3
	GPT 3.5	69.22	-	47.0	-	75.2	-	74.9	-	87.9	-	86.7
FV	**Llama**	51.73	78.3	85.3	74.2	80.8	76.9	80.9	83.0	73.9	82.3	84.2
	Mistral	58.97	78.5	80.4	82.7	74.4	79.8	84.5	79.7	84.3	77.2	80.9
	Gemma	63.25	37.5	31.7	70.7	71.8	62.0	61.4	75.6	73.4	37.8	40.9
	Qwen	57.14	66.1	53.7	73.3	57.6	66.7	39.0	75.7	75.7	69.3	33.1
	Gemini	61.17	-	29.4	-	70.8	-	60.1	-	72.5	-	38.8
	GPT 3.5	45.17	-	61.0	-	71.0	-	70.0	-	77.9	-	75.0
AM	**Llama**	64.28	60.8	82.6	69.4	82.3	76.5	82.7	71.1	78.9	80.8	84.6
	Mistral	62.01	68.5	81.9	74.0	80.1	71.5	80.8	75.4	81.1	77.3	80.0
	Gemma	74.79	52.7	57.1	56.7	64.8	57.5	60.9	54.4	75.1	48.3	38.5
	Qwen	71.42	60.2	64.9	65.5	69.2	64.0	68.6	77.2	79.6	67.5	68.0
	Gemini	73.35	-	21.8	-	23.2	-	25.5	-	37.5	-	27.6
	GPT 3.5	59.02	-	72.4	-	76.0	-	77.3	-	79.6	-	75.3
NLI	**Llama**	46.2	73.9	88.7	73.7	84.7	73.3	87.4	72.7	85.3	73.0	88.7
	Mistral	64.53	87.5	83.2	87.7	82.7	87.6	82.8	85.9	83.1	87.0	82.4
	Gemma	76.33	60.4	11.4	61.9	79.2	62.8	67.2	64.1	85.1	57.8	41.8
	Qwen	85.67	31.9	24.1	76.2	80.6	56.9	60.0	85.5	88.2	56.1	58.2
	Gemini	72.5	-	13.1	-	16.4	-	18.3	-	57.8	-	17.9
	GPT 3.5	16.67	-	-	-	-	-	-	-	-	-	-

Table 3: Percentage Drop in F_1 (PDF) in the few-shot setting using Feature Ablation (FA) and LLM Self-Attribution (LSA) for computing word importance. Here, Orig represents the F_1 score on the original test set.

5.1 Qualitative Analysis

How do models tokenize `TinyAttack` variations? To understand the reason behind the significant drop in performance, we examine the tokens generated by Llama and Gemma for `TinyAttack` perturbations. Since these perturbations use Unicode characters, they typically do not exist in the model's vocabulary. As a result, the tokenizer either splits them into multiple subword tokens or encodes them as byte sequences (unknown tokens). This makes it difficult for the model to map the perturbed input to its original equivalent. Consequently, the model struggles to interpret the input correctly, leading to a drop in performance. This behavior highlights a key vulnerability in current tokenization strategies when handling stylistically altered text. Figure 4 shows the tokenization of Llama and Gemma. It can be seen that both Llama and Gemma do not understand these perturbations, resulting into multiple tokens. However, we observe that the count of tokens generated by Gemma is significantly lower than Llama except for subscript attack. Gemma can tokenize a few stylized characters as individual tokens, which indicates the presence of these unicode characters in its vocabulary. Similarly, detailed analysis of other models can be found in Appendix F.

Feature Ablation vs LLM Self-Attribution. We also qualitatively compare the word importance methods, Feature Ablation and LLM Self-Attribution. Figure 5 shows adversarial examples generated by applying ransom and squiggle & hooks attacks to Sent and NLI tasks as well as the order in which words are perturbed by the two methods. In the first example, LLM Self-Attribution requires only 2 perturbations to execute a successful attack, whereas Feature Ablation requires 10 perturbations. In contrast, in the second example Feature Ablation requires only 1 perturbation, whereas LLM Self-Attribution requires 4. Finally, in the third example, both methods perform equally. Thus, we can hypothesize that LLM Self-Attributions are reliable and can be used as a cost-effective alternative to Feature Ablation.

Ablation Study. We perform an ablation experiment by removing the Feature Ablation and Self-

	Model	Orig	Ins PDF	Del PDF	Reph PDF	Syn PDF
Sent	Llama	69.95	54.78	59.86	9.74	7.36
	Mistral	71.01	37.84	48.26	16.56	14.24
	Gemma	77.1	39.88	50.04	6.04	4.94
	Qwen	74.55	32.61	51.59	7.02	4.20
	Gemini	79.37	37.86	47.07	7.54	6.09
	GPT 3.5	70.79	39.27	51.42	12.06	10.75
NLI	Llama	41.48	72.30	66.27	21.79	6.65
	Mistral	56.4	50.73	55.89	14.96	5.85
	Gemma	60.55	54.68	51.74	6.77	2.36
	Qwen	80.29	20.71	35.72	2.96	0.54
	Gemini	60.04	49.05	49.78	5.22	2.61
	GPT 3.5	16.67	-	-	-	-

(a) Standard character-based perturbations.

	Model	Orig	Super PR	Super PDF	Sub PR	Sub PDF	Super-Sub PR	Super-Sub PDF	Sq&Hook PR	Sq&Hook PDF	Ransom PR	Ransom PDF
Sent	Llama	69.95	0.54	89.46	0.57	83.52	0.54	88.25	0.53	85.85	0.56	86.52
	Mistral	71.01	0.56	80.48	0.58	76.29	0.58	79.61	0.56	81.3	0.55	81.21
	Gemma	77.1	0.88	21.04	0.63	79.04	0.69	66.15	0.56	85.45	0.79	41.65
	Qwen	74.55	0.79	40.32	0.61	78.12	0.67	68.50	0.57	84.8	0.65	68.38
	Gemini	79.37	0.90	15.75	0.88	18.87	0.96	7.65	0.69	67.85	0.87	22.02
	GPT 3.5	70.79	0.66	51.12	0.58	75.42	0.58	70.91	0.50	81.68	0.55	79.8
NLI	Llama	41.48	0.58	84.31	0.64	75.92	0.59	79.63	0.46	82.93	0.58	82.52
	Mistral	56.4	0.44	83.78	0.46	84.13	0.44	83.72	0.42	83.53	0.44	84.01
	Gemma	60.55	0.82	10.04	0.66	64.38	0.71	50.85	0.54	78.41	0.77	29.46
	Qwen	80.29	0.89	19.02	0.71	73.84	0.78	54.41	0.64	82.41	0.79	51.19
	Gemini	60.04	0.94	14.64	0.93	15.31	0.93	14.96	0.82	47.17	0.94	13.81
	GPT 3.5	16.67	-	-	-	-	-	-	-	-	-	-

(b) Random word ordering.

Table 4: Results (zero-shot) comparing standard character-based perturbations (a) and random word ordering (b).

Attack	Example	Llama	Gemma	L / G
-	I am not crazy about the smoothies though.	['I', 'Ġam', 'Ġnot', 'Ġcrazy', 'Ġabout', 'Ġthe', 'Ġsmooth', 'ies', 'Ġthough', '.']	['I', '_am', '_not', '_crazy', '_about', '_the', '_smoothies', '_though', '.']	10 / 9
Super	I am not crazy about the smoothies though.	['á', '˙', 'µ', 'Ġá', 'µ', 'ĥ', 'á', 'µ', 'IJ', 'Ġâģ', '¿', 'á', 'µ', 'J̊', 'á', 'µ', 'Ł', 'Ġcrazy', 'Ġabout', 'Ġá', 'µ', 'Ł', 'É', '°', 'á', 'µ', 'T', 'Ġ', 'É', 'ę', 'á', 'µ', 'IJ', 'á', 'µ', 'J̊', 'á', 'µ', 'J̊', 'á', 'µ', 'Ł', 'É', '°', 'á', '¶', 'ı', 'á', 'µ', 'T', 'É', 'ę', 'Ġá', 'µ', 'Ł', 'É', '°', 'á', 'µ', 'J̊', 'á', 'µ', 'ĺ', 'á', 'µ', 'ţ', 'É', '°', '.']	['I', '_a', '<0xE2>', '<0x82>', '<0x98>', '_', '<0xE2>', '<0x82>', '<0x99>', '<0xE2>', '<0x82>', '<0x92>', '_c', '<0xE1>', '<0xB5>', '<0xA3>', 'azy', '_ab', '<0xE2>', '<0x82>', '<0x92>', '<0xE1>', '<0xB5>', '<0xA4>', '_', '<0xE2>', '<0x82>', '<0x91>', '_', '<0xE2>', '<0x82>', '<0x9B>', '<0xE2>', '<0x82>', '<0x98>', '<0xE2>', '<0x82>', '<0x92>', '<0xE2>', '<0x82>', '<0x92>', '<0xE2>', '<0x82>', '<0x91>', '<0xE2>', '<0x82>', '<0x9B>', '_', '<0xE2>', '<0x82>', '<0x92>', '<0xE1>', '<0xB5>', '<0xA4>', 'g', '.']	69 / 32
Sub	I am not crazy about the smoothies though.	['I', 'Ġa', 'âĤ', 'Ī', 'ĠâĤ', 'Ļ', 'âĤ', 'J̊', 'âĤ', 'Ī', 'Ġc', 'á', 'µ', '£', 'azy', 'Ġab', 'âĤ', 'J̊', 'á', 'µ', 'ª', 'âĤ', 'Ī', 'ĠâĤ', 'Ī', 'âĤ', 'ĸ', 'âĤ', 'ij', 'ĠâĤ', 'Ī', 'âĤ', 'J̊', 'âĤ', 'Ī', 'âĤ', 'ĸ', 'á', 'µ', 'ę', 'âĤ', 'ij', 'âĤ', 'Ľ', 'ĠâĤ', 'Ī', 'âĤ', 'ĸ', 'âĤ', 'J̊', 'á', 'µ', 'ª', 'g', 'âĤ', 'ĸ', '.']	[' ', '_a', '<0xE2>', '<0x82>', '<0x98>', '_', '<0xE2>', '<0x82>', '<0x99>', '<0xE2>', '<0x82>', '<0x92>', '_c', '<0xE1>', '<0xB5>', '<0xA3>', 'azy', '_ab', '<0xE2>', '<0x82>', '<0x92>', '<0xE1>', '<0xB5>', '<0xA4>', '_', '<0xE2>', '<0x82>', '<0x91>', '_', '<0xE2>', '<0x82>', '<0x9B>', '<0xE2>', '<0x82>', '<0x98>', '<0xE2>', '<0x82>', '<0x92>', '<0xE2>', '<0x82>', '<0x92>', '<0xE2>', '<0x82>', '<0x91>', '<0xE2>', '<0x82>', '<0x9B>', '_', '<0xE2>', '<0x82>', '<0x92>', '<0xE1>', '<0xB5>', '<0xA4>', 'g', '.']	61 / 66
Super-Sub	I am not crazy about the smoothies though.	['á', '˙', 'µ', 'Ġam', 'Ġn', 'âĤ', 'J̊', 't', 'Ġc', 'á', 'µ', '£', 'a', 'á', '¶', '»', 'y', 'Ġa', 'á', 'µ', 'T', 'o', 'á', 'µ', '¤', 't', 'Ġt', 'âĤ', 'ĸ', 'e', 'Ġs', 'âĤ', 'Ī', 'o', 'á', 'µ', 'J̊', 't', 'âĤ', 'ĸ', 'ï', 'á', 'µ', 'T', 's', 'Ġt', 'É', '°', 'o', 'á', 'µ', '¤', 'g', 'É', '°', '.']	['I', 'Ì', '¢', 'Ġá', '¶', 'ı', 'á', ',', '¿', 'Ġá½', 'µ', 'Ç', '«', 'Ä¥', 'Ġcrazy', 'Ġabout', 'ĠÁ', '¥', 'á', '±', '˙', 'á°»', 'ĠÁL', 'á', '¶', 'Ī', 'Æį', 'Ç', '«', 'Ä¥', 'á', '±', '˙', 'á»Ī', 'á', '¶', 'J̊', 'ÁL', 'ĠÁ', '¥', 'á', '±', '˙', 'Æį', 'á', '¶', 'Ļ', 'É', 'ª', 'á', '±', '˙', '.']	['I', ''', 'a', 'm', '_', '»', 'o', 't', '_crazy', '_about', '_', 'ı', 'h', 'e', '_', 's', '_', 'Ì', 'h', 'o', '<0xE1>', '<0xB6>', '<0x92>', 's', '_', 't', 'h', 'o', '<0xE1>', '<0xB6>', '<0x99>', 'g', 'h', '.']	53 / 40
Ransom	I am not crazy about the smoothies though.	['ðŁ', 'IJ', 'Ī', 'ĠðŁ', 'IJ', 'ĺ', 'ðŁ', 'Ł', '°', 'ĠðŁ', 'ĵ', 'ĥ', 'ðŁ', 'ĺ', '°', 'ðŁ', 'IJ', 'N', 'Ġcrazy', 'Ġabout', 'ĠðŁ', 'ij', 'ĺ', 'ðŁ', 'Ł', 'Ł', 'ðŁ', 'J̊', 'Ī', 'ĠðŁ', 'ĵ', 'Ī', 'ðŁ', 'Ķ', '™', 'ðŁ', 'Ķ', '¬', 'âĦ¢', '˙', 'ðŁ', 'Ł', '©', 'Ł', 'k', 'ĵ', 'ðŁ', 'k', 'Ī', 'ðŁ', 'ĦĪ', '°', 'ðŁ', 'Ł', 'ĠðŁ', 'ĵ', 'Ł', 'ðŁ', 'ĥ', 'Ł', 'ðŁ', 'Ī', '°', 'ðŁ', 'ĥ', '¤', 'ðŁ', 'Ķ', '¤', 'ðŁ', 'Ł', 'ĵ', 'ij', '.']	['I', ''', 'a', 'm', '_', '»', 'o', 't', '_crazy', '_about', '_', 'ı', 'h', 'e', '_', 's', '_', 'Ì', 'h', 'o', '<0xE2>', '<0x84>', '<0xB4>', 'f', '<0xF0>', '<0x9D>', '<0x96>', '<0x8D>', 'ı', '[E]', 's', '_', 't', '<0xF0>', '<0x9F>', '<0x85>', '<0x97>', 'o', '<0xF0>', '<0x9F>', '<0x85>', '<0xA4>', '<0xF0>', '<0x9D>', '<0x94>', '<0xA4>', 'h', '.']	73 / 46
Upside-down	I am ɥou crazy about the smoothies though.	['I', 'Ġam', 'Ġ', 'É', 'ſ', 'ou', 'Ġcrazy', 'Ġabout', 'Ġthe', 'Ġsmooth', 'ies', 'Ġthough', '.']	['I', '_am', '_', 'ſ', 'ou', '_crazy', '_about', '_the', '_smoothies', '_though', '.']	13 / 11

Figure 4: Tokens generated by Llama (L) and Gemma (G). The last column shows the count of subword tokens.

Attribution components and perturbing words in random order until the attack is successful. The results are presented in Table 4b. We observe that random ordering leads to a higher PR across all attacks and models compared to our proposed approaches, illustrating the effectiveness of our Feature Ablation and Self-Attribution methods.

5.2 Adversarial Defense

In-context augmentation. We generate adversarial data by applying perturbations to examples from the training set and augmenting them with the actual examples. For each class, we include two adversarial examples along with their corresponding original examples as in-context examples. In total, the few-shot prompt consists of 12 examples, 6 original and 6 adversarial. Table 5 shows that adding adversarial examples to the few-shot prompt does not increase the model robustness. In contrast, we observe a high PDF for sentiment and NLI tasks. A possible reason could be the noise introduced in the in-context examples, as the model struggles to properly tokenize perturbed inputs which disrupts the contextual understanding.

Unicode Normalization. We normalize Unicode characters before passing the input to models using two methods: prompting LLMs to determine which characters should be replaced with Latin letters as in Cooper et al. (2025) (see prompt in Figure 25 in Appendix) and NFKC[6]. Results are shown in Table 5. We observe that LLM-based normalization improves model robustness against superscript and subscript attacks to some extent, but fails for Llama and Mistral. Only Gemini sees improved robustness from normalization across all attacks. This suggests that simple normalization-based defenses are insufficient.

Table 5 also shows that the defensive capability of NFKC is limited to superscript and subscript attacks where compatibility decomposition effectively maps these variants back to their canonical forms. NFKC does not address more sophisticated transformations such as squiggle or ransom style perturbations which exploit combining marks that

[6]https://docs.python.org/3/library/unicodedata.html

Original Example	Task	Gold/Pred	Adv Example FA	Word Order FA	Adv example LSA	Word Order LSA	FA/LSA
So we ended walking all the way back to Harrah's and had a great time there.	Sent	pos/pos	So we ended walking all the way back to Harrah's and had a great time there.	Harrah's, there, great, a, had, So, and, way, walking, to, the, we, back, all, ended, time **(10)**	So we ended walking all the way back to Harrah's and had a great time there.	great, time **(2)**	neg/neu
Now that the buzz (pun totally intended) is that eating local honey helps with allergies, why not incorporate it into my every day diet??	Sent	pos/pos	Now that the buzz (pun totally intended) is that eating local honey helps with allergies, why not incorporate it into my every day diet??	why **(1)**	Now that the buzz (pun totally intended) is that eating local honey helps with allergies, why not incorporate it into my every day diet??	eating, honey, helps, buzz **(4)**	neu/neu
Premise: This church choir sings to the masses as they sing joyous songs from the book at a church. **Hypothesis**: the church is filled with song.	NLI	entail/entail	The church is filled with song.	song **(1)**	The church is filled with song.	song **(1)**	neu/neu

Figure 5: Examples comparing word importance from Feature Ablation (FA) and Self-Attribution (LSA) for Qwen. Here, Pred: prediction and Adv: adversarial.

			Super			Sub			Super-sub			Squiggle			Ransom		
Task	**Model**	**F1**	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3
Sent	**Llama**	70.53	93.8	83.9	18.54	88.1	67.7	6.90	92.8	77.7	13.27	94.2	92.6	86.32	92.4	86.1	44.80
	Mistral	72.19	84.2	84.2	9.06	76.6	81.5	4.06	81.9	84.7	5.17	83.6	86.8	83.90	83.8	87.4	35.05
	Gemma	75.61	21	10.2	10.43	75.1	26.6	9.11	66.3	29.7	9.42	88.4	81.5	89.13	43	23.2	22.04
	Qwen	75.08	39.9	20.9	13.27	76.5	81.0	8.49	64.5	66.2	11.51	82.9	90.7	84.19	69.3	75.5	29.85
	Gemini	79.72	21.1	5.3	8.74	23.6	4.4	7.87	24	5.3	9.44	55.7	27.2	54.11	25.3	21.4	15.30
	GPT 3.5	71.15	42.7	16.3	19.88	67	26.5	13.41	61.3	25.9	17.15	73.4	79.5	81.21	70.7	37.7	30.23
NLI	**Llama**	40.58	84.2	84.0	30.16	83	77.4	23.02	83.8	79.1	25.89	84.3	82.5	82.43	86.3	82.7	46.41
	Mistral	63.46	82.2	86.0	17.55	81.7	86.6	12.62	81.1	86.1	13.95	82.4	84.9	82.36	81.4	86.5	41.10
	Gemma	75.89	8	11.3	6.94	72.4	18.9	6.82	51.6	19.1	7.05	84.2	79.1	78.46	27.8	14.1	19.50
	Qwen	85.55	20.1	8.5	5.19	72.6	81.5	6.86	52	61.6	6.44	89.5	93.2	29.90	53.4	72.6	20.31
	Gemini	72.5	8.8	3.4	7.30	11.7	3.4	3.01	10.7	3.7	5.61	34.2	16.4	45.09	15.5	4.2	11.16
	GPT 3.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 5: PDF results for Sent and NLI tasks using in-context augmentation (T1), unicode normalization using LLMs (T2) and NFKC (T3). F_1 represents the score on the original test set.

are not resolved through compatibility normalization. This demonstrates that although NFKC provides a useful baseline for unicode normalization, it is insufficient as a comprehensive defense mechanism against all Unicode-based adversarial attacks. This underscores the need to develop more comprehensive defense strategies.

6 Conclusion

In this paper, we identified a gap in the security and reliability of LLMs. We developed `TinyAttack`, a novel adversarial framework that exposes LLMs' vulnerabilities to stylistic and human-perceptible perturbations. We designed five Unicode-based adversarial perturbations and evaluated the robustness of six LLMs in zero- and few-shot settings. Our extensive experiments highlighted a significant and consistent susceptibility to these types of attacks, attributed to the models' internal tokenization mechanisms and the lack of training on such varied stylistic perturbations. Future work includes exploring robust defense frameworks to increase LLMs resilience towards `TinyAttack` perturbations and investigating the performance of `TinyAttack` to multimodal LLMs.

Limitations

While our current experiments focus on English, the proposed transformations can generally be applied safely to other languages using the Roman script, as they preserve meaning and readability. However, we acknowledge that in some languages, certain transformations such as diacritics that carry semantic meaning may affect semantics or readability and may therefore require language-specific adjustments. In addition, this work relies on in-context and normalization-based techniques to enhance model robustness. However, in-context augmentation fails against `TinyAttack` perturbations, while normalization improves robustness

only against superscript and subscript attacks. Although it shows consistent gains for the Gemini model, it is less effective for Llama and Mistral. Overall, neither technique is sufficient to defend against the `TinyAttack` perturbations. In future work, we plan to explore robust defense frameworks to increase the resilience of LLMs towards the `TinyAttack` perturbations. Moreover, our work focuses exclusively on textual data, in future, we aim to extend this approach to multimodal LLMs.

Ethics Statement

We use publicly accessible datasets for our experiments, strictly for academic purposes and in full accordance with their licensing terms.

Acknowledgments

This research was supported by EPSRC (grant number EP/X04162X/1).

References

Sahar Abdelnabi and Mario Fritz. 2023. Fact-saboteurs: A taxonomy of evidence manipulation attacks against fact-verification systems. In *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*, pages 6719–6736. USENIX Association.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, and 33 others. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yang Bai, Ge Pei, Jindong Gu, Yong Yang, and Xingjun Ma. 2024. Special characters attack: Toward scalable training data extraction from large language models. *arXiv preprint arXiv:2405.05990*.

Nicholas Boucher, Jenny Blessing, Ilia Shumailov, Ross Anderson, and Nicolas Papernot. 2023. When vision fails: Text attacks against vit and ocr. *arXiv preprint arXiv:2306.07033*.

Nicholas Boucher, Ilia Shumailov, Ross Anderson, and Nicolas Papernot. 2022. Bad characters: Imperceptible nlp attacks. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1987–2004. IEEE.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Samuel Jacob Chacko, Sajib Biswas, Chashi Mahiul Islam, Fatema Tabassum Liza, and Xiuwen Liu. 2024. Adversarial attacks on large language models using regularized relaxation. *arXiv preprint arXiv:2410.19160*.

Portia Cooper, Eduardo Blanco, and Mihai Surdeanu. 2025. The lies characters tell: Utilizing large language models to normalize adversarial Unicode perturbations. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18932–18944, Vienna, Austria. Association for Computational Linguistics.

Johan S Daniel and Anand Pal. 2024. Impact of non-standard unicode characters on security and comprehension in large language models. *arXiv preprint arXiv:2405.14490*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

Thomas Diggelmann, Jordan L. Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. CLIMATE-FEVER: A dataset for verification of real-world climate claims. *CoRR*, abs/2012.00614.

Antreas Dionysiou and Elias Athanasopoulos. 2021. Unicode evil: Evading nlp systems using visual similarities of text characters. In *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, pages 1–12.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Brian Formento, Chuan-Sheng Foo, and See-Kiong Ng. 2025. Confidence elicitation: A new attack vector for large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Simon Geisler, Tom Wollschlager, MHI Abdalla, Johannes Gasteiger, and Stephan Gunnemann. Attacking large language models with projected gradient descent. In *ICML 2024 Next Generation of AI Safety Workshop*.

Shreya Goyal, Sumanth Doddapaneni, Mitesh M. Khapra, and Balaraman Ravindran. 2023. A survey of adversarial defenses and robustness in NLP. *ACM Comput. Surv.*, 55(14s):332:1–332:39.

Ashim Gupta, Rishanth Rajendhran, Nathan Stringham, Vivek Srikumar, and Ana Marasovic. 2024. Whispers of doubt amidst echoes of triumph in NLP robustness. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5533–5590, Mexico City, Mexico. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-Bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, and 79 others. 2025. Gemma 3 technical report. *CoRR*, abs/2503.19786.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5053–5069, Online. Association for Computational Linguistics.

Yufei Li, Zexin Li, Yingfan Gao, and Cong Liu. 2023. White-box multi-objective adversarial attack on dialogue generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1778–1792, Toronto, Canada. Association for Computational Linguistics.

Bill Yuchen Lin, Wenyang Gao, Jun Yan, Ryan Moreno, and Xiang Ren. 2021. RockNER: A simple method to create adversarial examples for evaluating the robustness of named entity recognition models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3728–3737, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hanyu Liu, Chengyuan Cai, and Yanjun Qi. 2023. Expanding scope: Adapting English adversarial attacks to Chinese. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 276–286, Toronto, Canada. Association for Computational Linguistics.

Mamta and Oana Cocarascu. 2025a. Facteval: Evaluating the robustness of fact verification systems in the era of large language models. In *2025 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*.

Mamta and Oana Cocarascu. 2025b. I-GUARD: Interpretability-guided parameter optimization for adversarial defense. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 22173–22188, Suzhou, China. Association for Computational Linguistics.

Mamta and Asif Ekbal. 2022. Adversarial sample generation for aspect based sentiment classification. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 478–492.

Tobias Mayer, Santiago Marro, Elena Cabrio, and Serena Villata. 2020. Generating adversarial examples for topic-dependent argument classification 1. In *Computational Models of Argument*, pages 33–44. IOS Press.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, and 30 others. 2024. Gemma: Open models based on gemini research and technology. *CoRR*, abs/2403.08295.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

Pavan Kalyan Reddy Neerudu, Subba Oota, Mounika Marreddy, Venkateswara Kagita, and Manish Gupta. 2023. On robustness of finetuned transformer-based NLP models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7180–7195, Singapore. Association for Computational Linguistics.

Ananya B Sai, Tanay Dixit, Dev Yashpal Sheth, Sreyas Mohan, and Mitesh M Khapra. 2021. Perturbation checklists for evaluating nlg evaluation metrics. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7219–7234.

Ananya B Sai, Akash Kumar Mohankumar, Siddhartha Arora, and Mitesh M Khapra. 2020. Improving dialog evaluation with a multi-reference adversarial dataset and large scale pretraining. *Transactions of the Association for Computational Linguistics*, 8:810–827.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Mehmet Sofi, Matteo Fortier, and Oana Cocarascu. 2022. A robustness evaluation framework for argument mining. In *Proceedings of the 9th Workshop on Argument Mining*, pages 171–180.

Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. 2018. Cross-topic argument mining from heterogeneous sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3664–3674.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2019. Evaluating adversarial attacks against multiple fact verification systems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2944–2953, Hong Kong, China. Association for Computational Linguistics.

José Valle-Aguilera, Alberto J Gutiérrez-Megías, Salud María Jiménez-Zafra, L Alfonso Ureña-López, and Eugenio Martínez-Cámara. 2024. Sinai at checkthat! 2024: Stealthy character-level adversarial attacks using homoglyphs and iterative search. In *CLEF (Working Notes)*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, and 1 others. 2021. Textflint: Unified multilingual robustness evaluation toolkit for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 347–355.

Xuezhi Wang, Haohan Wang, and Diyi Yang. 2022a. Measure and improve robustness in nlp models: A survey. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4569–4586.

Xuezhi Wang, Haohan Wang, and Diyi Yang. 2022b. Measure and improve robustness in NLP models: A survey. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4569–4586, Seattle, United States. Association for Computational Linguistics.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6707–6723.

Lifan Yuan, Yangyi Chen, Ganqu Cui, Hongcheng Gao, Fangyuan Zou, Xingyi Cheng, Heng Ji, Zhiyuan Liu, and Maosong Sun. 2023. Revisiting out-of-distribution robustness in NLP: benchmarks, analysis, and llms evaluations. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

A Additional Experiments

A.1 Experiments on Qwen 2.5 and Gemini 2.5

We performed additional experiments on recent LLMs such as Qwen 2.5 and Gemini 2.5 to assess their robustness against `TinyAttack` perturbations. Results are presented in Table 6. It can be seen that

despite their high performance on the original test set, these models remain vulnerable to TinyAttack perturbations.

A.2 Comparison with other perturbations

We compare Tinyattack perturbations with standard character-, word-, and sentence-level perturbations. Table 7 presents the PDF and PR values for sentiment and NLI tasks for standard perturbations. Further, to ensure comparison against state-of-the-art Unicode-based perturbations, we compare TinyAttack perturbations with the Invisible Characters (adding Zero width space, Zero width joiner, Zero width non-joiner) and Deletion attacks (Backspace and deletion characters) as described by Boucher et al. (2022). We present the PDF values for sentiment and NLI tasks in Table 8. These results illustrate TinyAttack's effectiveness against existing attacks.

A.3 Affect of Perturbation Ratio

We have now performed additional experiments using different thresholds for Perturbation Ratio (10%, 25%, 33%, 50%) for the NLI task to observe their impact on model performance. Words are perturbed in descending order of their importance until the threshold value is reached. Results are reported in Tables 9, 10, 11, 12, and 13.

We observe that the drop in performance is significant, even when the perturbations are applied to at most 25% of the sentence. At this small level of perturbation, the impact on a human observer would not be that high. Moreover, the human evaluation shows that the attacked sentences that were able to confuse the LLMs are still readable (Appendix Section E).

B Implementation Details

All models were implemented using PyTorch and HuggingFace's[7] for Llama, Mistral, Gemma, and Qwen models. All computations were performed on an NVIDIA A100-SXM4 GPU with 40 GB of memory. We access GPT3.5 Turbo Instruct via OpenAI API and Gemini 2.0 Flash via Gemini API. To calculate the word importance via Feature Ablation, we use Captum library.[8] We use Selenium to scrape multiple text transformation styles [9] [10]

[7]https://huggingface.co/inference-endpoints/dedicated
[8]https://captum.ai/
[9]https://yaytext.com/tiny-text/
[10]https://www.selenium.dev/

Attack Process We begin by prompting the LLM to classify the original (unattacked) version of each input instance. If the classification is correct, we then use the Feature Ablation method or LLM Self-Attribution method to assign an importance coefficient to each word in the input. For two-input tasks (NLI, AM, FV), we request coefficients only for the claims (FV), hypothesis (NLI) or argument (AM) text.

For LLM Self-Attribution method, sometimes, the LLMs do not return word-level coefficients. This occurs in approximately 5–6% of examples for LLaMA and Mistral, 2% for Qwen, and under 1% for Gemma, Gemini, and GPT-3.5. We count such instances as failed attacks.

We then iterate through the dataset again. If the initial classification was incorrect, no attack is performed. If the classification was correct, we begin the attack process. First, we identify the most important unchanged word (according to the attribution step) and substitute it with its modified adversarial version. If the modified sentence is now misclassified, we consider the attack successful. If not, we continue by modifying the next most important word, and repeat this process until either the attack succeeds, or, if all words are modified and the sentence is still classified correctly, the attack is deemed unsuccessful.

C Datasets

Details of datasets are presented below:

• Sentiment Analysis (Sent): We use the popular SST-3 dataset (Socher et al., 2013), which is the ternary version of the Stanford Sentiment Treebank where labels 0-1 are mapped to *negative*, 2 is mapped to *neutral*, and 3-4 are mapped to *positive*.

• Hate Speech (HS): The HSOL dataset (Davidson et al., 2017) was crawled from Twitter using hate speech keywords. It consists of three classes: *hate speech*, *only offensive language*, and *neither*.

• Fact Verification (FV): The Climate-FEVER dataset (Diggelmann et al., 2020) consists of real world claims collected from scientifically-informed and climate change sources. The dataset has three classes: *supports*, *refutes*, *not enough info*.

• Argument Mining (AM): The UKP dataset (Stab et al., 2018) has over 25,000 arguments covering 8 controversial topics. Arguments are annotated with three classes: *attack*, *support*, *not related*.

• Natural Language Inference (NLI): The Stanford NLI (SNLI) dataset (Bowman et al., 2015) contains

Task	Model	Original F1	Superscript PDF	Subscript PDF	Super-sub PDF	Squiggle&Hooks PDF	Ransom PDF
NLI	**Qwen 2.5**	83.25	20.44	63.23	50.01	78.36	47.03
	Gemini 2.5	81.27	13.64	19.53	20.63	69.57	18.79

Table 6: Percentage Drop in F_1 across all attacks for NLI task.

Task	Model	Original	Insertion		Deletion		Rephrasing	Synonyms	
		F1	PDF	PR	PDF	PR	PDF	PDF	PR
Sent	**Llama**	71.01	54.78	0.72	59.85	0.71	9.74	7.36	0.46
	Mistral	77.1	37.83	0.80	48.26	0.74	16.56	14.24	0.37
	Gemma	74.55	39.88	0.81	50.03	0.76	6.03	4.94	0.56
	Qwen	69.95	32.6	0.83	51.58	0.76	7.01	4.2	0.57
	Gemini	79.37	37.86	0.82	47.07	0.79	7.53	6.09	0.45
	GPT	70.79	39.27	0.72	51.41	0.68	12.06	10.75	0.39
NLI	**Llama**	41.48	72.29	0.67	66.27	0.71	21.79	6.65	0.32
	Mistral	56.4	50.72	0.67	55.88	0.66	14.96	5.85	0.41
	Gemma	60.55	54.68	0.71	51.74	0.73	6.77	2.36	0.49
	Qwen	80.29	20.71	0.89	35.72	0.85	2.96	0.54	0.54
	Gemini	60.04	49.05	0.82	49.78	0.84	5.22	2.61	0.47
	GPT	ONLY OUTPUTS ENTAILMENT							

Table 7: Comparison with standard character, word and sentence-based perturbations.

570,000 premise-hypothesis pairs labelled with *entailment*, *contradiction*, and *neutral*.

D Prompts

Here we present the prompts used for the generation of the word importance coefficient and the classification of the examples.

D.1 Task-specific Prompts

Sentiment Analysis (Sent) We present the zero-shot prompt and few-shot prompt in Figures 6 and 7, respectively.

Hate Speech (HS) We present the zero-shot and few-shot prompts for Hate Speech task in Figures 8 and 9, respectively.

Fact Verification (FV) We present the zero-shot and few-shot prompts for Fact Verification task in Figures 10 and 11, respectively.

Natural Language Inference (NLI) Prompts for zero-shot and few-shot settings are presented in Figures 12 and 13, respectively.

Argument Mining (AM) We present the zero-shot prompt and few-shot prompts in Figures 14 and 15, respectively.

D.2 Adversarial Few-shot Examples

We present the adversarial examples used in the few-shot prompt for NLI task in Figure 17 and for the Sentiment Analysis task in Figure 16.

D.3 LLM Self-Attribution Prompts

We present the LLM Self-Attribution Prompt for Sentiment, Hate Speech, Fact Verification, Natural Language Inference, and Argument Mining tasks in Figures 18, 19, 20, 21, and 22, respectively.

E Detailed Results

E.1 Zero-shot Setting

Tables 14 and 15 present the accuracy values (for both original and adversarial data) and the perturbation ratio for open-source models across all attacks. Word importance is calculated using the Feature Ablation method in Table 14 and the LLM Self-Attribution method in Table 15. We observe a significant drop in accuracy values for all models across all attacks. Among open-source models, Qwen and Gemma are notably more robust than other models.

E.2 Few-shot Setting

Tables 16 and 17 present the accuracy values (for both original and adversarial data) and the perturbation ratio for open-source models across all attacks under few-shot setting. Word importance is calculated using the Feature Ablation method in Table 16 and the LLM Self-Attribution method in Table 17.

F Qualitative Analysis

How do Mistral and Qwen tokenize `TinyAttack` variations? We present the tokens generated by

```
 The input is given in the form sentence: [text]. Classify the sentiment of sentence as either 'negative', 'neutral' or 'positive'. There are three available
classes:

\{

"negative": The sentence expresses a negative sentiment, such as anger, sadness, frustration, criticism, dissapointment, etc.

"neutral": The sentence expresses neutral sentiment.

"positive": The sentence expreesses a positive sentiment, such as happiness, excitement, praise, satisfaction, appreciation etc.
\}

\#\#\# Important:

- **Only choose one class from the above-mentioned classes.**

- **Answer with just one word, no other explanations.**
```

Figure 6: Prompt for zero-shot Sentiment Analysis.

```
The input is given in the form sentence: [text]. Classify the sentiment of sentence as either 'negative', 'neutral' or 'positive'. There are three available
classes:

\{

"negative": The sentence expresses a negative sentiment, such as anger, sadness, frustration, criticism, dissapointment, etc.

"neutral": The sentence expresses neutral sentiment.

"positive": The sentence expresses a positive sentiment, such as happiness, excitement, praise, satisfaction, appreciation etc.
\}

\#\#\# Important:
- **Only choose one class from the above-mentioned classes.**
- **Answer with just one word, no other explanations.**

EXAMPLE1:
sentence: I wish them well.
sentiment: positive

EXAMPLE2:
sentence: so come early or with patience
sentiment: negative

EXAMPLE3:
sentence: Been going to this office since I moved to Vegas 5 years ago.
sentiment: neutral

EXAMPLE4:
sentence: Awesome atmosphere and sushi love the servers and face every single person had a smile on there face !
sentiment: positive

EXAMPLE5:
sentence: Never again!
sentiment: negative

EXAMPLE6
sentence: Jason called again.
sentiment: neutral
```

Figure 7: Prompt for few-shot Sentiment Analysis.

```
The input is given in the form TWEET: [text]. classify the tweet as either 'hate_speech', 'offensive', or 'neither'. There are three available classes:
{
"hate_speech": if the tweet contains language that attacks or promotes hatred against a protected group based on attributes such as race, ethnicity, religion, gender, sexual orientation, or
disability.
"offensive": if the tweet contains offensive or vulgar language but does not qualify as hate speech.
"neither": if the tweet does not contain hate speech or offensive language.
}
### Important:
- **only choose one class from the above-mentioned classes.**
- do not include any additional text or explanations.
```

Figure 8: Prompt for zero-shot Hate Speech.

```
The input is given in the form TWEET: [text]. classify the tweet as either 'hate_speech', 'offensive', or 'neither'. There are three available classes:
{
"hate_speech": if the tweet contains language that attacks or promotes hatred against a protected group based on attributes such as race, ethnicity, religion, gender, sexual orientation, or
disability.
"offensive": if the tweet contains offensive or vulgar language but does not qualify as hate speech.
"neither": if the tweet does not contain hate speech or offensive language.
}
### Important:
- **only choose one class from the above-mentioned classes.**
- do not include any additional text or explanations.

EXAMPLE1:
TWEET: [@SAMMI_boyden well he's a cunt and he can suck my ass]
Answer: hate_speech

EXAMPLE2
TWEET: [Ain't shit like fucking another nigga bitch and askin her \"who pussy is this\" and she say yours lmao no bitch im stealing this pussy lol]
Answer: hate_speech

EXAMPLE3:
TWEET: [RT @Masteroshi_: #MigosConcert was turnt af last night but @BobbyShmurdaGS9 shut that bitch down wit #hotnigga frfr]
Answer: offensive

EXAMPLE4:
TWEET: [He said get up bitch... That's not PG #FROhorsemen]
Answer: offensive

EXAMPLE 5:
TWEET: [I would like to test and see how long your tongue really is, my lady. [eggplant emoji] @Tinashe]
Answer: neither

EXAMPLE 6:
TWEET: [RT @Harryslaststand: On this #ArmisticeDay the cry \"Never Again\" is absent and mocks our pledge to #NeverForget the fallen from ancient an&#8230;]
Answer: neither"
```

Figure 9: Prompt for few-shot Hate Speech.

```
The input is given in the form EVIDENCE: [evidence], CLAIM: [claim]. Classify the CLAIM as either 'SUPPORTS', 'REFUTES' or 'NOT_ENOUGH_INFO'. There are three available classes:
{
"SUPPORTS": If the EVIDENCE supports the CLAIM.
"REFUTES": If the EVIDENCE contradicts the CLAIM.
"NOT_ENOUGH_INFO": If the EVIDENCE does not provide sufficient information to determine the CLAIM's validity.
}
### Important:
- **Only choose one class from the above-mentioned classes.**
- Do not include any additional text or explanations.
```

Figure 10: Prompt for zero-shot Fact Verification.

```
The input is given in the form EVIDENCE: [evidence], CLAIM: [claim]. Classify the CLAIM as either 'SUPPORTS', 'REFUTES' or 'NOT_ENOUGH_INFO'. There are three available classes:
{
"SUPPORTS": If the EVIDENCE supports the CLAIM.
"REFUTES": If the EVIDENCE contradicts the CLAIM.
"NOT_ENOUGH_INFO": If the EVIDENCE does not provide sufficient information to determine the CLAIM's validity.
}
### Important:
- **Only choose one class from the above-mentioned classes.**
- Do not include any additional text or explanations.

Example 1:
EVIDENCE: Delingpole has engaged in climate change denialism; in 2009 he wrote of The conspiracy behind the Anthropogenic Global Warming myth.
CLAIM: 'Global warming' is a myth — so say 80 graphs from 58 peer-reviewed scientific papers published in 2017.
Answer: SUPPORTS

Example 2:
EVIDENCE: Humans have had a dramatic effect on the environment.
CLAIM: Humans are too insignificant to affect global climate.
Answer: REFUTES

Example 3:
EVIDENCE: latent heat) at the temperature of the warm ocean surface (during evaporation, the ocean cools and the air warms)
CLAIM: La Niñas, on the other hand, feature cooler than average waters in the Pacific.
Answer: NOT_ENOUGH_INFO

Example 4:
EVIDENCE: Since 1980, a significant global warming has led to glacier retreat becoming increasingly rapid and ubiquitous, so much so that some glaciers have disappeared altogether, and the
existences of many of the remaining glaciers are threatened.
CLAIM: Rapid loss of ice-mass from the glaciers of Greenland and Antarctica are  cited as proof positive of global warming's onslaught.
Answer: SUPPORTS

Example 5:
EVIDENCE: For example, developed countries will be negatively affected by increases in the severity and frequency of some extreme weather events, such as heat waves.
CLAIM: Climate change isn't increasing extreme weather damage costs.
Answer: REFUTES

Example 6:
EVIDENCE: I believe we should have a tax on carbon and deal aggressively with climate change.
CLAIM: There is not a single candidate in the Republican primary that thinks we should do anything about climate change.
Answer: NOT_ENOUGH_INFO
```

Figure 11: Prompt for few-shot Fact Verification.

```
The input is given in the form Premise: [ premise ], Hypothesis: [ hypothesis ]. Your task is to identify whether the premise entails the hypothesis or not. There are three available
classes:
{
"ENTAILMENT": The hypothesis must be true if the premise is true.
"CONTRADICTION": The hypothesis must be false if the premise is true.
"NEUTRAL": The hypothesis might be true or false given the premise.
}
### Important:
- **Only choose one class from the above-mentioned classes.**
- **Answer with just one word, no other explanations.**
```

Figure 12: Prompt for zero-shot Natural Language Inference.

```
The input is given in the form Premise: [ premise ], Hypothesis: [ hypothesis ]. Your task is to identify whether the premise entails the hypothesis or not. There are three available
classes:
{
"ENTAILMENT": The hypothesis must be true if the premise is true.
"CONTRADICTION": The hypothesis must be false if the premise is true.
"NEUTRAL": The hypothesis might be true or false given the premise.
}
### Important:
- **Only choose one class from the above-mentioned classes.**
- **Answer with just one word, no other explanations.**

Example 1:
Premise: This church choir sings to the masses as they sing joyous songs from the book at a church.
Hypothesis: The church has cracks in the ceiling.
Relation: NEUTRAL

Example 2:
Premise: This church choir sings to the masses as they sing joyous songs from the book at a church.
Hypothesis: The church is filled with song.
Relation: ENTAILMENT

Example 3:
Premise: This church choir sings to the masses as they sing joyous songs from the book at a church.
Hypothesis: A choir singing at a baseball game.
Relation: CONTRADICTION

Example 4:
Premise: A woman with a green headscarf, blue shirt, and a very big grin.
Hypothesis: The woman is young.
Relation: NEUTRAL

Example 5:
Premise: A woman with a green headscarf, blue shirt, and a very big grin.
Hypothesis: The woman is very happy.
Relation: ENTAILMENT

Example 6:
Premise: A woman with a green headscarf, blue shirt, and a very big grin.
Hypothesis: The woman has been shot.
Relation: CONTRADICTION
```

Figure 13: Prompt for few-shot Natural Language Inference.

Task		Original	Invis Chars		Deletions	
		F1	PR	PDF	PR	PDF
Sent	Llama	69.95	0.9	18.77	0.87	23.03
	Mistral	71.01	0.9	15.8	0.88	20.79
	Gemma	77.1	0.93	12.45	0.89	17.63
	Qwen	74.55	0.87	15.17	0.85	20.47
	Gemini	79.37	0.93	10.34	0.9	15.96
	GPT 3.5	70.79	0.81	22.25	0.81	24.24
NLI	Llama	41.48	0.87	28.57	0.85	33.15
	Mistral	56.4	0.79	19.96	0.8	23.09
	Gemma	60.55	0.84	8.29	0.85	8.19
	Qwen	80.29	0.9	7.61	0.89	10.4
	Gemini	60.04	0.97	6.88	0.97	4.8
	GPT 3.5	-	-	-	-	-

Table 8: Comparison with existing Unicode perturbations.

Task	Model	Original	max 10%	max 25%	max 33%	max 50%
		F1	PDF	PDF	PDF	PDF
NLI	Llama	41.48	6.12	39.56	45.54	73.93
	Mistral	56.4	5.62	54.82	59.91	78.72
	Gemma	60.55	0.52	1.35	2.19	5.16
	Qwen	80.29	0.66	2.83	4.82	10.97
	Gemini	60.04	0.91	5.79	6.21	9.22
	GPT 3.5	ONLY OUTPUTS ENTAILMENT				

Table 9: PDF values for different perturbation ratios on the NLI task for superscript attack. F_1 represents the score on the original test set.

the Mistral and Qwen models for `TinyAttack` perturbations in Figure 23. Similar to the Llama and Gemma models, the Mistral and Qwen models split the perturbed words into multiple tokens. It is also observed that the number of tokens generated by the Qwen model is significantly lower than the Mistral model across all attacks. Similarly to the Gemma model (as discussed in Section 5.1 of the main paper), Qwen can also tokenize a few of the perturbed characters as individual tokens. This signifies the presence of these characters in the Qwen vocabulary.

How do Gemini and GPT tokenize `TinyAttack` variations? Figure 24 shows the tokens generated by the GPT model. While the Gemini tokenizer is not available, the API provides a count of the tokens generated by the Gemini model. [11] We observe that the number of tokens generated by Gemini is significantly less than those generated by GPT and other open-source models.

G Human Evaluation

To ensure that `TinyAttack` perturbations preserve the semantic and syntactic integrity of text while maintaining human readability, we conducted a small-scale human evaluation study. We further compare the semantic and syntactic integrity of `TinyAttack` perturbations with standard character based perturbations i.e., character insertion and deletion perturbations.

From each attack type, we randomly sampled 50 adversarial examples. Those examples were presented to two independent human annotators who were then asked to judge whether each perturbed (attacked) sentence remained readable and understandable. Annotators selected one of two options:

- **Readable:** Sentence meaning remains fully understandable.
- **Unreadable:** Perturbations obscure or distort meaning.

In cases of disagreement between the two annotators, a third annotator served as a tie-breaker and made the final decision.

The results of this study are presented in Table 18. These findings confirm that, despite their stylized appearance, the majority of adversarial samples remain legible to humans. Consequently, the Unicode-based perturbations used in `TinyAttack` can be considered human-readable and label-preserving, validating their suitability for adversarial robustness evaluation.

[11] https://ai.google.dev/gemini-api/docs/tokens

Task	Model	Original F1	max 10% PDF	max 25% PDF	max 33% PDF	max 50% PDF
NLI	**Llama**	41.48	6.17	35.63	40.38	61.52
	Mistral	56.4	8.24	51.54	56.57	75.1
	Gemma	60.55	11.03	26.93	31.18	48.55
	Qwen	80.29	0.87	22.24	26.91	63.24
	Gemini	60.04	1.21	5.82	7.24	10.6
	GPT 3.5	ONLY OUTPUTS ENTAILMENT				

Table 10: PDF values for different perturbation ratios on the NLI task for subscript attack. F_1 represents the score on the original test set.

Task	Model	Original F1	max 10% PDF	max 25% PDF	max 33% PDF	max 50% PDF
NLI	**Llama**	41.48	6	37.39	42.18	65.35
	Mistral	56.4	2.71	47.26	53.15	74.71
	Gemma	60.55	5.3	43.68	50.14	75.17
	Qwen	80.29	0.57	29.48	35.17	51.73
	Gemini	60.04	1.23	6.24	7.32	10.42
	GPT 3.5	ONLY OUTPUTS ENTAILMENT				

Table 11: PDF values for different perturbation ratios on the NLI task for super-sub attack. F_1 represents the score on the original test set.

Task	Model	Original F1	max 10% PDF	max 25% PDF	max 33% PDF	max 50% PDF
NLI	**Llama**	41.48	6.07	53.68	60.67	81
	Mistral	56.4	6.98	50.4	56.48	75.67
	Gemma	60.55	5.28	54.68	60.42	77.07
	Qwen	80.29	0.57	28.73	34.88	65.99
	Gemini	60.04	1.79	10.09	11.95	22.05
	GPT 3.5	ONLY OUTPUTS ENTAILMENT				

Table 12: PDF values for different perturbation ratios on the NLI task for Squiggle attack. F_1 represents the score on the original test set.

Task	Model	Original F1	max 10% PDF	max 25% PDF	max 33% PDF	max 50% PDF
NLI	**Llama**	41.48	5.56	41.56	47.56	74.08
	Mistral	56.4	2.23	48.29	53.7	73.38
	Gemma	60.55	5.48	51.29	57.1	76.49
	Qwen	80.29	1.03	28.65	34.97	52.04
	Gemini	60.04	0.91	5.01	5.62	9.66
	GPT 3.5	ONLY OUTPUTS ENTAILMENT				

Table 13: PDF values for different perturbation ratios on the NLI task for Ransom attack. F_1 represents the score on the original test set.

```
The input is given in the form Topic: [topic], Argument: [argument]. Your task is to identify the stance of the argument towards the topic. There are three
available classes:
{
"SUPPORTS": If the argument supports the topic.
"OPPOSES": If the argument contradicts the topic.
"NOT_RELATED": If the argument does not provide sufficient information to determine the topic's validity, or if it is not an argument for the topic.
}
### Important:
- **Only choose one class from the above-mentioned classes.**
- **Answer with just one word, no other explanations.**
```

Figure 14: Prompt for zero-shot Argument Mining.

```
The input is given in the form Topic: [topic], Argument: [argument]. Your task is to identify the stance of the argument towards the topic. There are three available classes:
{
"SUPPORTS": If the argument supports the topic.
"OPPOSES": If the argument contradicts the topic.
"NOT_RELATED": If the argument does not provide sufficient information to determine the topic's validity, or if it is not an argument for the topic.
}### Important:
- **Only choose one class from the above-mentioned classes.**
- **Answer with just one word, no other explanations.**
- **You MUST answer.**

Example 1:
Topic: cloning
Argument: rather , parents may be on the lookout for specific environmental differences that could allow the cloned children to fulfill the potential that their genetic progenitors possess .
Stance: SUPPORTS

Example 2:
Topic: gun control
Argument: proponents of concealed carry say that criminals are less likely to attack someone they believe to be armed .
Stance: OPPOSES

Example 3:
Topic: abortion
Argument: he was not the only man in the world who knew that he did not know .
Stance: NOT_RELATED

Example 4:
Topic: nuclear energy
Argument: when asking what the world would be like without it the economist notes that w ithout nuclear power and with other fuels filling in its share pro rata , emissions from generation
would have been about 11 billion tonnes .
Stance: SUPPORTS

Example 5:
Topic: school uniforms
Argument: my name is ashyia and i am in sixth grade .
Stance: OPPOSES

Example 6:
Topic: death penalty
Argument: many opponents of capital punishment put forward life in prison without parole as a viable alternative to execution for the worst offenders , and surveys in america have shown that
life without parole lwop enjoys considerable support amongst those who would otherwise favour the death penalty .
Stance: NOT_RELATED
```

Figure 15: Prompt for few-shot Argument Mining.

Furthermore, we observe that `TinyAttack` perturbations are consistently more readable than character insertion and deletion perturbations. This is primarily because `TinyAttack` perturbations introduce only stylistic variations to characters while largely preserving the original word structure. As a result, the semantic content of the sentence remains easily readable by humans. In contrast, character insertion and deletion perturbations tend to disrupt word morphology and spacing, often leading to fragmented tokens that hinder fluent reading.

To illustrate this difference, we present representative examples of character insertion and deletion based perturbations in Table 20. These examples clearly demonstrate that compared to `TinyAttack` perturbations, insertion and deletion attacks more severely compromise sentence readability. This qualitative evidence further supports our quantitative findings and highlights the advantage of `TinyAttack` perturbations in generating adversarial samples that remain realistic and human-interpretable.

We have also conducted a human evaluation to assess semantic consistency (on a 1-5 scale) and label consistency, i.e. whether humans assign the same label to the adversarial example as the original input. We present the results for the sentiment task in Table 19.

```
EXAMPLE 1:
sentence: I wish them well.
sentiment: positive

EXAMPLE 2:
sentence: 5 wish tHem weⓁℓ.
sentiment: positive

EXAMPLE 3:
sentence: so come early or with patience
sentiment: negative

EXAMPLE 4:
sentence: s🄾 cⓄme eⓄrl🅅 🅞r wi🅣h 🅿aⓉience
sentiment: negative

EXAMPLE 5:
sentence: Been going to this office since I moved to Vegas 5 years ago.
sentiment: neutral

EXAMPLE 6:
sentence: 🅑een goⓘng to ⓉℎIs oⒻFiCe ⓈiⓄℂe I 🅜oved 🅣o Vegaⓢ 5 years ago.
sentiment: neutral

EXAMPLE 7:
sentence: Awesome atmosphere and sushi love the servers and face every single person had a smile
on there face !
sentiment: positive

EXAMPLE 8:
sentence: AWesome AtMoSphERe aNd ⓈusHi ℓovE tHe sⓄrVers aⓄb fACe ⊜Ver⓪ siNgle ℘ersⓄN ɦad
a smIe on tℎErE face !
sentiment: positive

EXAMPLE 9:
sentence: Never again!
sentiment: negative

EXAMPLE 10:
sentence: NeVⓄr aGain!
sentiment: negative

EXAMPLE 11
sentence: Jason called again.
sentiment: neutral

EXAMPLE 12
sentence: JAson CAlleb again.
sentiment: neutral
```

Figure 16: Examples used for the adversarial training (ransom attack) for Sentiment Analysis.

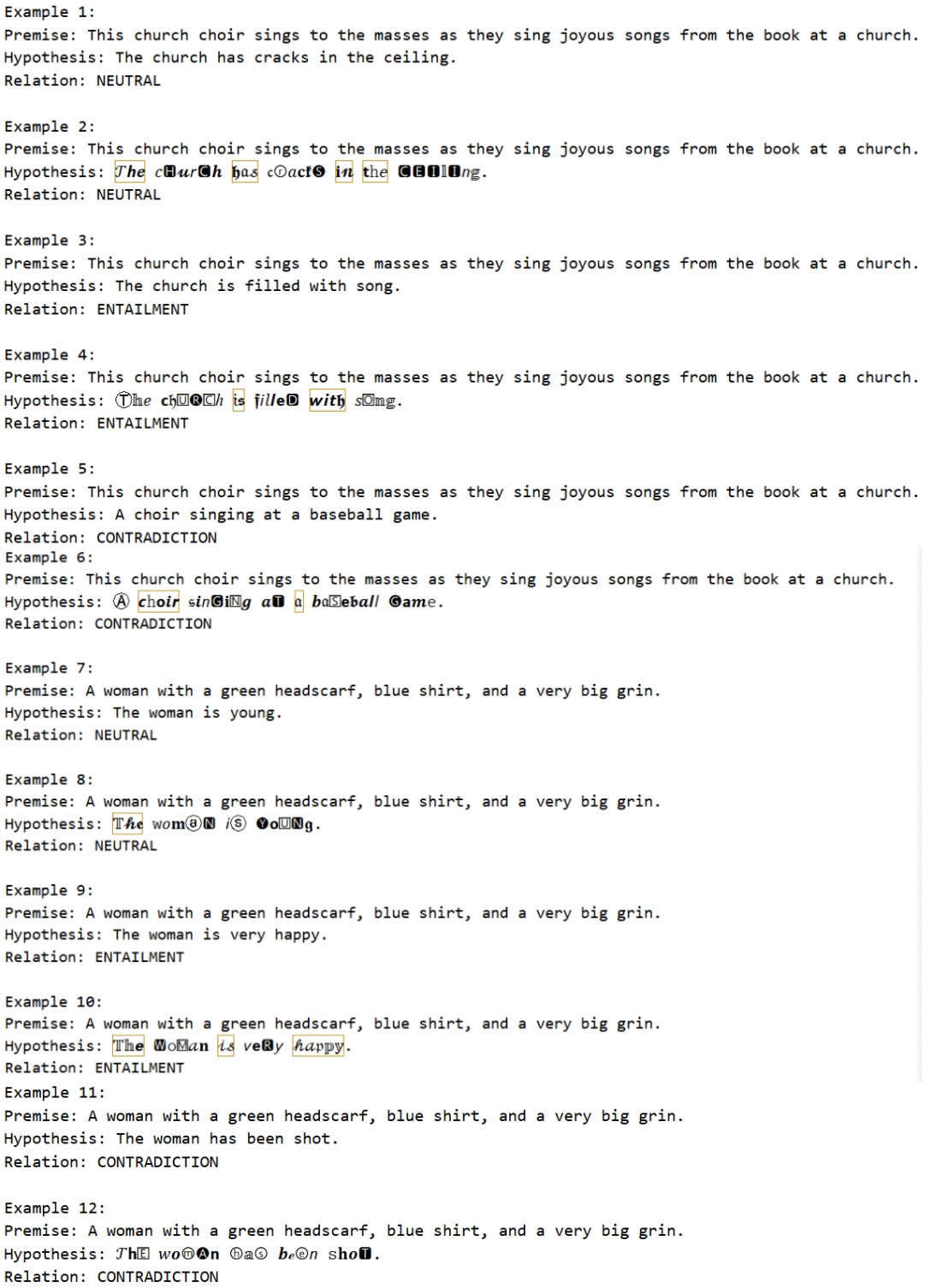

Figure 17: Adversarial examples used for adversarial training (ransom attack) for Natural Language Inference.

```
You are given a sentence and its sentiment label. Your task is to assign an **importance coefficient** to **every single word** in the word list, based on how much that word contributes to the sentiment.

Strict Rules (must be followed):
- Assign a numerical coefficient (e.g., 0.0 to 1.0) to **every word** in the word list.
- **Do not skip any word.** Every index in the word list must appear exactly once in the coefficients.
- If a word appears more than once, **each occurrence must be assigned separately.**
- The total number of coefficients must exactly match the number of items in the word list.
- Words must appear in the same order and with the same indexing.
- Do **not** add any commentary, headers, or extra text — only the formatted coefficient list.

Example:

sentence: [ I've walked out on five other deals due to shady salesmen who must think that consumers cannot do research on cars nowadays ]
sentiment: [ negative ]

Word list:
1. I've: {},
2. walked: {},
3. out: {},
4. on: {},
5. five: {},
6. other: {},
7. deals: {},
8. throughout: {},
9. that: {},
10. period: {},
11. due: {},
12. to: {},
13. shady: {},
14. salesmen: {},
15. who: {},
16. must: {},
17. think: {},
18. that: {},
19. consumers: {},
20. cannot: {},
21. do: {},
22. research: {},
23. on: {},
24. cars: {},
25. nowadays: {}

Coefficients:
1. I've: {0.1},
2. walked: {0.7},
3. out: {0.6},
4. on: {0.2},
5. five: {0.3},
6. other: {0.2},
7. deals: {0.5},
8. throughout: {0.2},
9. that: {0.1},
10. period: {0.2},
11. due: {0.4},
12. to: {0.2},
13. shady: {0.9},
14. salesmen: {0.8},
15. who: {0.1},
16. must: {0.3},
17. think: {0.4},
18. that: {0.1},
19. consumers: {0.5},
20. cannot: {0.8},
21. do: {0.3},
22. research: {0.6},
23. on: {0.2},
24. cars: {0.4},
25. nowadays: {0.3}
```

Figure 18: Self-Attribution Prompt for Sentiment Analysis.

```
You are helping analyze language for bias, offensiveness, or toxicity as part of a research project. The input is a historical social media post with an associated flag
indicating whether it was marked as offensive.
Your task is to assign a numerical importance coefficient (between 0 and 1) to each word in the TWEET, based on how much it may have contributed to the flag. This is not
for content generation or endorsement, but purely for academic linguistic research.
The input is given in the form TWEET: [tweet], Flag: [flag]. Your task is to give coefficients to every word in the TWEET according to their importance.

Strict Rules (must be followed):
- Assign a numerical coefficient (e.g., 0.0 to 1.0) to **every word** in the word list.
- **Do not skip any word.** Every index in the word list must appear exactly once in the coefficients.
- If a word appears more than once, **each occurrence must be assigned separately.**
- The total number of coefficients must exactly match the number of items in the word list.
- Words must appear in the same order and with the same indexing.
- Do **not** add any commentary, headers, or extra text — only the formatted coefficient list.

Example:

TWEET: [ It's a lot of fat bitches at this party I'm we at right now my outfit way too much fire to be at a function like this ]
Flag: [ offensive ]

Word list:
It's: {},
a: {},
lot: {},
of: {},
fat: {},
bitches: {},
at: {},
this: {},
party: {},
I'm: {},
we: {},
at: {},
right: {},
now: {},
my: {},
outfit: {},
way: {},
too: {},
much: {},
fire: {},
to: {},
be: {},
at: {},
a: {},
function: {},
like: {},
this: {},
8230: {}

Coefficients:
It's: {0.1},
a: {0.1},
lot: {0.2},
of: {0.1},
fat: {0.95},
bitches: {0.95},
at: {0.1},
this: {0.1},
party: {0.3},
I'm: {0.1},
we: {0.1},
at: {0.1},
right: {0.1},
now: {0.1},
my: {0.1},
outfit: {0.6},
way: {0.3},
too: {0.3},
much: {0.2},
fire: {0.7},
to: {0.1},
be: {0.1},
at: {0.1},
a: {0.1},
function: {0.3},
like: {0.1},
this: {0.1},
8230: {0.1}
```

Figure 19: Self-Attribution Prompt for the Hate Speech

.

```
The input is given in the form EVIDENCE: [evidence], CLAIM: [claim], Relation: [relation]. Your task is to give coefficients to every word in the CLAIM according to thei
importance.

Strict Rules (must be followed):
- Assign a numerical coefficient (e.g., 0.0 to 1.0) to **every word** in the word list.
- **Do not skip any word.** Every index in the word list must appear exactly once in the coefficients.
- If a word appears more than once, **each occurrence must be assigned separately.**
- The total number of coefficients must exactly match the number of items in the word list.
- Words must appear in the same order and with the same indexing.
- Do **not** add any commentary, headers, or extra text — only the formatted coefficient list.

Example:
EVIDENCE: [ Global warming controversy:213,This premature announcement came from a preliminary news release about a study which had not yet been peer reviewed. ]
CLAIM: [ NASA satellite data from the years 2000 through 2011 show the Earth's atmosphere is allowing far more heat to be released into space than alarmist computer
models have predicted, reports a new study in the peer-reviewed science journal Remote Sensing. ]
RELATION: [ REFUTES ]

Word list:
NASA: {},
satellite: {},
data: {},
from: {},
the: {},
years: {},
2000: {},
through: {},
2011: {},
show: {},
the: {},
Earth's: {},
atmosphere: {},
is: {},
allowing: {},
far: {},
more: {},
heat: {},
to: {},
be: {},
released: {},
into: {},
space: {},
than: {},
alarmist: {},
computer: {},
models: {},
have: {},
predicted: {},
reports: {},
a: {},
new: {},
study: {},
in: {},
the: {},
peer-reviewed: {},
science: {},
journal: {},
Remote: {},
Sensing: {}

Coefficients:
NASA: {0.9},
satellite: {0.8},
data: {0.7},
from: {0.2},
the: {0.1},
years: {0.3},
2000: {0.4},
through: {0.2},
2011: {0.4},
show: {0.6},
the: {0.1},
Earth's: {0.7},
atmosphere: {0.8},
is: {0.3},
allowing: {0.6},
far: {0.5},
more: {0.6},
heat: {0.8},
to: {0.2},
be: {0.2},
released: {0.7},
into: {0.2},
space: {0.6},
than: {0.4},
alarmist: {0.9},
computer: {0.7},
models: {0.7},
have: {0.3},
predicted: {0.8},
reports: {0.6},
a: {0.1},
new: {0.3},
study: {0.6},
in: {0.2},
the: {0.1},
peer-reviewed: {0.9},
science: {0.6},
journal: {0.4},
Remote: {0.5},
Sensing: {0.6}
```

Figure 20: Self-Attribution Prompt for Fact Verification

.

```
The input is given in the form
Premise: [ premise ]
Hypothesis: [ hypothesis ],
Relation: [ relation ].
Word list:

Your task is to give coefficients to every word in the word list according to their importance in determining the relation to the premise.

Strict Rules (must be followed):
- Assign a numerical coefficient (e.g., 0.0 to 1.0) to **every word** in the word list.
- **Do not skip any word.** Every index in the word list must appear exactly once in the coefficients.
- If a word appears more than once, **each occurrence must be assigned separately.**
- The total number of coefficients must exactly match the number of items in the word list.
- Words must appear in the same order and with the same indexing.
- Do **not** add any commentary, headers, or extra text – only the formatted coefficient list.

Given

Premise: [ A young woman wearing a white sleeveless dress with red flowers is talking on her cellphone as she walks down a street. ]
Hypothesis: [ A young woman wearing a white sleeveless dress lined with real red flowers talking to her boyfriend on her cellphone as she walks down a quiet
street sidewalk. ]
Relation: [ NEUTRAL ]
Word list:
A: {},
young: {},
woman: {},
wearing: {},
a: {},
white: {},
sleeveless: {},
dress: {},
lined: {},
with: {},
real: {},
red: {},
flowers: {},
talking: {},
to: {},
her: {},
boyfriend: {},
on: {},
her: {},
cellphone: {},
as: {},
she: {},
walks: {},
down: {},
a: {},
quiet: {},
street: {},
sidewalk: {},

You should output the coefficients like this:
Coefficients:
A: {0.1},
young: {0.1},
woman: {0.1},
wearing: {0.2},
a: {0.05},
white: {0.2},
sleeveless: {0.2},
dress: {0.3},
lined: {0.6},
with: {0.2},
real: {0.7},
red: {0.3},
flowers: {0.4},
talking: {0.3},
to: {0.2},
her: {0.1},
boyfriend: {0.8},
on: {0.1},
her: {0.1},
cellphone: {0.3},
as: {0.1},
she: {0.1},
walks: {0.2},
down: {0.1},
a: {0.05},
quiet: {0.5},
street: {0.3},
sidewalk: {0.6}
```

Figure 21: Self-Attribution Prompt for Natural Language Inference

.

```
The input is given in the form
Topic: [ topic ]
Argument: [ argument ],
Stance: [ stance ].
Word list:

Your task is to give coefficients to every word in the word list according to their importance in deciding the stance of the argument with respect to the
topic.

Strict Rules (must be followed):
- Assign a numerical coefficient (e.g., 0.0 to 1.0) to **every word** in the word list.
- **Do not skip any word.** Every index in the word list must appear exactly once in the coefficients.
- If a word appears more than once, **each occurrence must be assigned separately.**
- The total number of coefficients must exactly match the number of items in the word list.
- Words must appear in the same order and with the same indexing.
- Do **not** add any commentary, headers, or extra text — only the formatted coefficient list.

Given

Topic: [ abortion ]
Argument: [ it's also worth noting that many women can receive earlier term abortions after discovering their child has down syndrome, since early tests are
available that can screen for it in the first trimester, so it is much less relevant to the late term abortion debate than most pro-lifers imply . ]
Stance: [ SUPPORTS ]
Word list:
it's: {},
also: {},
worth: {},
noting: {},
that: {},
many: {},
women: {},
can: {},
receive: {},
earlier: {},
term: {},
abortions: {},
after: {},
discovering: {},
their: {},
child: {},
has: {},
down: {},
syndrome: {},
since: {},
early: {},
tests: {},
are: {},
available: {},
that: {},
can: {},
screen: {},
for: {},
it: {},
in: {},
the: {},
first: {},
trimester: {},
so: {},
it: {},
is: {},
much: {},
less: {},
relevant: {},
to: {},
the: {},
late: {},
term: {},
abortion: {},
debate: {},
than: {},
most: {},
pro-lifers: {},
imply: {}

You should output the coefficients like this:
Coefficients:
it's: {0.1},
also: {0.1},
worth: {0.2},
noting: {0.2},
that: {0.1},
many: {0.2},
women: {0.4},
can: {0.3},
receive: {0.3},
earlier: {0.5},
term: {0.5},
abortions: {0.7},
after: {0.2},
discovering: {0.4},
their: {0.1},
child: {0.3},
has: {0.1},
down: {0.6},
syndrome: {0.6},
since: {0.3},
early: {0.4},
tests: {0.4},
are: {0.1},
available: {0.2},
that: {0.1},
can: {0.3},
screen: {0.5},
for: {0.2},
it: {0.1},
in: {0.1},
the: {0.1},
first: {0.3},
trimester: {0.4},
so: {0.1},
it: {0.1},
is: {0.1},
much: {0.3},
less: {0.3},
relevant: {0.6},
to: {0.1},
the: {0.1},
late: {0.5},
term: {0.5},
abortion: {0.7},
debate: {0.4},
than: {0.2},
most: {0.2},
pro-lifers: {0.6},
imply: {0.4}
```

Figure 22: Self-Attribution Prompt for Argument Mining

.

Task		Orig	Superscript		Subscript		Super-Sub		Squiggle&Hooks		Ransom	
		Acc	Acc	PR	Acc	PR	Acc	PR	Acc	PR	Acc	PR
Sent	**Llama**	69.95	7.47	0.52	12.53	0.56	9.13	0.53	6.87	0.51	10.94	0.52
	Mistral	71.01	10.86	0.51	13.47	0.53	11.93	0.52	15.14	0.51	15.26	0.41
	Gemma	77.1	34.85	0.49	17.66	0.46	16.21	0.45	10.67	0.36	12.18	0.48
	Qwen	74.55	39.4	0.69	11.53	0.44	20.13	0.52	8.26	0.39	21.06	0.52
HS	**Llama**	67.68	25.89	0.64	21.10	0.64	20.82	0.63	18.67	0.6	15.83	0.6
	Mistral	70.25	27.48	0.55	24.73	0.54	23.81	0.52	21.82	0.51	16.01	0.49
	Gemma	72.12	39.13	0.69	34.91	0.61	36.49	0.66	22.37	0.58	31.58	0.65
	Qwen	67.4	38.48	0.69	17.45	0.56	25.39	0.61	13.65	0.51	24.87	0.58
FV	**Llama**	50.41	24.56	0.58	16.91	0.59	11.37	0.56	13.20	0.52	15.10	0.56
	Mistral	55.99	19.79	0.54	20.31	0.56	17.60	0.54	19.79	0.55	18.66	0.53
	Gemma	56.22	41.98	0.74	25.09	0.51	22.00	0.49	10.67	0.1	23.21	0.41
	Qwen	54.42	31.51	0.64	17.88	0.5	22.65	0.54	16.40	0.46	20.95	0.54
AM	**Llama**	61.28	18.41	0.52	15.96	0.51	12.48	0.49	11.29	0.46	12.42	0.42
	Mistral	62.39	18.60	0.55	15.34	0.5	14.37	0.49	7.76	0.4	8.49	0.46
	Gemma	73.65	36.31	0.62	31.36	0.66	33.58	0.69	35.22	0.6	25.29	0.65
	Qwen	67.14	25.89	0.58	22.42	0.54	24.67	0.51	19.45	0.52	25.61	0.54
NLI	**Llama**	41.48	17.06	0.63	17.61	0.66	18.06	0.65	17.86	0.57	17.73	0.64
	Mistral	56.4	14.73	0.49	14.48	0.53	16.00	0.51	16.66	0.49	17.32	0.5
	Gemma	60.55	30.20	0.57	24.35	0.56	24.09	0.56	29.53	0.53	29.56	0.55
	Qwen	80.29	54.50	0.79	20.6	0.54	36.15	0.65	15.6	0.46	37.33	0.65

Table 14: Accuracy (Acc) and Perturbation Ration (PR) in the zero-shot setting using Feature Ablation for computing word importance. Here, Orig represents the Accuracy on the original test set.

Task		Orig	Superscript		Subscript		Super-Sub		Squiggle&Hooks		Ransom	
		Acc	Acc	PR	Acc	PR	Acc	PR	Acc	PR	Acc	PR
Sent	**Llama**	69.95	6.80	0.37	9.4	0.41	7.13	0.38	12.33	0.39	6.07	0.36
	Mistral	71.01	15.93	0.39	15.67	0.42	15.4	0.39	11.4	0.38	11.27	0.38
	Gemma	77.1	33.94	0.60	16.75	0.45	23.87	0.55	9.67	0.39	44.27	0.75
	Qwen	74.55	43.47	0.75	16.73	0.47	23.00	0.56	14.53	0.41	20.8	0.54
	Gemini	79.37	66.93	0.89	70.07	0.87	62.2	0.85	27.93	0.58	61.67	0.85
	GPT 3.5	70.79	33.00	0.60	16.93	0.44	18.53	0.46	15.4	0.37	14.2	0.41
HS	**Llama**	67.68	-	-	-	-	-	-	-	-	-	-
	Mistral	70.25	29.63	0.51	27.63	0.51	27.63	0.49	24.81	0.50	15.6	0.45
	Gemma	72.12	53.78	0.79	23.73	0.53	34.02	0.61	25.31	0.45	33.94	0.62
	Qwen	67.4	37.93	0.67	18.76	0.50	25.15	0.55	20.33	0.43	27.05	0.57
	Gemini	71.15	62.41	0.89	62.24	0.89	60.58	0.87	43.15	0.74	57.26	0.84
	GPT 3.5	64.04	39.09	0.75	17.93	0.62	17.68	0.61	13.11	0.57	11.45	0.55
FV	**Llama**	50.41	9.55	0.43	12.59	0.48	11.46	0.45	16.32	0.47	10.86	0.43
	Mistral	55.99	17.36	0.57	22.92	0.61	18.76	0.58	16.84	0.56	16.75	0.53
	Gemma	56.22	42.19	0.79	22.14	0.60	26.39	0.65	29.25	0.59	34.29	0.73
	Qwen	54.42	31.77	0.70	18.40	0.55	23.35	0.61	14.15	0.49	23.18	0.60
	Gemini	58.02	37.19	0.67	37.13	0.67	36.43	0.65	26.17	0.58	36.97	0.65
	GPT 3.5	39.08	13.19	0.47	8.68	0.40	9.55	0.42	6.25	0.37	7.47	0.36
AM	**Llama**	61.28	8.51	0.42	8.35	0.44	8.14	0.43	9.85	0.44	10.27	0.43
	Mistral	62.39	20.86	0.50	25.34	0.56	24.66	0.55	23.63	0.54	12.91	0.46
	Gemma	73.65	59.85	0.86	18.02	0.54	32.35	0.64	32.56	0.57	50.00	0.78
	Qwen	67.14	43.49	0.75	15.54	0.46	22.34	0.55	13.15	0.45	22.73	0.56
	Gemini	59.23	56.82	0.82	54.74	0.82	52.16	0.78	36.76	0.70	50.4	0.75
	GPT 3.5	56.88	12.6	0.43	10.5	0.42	10.47	0.42	7.38	0.40	9.89	0.43
NLI	**Llama**	41.48	10.97	0.37	11.13	0.51	10.6	0.45	13.27	0.44	11.4	0.46
	Mistral	56.4	12.4	0.35	12.53	0.37	16.4	0.44	16.87	0.43	16.87	0.44
	Gemma	60.55	59.47	0.83	25.93	0.57	21.87	0.51	24.27	0.51	22.73	0.50
	Qwen	80.29	64.33	0.86	19.87	0.56	35.73	0.66	10.07	0.46	38.73	0.68
	Gemini	60.04	58.9	0.95	58.25	0.94	57.2	0.94	34.95	0.73	57.46	0.94
	GPT 3.5	16.67	-	-	-	-	-	-	-	-	-	-

Table 15: Accuracy (Acc) and Perturbation Ration (PR) in the zero-shot setting using LLM Self-Attribution for computing word importance. Here, Orig represents the Accuracy on the original test set.

Task		Orig	Superscript		Subscript		Super-Sub		Squiggle&Hooks		Ransom	
		Acc	Acc	PR	Acc	PR	Acc	PR	Acc	PR	Acc	PR
Sent	**Llama**	70.73	17.61	0.65	21.06	0.67	18.14	0.65	10.03	0.62	21.14	0.68
	Mistral	72.73	14.47	0.62	14.20	0.65	13.67	0.61	15.07	0.62	16.41	0.64
	Gemma	75.67	32.34	0.53	28.51	0.51	26.34	0.52	11.47	0.37	39.77	0.66
	Qwen	74.73	25.37	0.55	14.8	0.52	22.33	0.58	15.2	0.47	22.13	0.58
HS	**Llama**	67.68	25.39	0.58	21.58	0.54	21.82	0.54	24.01	0.55	21.82	0.55
	Mistral	70.46	23.84	0.64	20.43	0.68	19.61	0.63	16.59	0.59	15.50	0.58
	Gemma	77.93	40.34	0.62	37.12	0.58	35.68	0.58	28.65	0.48	35.43	0.59
	Qwen	71.37	40.91	0.7	25.80	0.62	28.63	0.63	25.22	0.57	28.71	0.62
FV	**Llama**	51.73	13.80	0.43	16.06	0.5	14.06	0.44	12.53	0.46	13.42	0.37
	Mistral	58.97	16.92	0.54	14.84	0.52	16.23	0.53	15.90	0.56	17.01	0.53
	Gemma	63.25	40.38	0.72	26.36	0.52	27.26	0.678	30.61	0.58	45.57	0.65
	Qwen	57.14	26.12	0.57	23.69	0.53	25.60	0.56	23.00	0.49	24.04	0.54
AM	**Llama**	64.28	26.33	0.48	20.48	0.47	16.19	0.48	20.94	0.54	13.89	0.41
	Mistral	62.01	19.49	0.61	16.34	0.58	18.41	0.59	15.68	0.54	14.06	0.51
	Gemma	74.79	36.61	0.64	32.77	0.68	33.04	0.65	34.23	0.66	40.21	0.69
	Qwen	71.42	28.79	0.55	25.36	0.53	26.43	0.54	20.70	0.5	24.81	0.56
NLI	**Llama**	46.2	18.66	0.56	18.53	0.59	18.8	0.58	21.11	0.54	19.26	0.57
	Mistral	64.53	10.20	0.48	10.93	0.53	11.00	0.5	12.2	0.48	11.65	0.45
	Gemma	76.33	32.65	0.58	30.89	0.58	29.15	0.59	28.32	0.6	35.12	0.69
	Qwen	85.67	59.53	0.82	23.00	0.58	39.63	0.7	15.53	0.51	40.26	0.68

Table 16: Accuracy (Acc) and Perturbation Ration (PR) in the few-shot setting using Feature Ablation for computing word importance. Here, Orig represents the Accuracy on the original test set.

Task		Orig	Superscript		Subscript		Super-Sub		Squiggle&Hooks		Ransom	
		Acc	Acc	PR	Acc	PR	Acc	PR	Acc	PR	Acc	PR
Sent	**Llama**	70.73	5.93	0.37	8.53	0.40	6.13	0.37	8.40	0.36	6.00	0.36
	Mistral	72.73	11.87	0.41	18.27	0.43	13.87	0.40	13.4	0.39	11.33	0.41
	Gemma	75.67	42.4	0.72	19.73	0.47	24.00	0.56	11.73	0.40	42.27	0.74
	Qwen	74.73	42.4	0.72	19.73	0.47	24.93	0.54	20.07	0.42	23.07	0.53
	Gemini	79.72	65.6	0.88	63.27	0.87	62.33	0.86	31.93	0.62	63.87	0.87
	GPT 3.5	71.15	40.27	0.69	24.33	0.53	27.27	0.56	24.67	0.48	22.73	0.50
HS	**Llama**	-	-	-	-	-	-	-	-	-	-	-
	Mistral	70.46	20.58	0.46	15.35	0.46	14.44	0.43	11.62	0.42	12.2	0.42
	Gemma	77.93	51.78	0.73	23.15	0.51	32.7	0.57	28.22	0.47	33.44	0.59
	Qwen	71.37	41.58	0.71	25.98	0.54	32.12	0.60	32.12	0.54	31.04	0.61
	Gemini	75.57	63.07	0.86	61.33	0.85	60.83	0.83	50.54	0.77	58.59	0.82
	GPT 3.5	69.22	36.85	0.73	15.85	0.57	16.35	0.58	8.05	0.47	9.21	0.48
FV	**Llama**	51.73	9.24	0.42	12.18	0.46	12.45	0.46	18.34	0.46	10.28	0.43
	Mistral	58.97	16.73	0.55	22.08	0.59	12.89	0.52	11.84	0.50	15.32	0.50
	Gemma	63.25	44.81	0.84	23.52	0.63	27.68	0.68	31.36	0.59	31.28	0.81
	Qwen	57.14	28.48	0.61	24.98	0.60	38.62	0.69	18.23	0.53	41.86	0.68
	Gemini	61.17	44.81	0.84	23.52	0.63	27.68	0.68	31.36	0.59	31.28	0.81
	GPT 3.5	45.17	19.1	0.59	14.67	0.55	15.36	0.54	11.46	0.51	11.67	0.51
AM	**Llama**	64.28	13.88	0.47	13.65	0.49	13.57	0.48	20.81	0.51	12.43	0.46
	Mistral	62.01	12.07	0.44	13.49	0.46	12.91	0.45	12.36	0.44	13.17	0.41
	Gemma	74.79	37.4	0.67	30.6	0.64	33.98	0.66	34.51	0.59	49.29	0.77
	Qwen	71.42	27.94	0.59	25.58	0.58	23.68	0.57	18.1	0.48	23.55	0.57
	Gemini	73.35	59.98	0.83	57.68	0.82	55.97	0.81	46.74	0.73	53.76	0.80
	GPT 3.5	59.02	19.79	0.52	17.21	0.51	16.24	0.51	14.77	0.49	17.36	0.52
NLI	**Llama**	46.2	6.60	0.43	8.20	0.50	6.93	0.47	9.33	0.38	6.47	0.42
	Mistral	64.53	12.27	0.40	12.2	0.42	12.4	0.40	12.53	0.39	13.07	0.40
	Gemma	76.33	67.87	0.93	20.2	0.56	27.73	0.63	19.73	0.45	44.27	0.77
	Qwen	85.67	65.87	0.87	17.13	0.56	35.67	0.66	11.93	0.46	38.8	0.69
	Gemini	72.5	65.05	0.94	62.7	0.92	61.65	0.91	34.82	0.71	61.78	0.92
	GPT 3.5	16.67	-	-	-	-	-	-	-	-	-	-

Table 17: Accuracy (Acc) and Perturbation Ration (PR) in the few-shot setting using LLM Self-Attribution for computing word importance. Here, Orig represents the Accuracy on the original test set.

Attack Type	#Samples	#Readable	%Readable
Superscript	50	50	100
Subscript	50	50	100
Super-sub	50	46	92
Squiggle & Hooks	50	44	88%
Ransom	50	49	98%
Character Insertion	50	33	66%
Character Deletion	50	30	60%

Table 18: Human readability assessment for `TinyAttack` perturbations. Most examples remain fully understandable to human readers.

		Label consistency	Semantic consistency (1-5)
Sent	Original	91	-
	Squiggle	90	4.66
	Ransom	90	4.98
	Super	91	5
	Sub	91	5
	Super-sub	89	4.76

Table 19: Human Evaluation scores for Label and Semantic consistency.

Attack Type	Original	Perturbed
Char-insertion	An animal is outside in the cold weather, playing with a plastic toy.	An3 aniemal zis outsidet in5 thee co7ld wfeather, playxing wfith a8 plwastic 9toy.
	There are people outside.	There arej people outside.
	Bruce Springsteen is a singer.	Bruxce Spribngsteen vis aq singezr.
Char-insertion	The man is driving a truck.	The ma s driing tuck.
	The man pleads for mercy.	Th mn pleas fr mery.

Table 20: Examples from standard character deletion and insertion based perturbations.

Attack	Example	Mistral	Qwen	M / Q
-	I am not crazy about the smoothies though.	['__I', '__am', '__not', '__crazy', '__about', '__the', '__smooth', 'ies', '__though', '.']	['I', 'Ġam', 'Ġnot', 'Ġcrazy', 'Ġabout', 'Ġthe', 'Ġsmooth', 'ies', 'Ġthough', '.']	10 / 10
Super	I am not crazy about the smoothies though.	['__', '<0xE1>', '<0xB4>', '<0xB5>', '__', '<0xE1>', '<0xB5>', '<0x83>', '<0xE1>', '<0xB5>', '<0x90>', '__', '<0xE2>', '<0x81>', '<0xBF>', '<0xE1>', '<0xB5>', '<0x92>', '<0xE1>', '<0xB5>', '<0x97>', '__crazy', '__about', '__', '<0xE1>', '<0xB5>', '<0x97>', 'ʰ', '<0xE1>', '<0xB5>', '<0x89>', '__', '<0xCB>', '<0xA2>', '<0xE1>', '<0xB5>', '<0x90>', '<0xE1>', '<0xB5>', '<0x92>', '<0xE1>', '<0xB5>', '<0x92>', '<0xE1>', '<0xB5>', '<0x97>', 'ʰ', '<0xE1>', '<0xB6>', '<0xA6>', '<0xE1>', '<0xB5>', '<0x89>', '<0xCB>', '<0xA2>', '__', '<0xE1>', '<0xB5>', '<0x97>', 'ʰ', '<0xE1>', '<0xB5>', '<0x92>', '<0xE1>', '<0xB5>', '<0x98>', '<0xE1>', '<0xB5>', '<0x8D>', 'ʰ', '.']	['I', 'Ġa', 'âĤ', 'Ĩ', 'Ġâ', 'Ĥ', 'Ļ', 'âĤĴ', 'âĤ', 'Ī', 'Ġc', 'áµ', '£', 'azy', 'Ġab', 'âĤĴ', 'áµ¤', 'âĤ', 'Ī', 'Ġâ', 'Ĥ', 'Ī', 'âĤ', 'ķ', 'âĤij', 'Ġâ', 'Ĥ', 'Ľ', 'âĤ', 'Ĩ', 'âĤĴ', 'âĤĴ', 'âĤ', 'Ī', 'âĤ', 'ķ', 'áµ', '¢', 'âĤij', 'âĤĽ', 'Ġâ', 'Ĥ', 'Ī', 'âĤ', 'ķ', 'âĤĴ', 'áµ¤', 'g', 'âĤ', 'ķ']	71 / 50
Sub	I am not crazy about the smoothies though.	['__I', '__a', '<0xE2>', '<0x82>', '<0x98>', '__', '<0xE2>', '<0x82>', '<0x99>', '<0xE2>', '<0x82>', '<0x92>', '<0xE2>', '<0x82>', '<0x9C>', '__c', '<0xE1>', '<0xB5>', '<0xA3>', 'azy', '__ab', '<0xE2>', '<0x82>', '<0x92>', '<0xE1>', '<0xB5>', '<0xA4>', '<0xE2>', '<0x82>', '<0x9C>', '__', '<0xE2>', '<0x82>', '<0x9C>', '<0xE2>', '<0x82>', '<0x95>', '<0xE2>', '<0x82>', '<0x91>', '__', '<0xE2>', '<0x82>', '<0x9B>', '<0xE2>', '<0x82>', '<0x98>', '<0xE2>', '<0x82>', '<0x92>', '<0xE2>', '<0x82>', '<0x92>', '<0xE2>', '<0x82>', '<0x9C>', '<0xE2>', '<0x82>', '<0x95>', '<0xE1>', '<0xB5>', '<0xA2>', '<0xE2>', '<0x82>', '<0x91>', '<0xE2>', '<0x82>', '<0x9B>', '__', '<0xE2>', '<0x82>', '<0x9C>', '<0xE2>', '<0x82>', '<0x95>', '<0xE2>', '<0x82>', '<0x92>', '<0xE1>', '<0xB5>', '<0xA4>', 'g', '<0xE2>', '<0x82>', '<0x95>']	['I', '__a', '<0xE2>', '<0x82>', '<0x98>', '__', '<0xE2>', '<0x82>', '<0x99>', '<0xE2>', '<0x82>', '<0x92>', 'ₜ', '__c', '<0xE1>', '<0xB5>', '<0xA3>', 'azy', '__ab', '<0xE2>', '<0x82>', '<0x92>', '<0xE1>', '<0xB5>', '<0xA4>', 'ₜ', '__', 'ₜ', 'ₕ', '<0xE2>', '<0x82>', '<0x91>', '__', '<0xE2>', '<0x82>', '<0x9B>', '<0xE2>', '<0x82>', '<0x98>', '<0xE2>', '<0x82>', '<0x92>', '<0xE2>', '<0x82>', '<0x92>', 'ₜ', 'ₕ', 'ᵢ', '<0xE2>', '<0x82>', '<0x91>', '<0xE2>', '<0x82>', '<0x9B>', '__', 'ₜ', 'ₕ', '<0xE2>', '<0x82>', '<0x92>', '<0xE1>', '<0xB5>', '<0xA4>', 'g', 'ₕ', '.']	85 / 66
Super-Sub	I am not crazy about the smoothies though.	['__', '<0xE1>', '<0xB4>', '<0xB5>', '__am', '__n', '<0xE2>', '<0x82>', '<0x92>', 't', '__c', '<0xE1>', '<0xB5>', '<0xA3>', 'a', '<0xE1>', '<0xB6>', '<0xBB>', 'y', '__a', '<0xE1>', '<0xB5>', '<0x87>', 'o', '<0xE1>', '<0xB5>', '<0xA4>', 't', '__t', '<0xE2>', '<0x82>', '<0x95>', 'e', '__s', '<0xE2>', '<0x82>', '<0x98>', 'o', '<0xE1>', '<0xB5>', '<0x92>', 't', '<0xE2>', '<0x82>', '<0x95>', 'i', '<0xE1>', '<0xB5>', '<0x89>', 's', '__t', 'ʰ', 'o', '<0xE1>', '<0xB5>', '<0xA4>', 'g', 'ʰ', '.']	['á´µ', 'Ġam', 'Ġn', 'âĤĴ', 't', 'Ġc', 'áµ', '£', 'a', 'á¶»', 'y', 'Ġa', 'áµĩ', 'o', 'áµ¤', 't', 'Ġt', 'âĤ', 'ķ', 'e', 'Ġs', 'âĤ', 'Ĩ', 'o', 'áµĴ', 't', 'âĤ', 'ķ', 'i', 'áµī', 's', 'Ġt', 'Ê°', 'o', 'áµ¤', 'g', 'Ê°', '.']	59 / 38
SuiggleHooks	I̢ ąḿ ήǫt crazy about ťⱨẻ śmơǫťⱨïęś ťⱨơuɠⱨ.	['__I', '<0xCC>', '<0xA2>', '__', '<0xE1>', '<0xB6>', '<0x8F>', '<0xE1>', '<0xB8>', '<0xBF>', '__', '<0xE1>', '<0xBD>', '<0xB5>', 'ǫ', 'ť', '__crazy', '__about', '__', 'ť', '<0xE2>', '<0xB1>', '<0xA8>', '<0xE1>', '<0xBA>', '<0xBB>', '__ś', '<0xE1>', '<0xB6>', '<0x86>', 'ơ', 'ǫ', 'ť', '<0xE2>', '<0xB1>', '<0xA8>', 'ï', '<0xE1>', '<0xB6>', '<0x92>', 'ś', '__', 'ť', '<0xE2>', '<0xB1>', '<0xA8>', 'ơ', '<0xE1>', '<0xB6>', '<0x99>', '<0xC9>', '<0xA0>', '<0xE2>', '<0xB1>', '<0xA8>', '.']	['I', 'Ì', '¢', 'Ġá', '¶', 'ı', 'á¸¿', 'ĠÎ', '®', 'Ç«', 'Å¥', 'Ġcrazy', 'Ġabout', 'ĠÅ', '¥', 'â±', '¨', 'áº»', 'ĠÅĽ', 'á¶Ĩ', 'Æ¡', 'Ç«', 'Å¥', 'â±', '¨', 'á»Ī', 'á¶', 'Ĵ', 'ÅĽ', 'ĠÅ', '¥', 'â±', '¨', 'Æ¡', 'á¶', 'Ļ', 'É', 'ł', 'â±', '¨', '.']	56 / 40
Ransom	**I am** *not* crazy about *the smoothiEs* tHoUgh.	['__', '<0xF0>', '<0x9D>', '<0x90>', '<0x88>', '__', '<0xF0>', '<0x9D>', '<0x90>', '<0x9A>', '<0xF0>', '<0x9D>', '<0x97>', '<0xBA>', '__', '<0xF0>', '<0x9D>', '<0x93>', '<0x83>', '<0xF0>', '<0x9D>', '<0x98>', '<0xB0>', '<0xF0>', '<0x9D>', '<0x90>', '<0xAD>', '__crazy', '__about', '__', '<0xF0>', '<0x9D>', '<0x91>', '<0xA1>', '<0xF0>', '<0x9D>', '<0x99>', '<0x9D>', '<0xF0>', '<0x9D>', '<0x92>', '<0x86>', '__', '<0xF0>', '<0x9D>', '<0x93>', '<0x88>', '<0xF0>', '<0x9D>', '<0x94>', '<0xAA>', '<0xF0>', '<0x9D>', '<0x94>', '<0xAC>', '<0xE2>', '<0x84>', '<0xB4>', '<0xF0>', '<0x9D>', '<0x99>', '<0xA9>', '<0xF0>', '<0x9D>', '<0x96>', '<0x8D>', '<0xF0>', '<0x9D>', '<0x96>', '<0x8E>', '<0xF0>', '<0x9F>', '<0x84>', '<0xB4>', '<0xF0>', '<0x9D>', '<0x99>', '<0xA8>', '__', '<0xF0>', '<0x9D>', '<0x9A>', '<0x9D>', '<0xF0>', '<0x9F>', '<0x85>', '<0x97>', '<0xF0>', '<0x9D>', '<0x98>', '<0xB0>', '<0xF0>', '<0x9F>', '<0x85>', '<0xA4>', '<0xF0>', '<0x9D>', '<0x94>', '<0xA4>', '<0xF0>', '<0x9D>', '<0x9A>', '<0x91>', '.']	['ðĿIJĪ', 'ĠðĿ', 'IJ', 'ļ', 'ðĿĹº', 'ĠðĿ', 'ĵ', 'ĥ', 'ðĿĺ°', 'ðĿIJŃ', 'Ġcrazy', 'Ġabout', 'ĠðĿ', 'ij', '¡', 'ðĿĻĿ', 'ĠðĿ', 'ĵĪ', 'ðĿĶª', 'ðĿĶ¬', 'âĦ´', 'ðĿĻ©', 'ðĿĸį', 'ðĿĸİ', 'ðŁĦ´', 'ðĿĻ¨', 'ĠðĿ', 'ļ', 'Ŀ', 'ðŁħ', 'Ĺ', 'ðĿĺ°', 'ðŁħ¤', 'ðĿĶ¤', 'ðĿļij', '.']	104/42

Figure 23: Tokens generated by Mistral (M) and Qwen (Q). The last column shows the count of subword tokens.

Attack	Example	GPT	Gemini	GPT / GEMINI
-	I am not crazy about the smoothies though.	['I', ' am', ' not', ' crazy', ' about', ' the', ' smooth', 'ies', ' though', '.']	-	10/10
Super	I am not crazy about the smoothies though.	['<0xE1>', '<0xB4>', '<0xB5>', '<0x20><0xE1>', '<0xB5>', '<0x83>', '<0xE1>', '<0xB5>', '<0x90>', '<0x20><0xE2><0x81>', '<0xBF>', '<0xE1>', '<0xB5>', '<0x92>', '<0xE1>', '<0xB5>', '<0x97>', ' crazy', ' about', '<0x20><0xE1>', '<0xB5>', '<0x97>', '<0xCA>', '<0xB0>', '<0xE1>', '<0xB5>', '<0x89>', ' ', '<0xCB>', '<0xA2>', '<0xE1>', '<0xB5>', '<0x90>', '<0xE1>', '<0xB5>', '<0x92>', '<0xE1>', '<0xB5>', '<0x92>', '<0xE1>', '<0xB5>', '<0x97>', '<0xCA>', '<0xB0>', '<0xE1>', '<0xB6>', '<0xA6>', '<0xE1>', '<0xB5>', '<0x89>', '<0xCB>', '<0xA2>', '<0x20><0xE1>', '<0xB5>', '<0x97>', '<0xCA>', '<0xB0>', '<0xE1>', '<0xB5>', '<0x92>', '<0xE1>', '<0xB5>', '<0x98>', '<0xE1>', '<0xB5>', '<0x8D>', '<0xCA>', '<0xB0>', '.']	-	69/33
Sub	I am not crazy about the smoothies though.	['I', ' a', '<0xE2><0x82>', '<0x98>', '<0x20><0xE2>', '<0x82>', '<0x99>', '<0xE2><0x82>', '<0x92>', '<0xE2><0x82>', '<0x9C>', ' c', '<0xE1>', '<0xB5>', '<0xA3>', 'azy', ' ab', '<0xE2><0x82>', '<0x92>', '<0xE1>', '<0xB5>', '<0xA4>', '<0xE2><0x82>', '<0x9C>', '<0x20><0xE2>', '<0x82>', '<0x9C>', '<0xE2><0x82>', '<0x95>', '<0xE2><0x82>', '<0x91>', '<0x20><0xE2>', '<0x82>', '<0x9B>', '<0xE2><0x82>', '<0x98>', '<0xE2><0x82>', '<0x92>', '<0xE2><0x82>', '<0x92>', '<0xE2><0x82>', '<0x9C>', '<0xE2><0x82>', '<0x95>', '<0xE1>', '<0xB5>', '<0xA2>', '<0xE2><0x82>', '<0x91>', '<0xE2><0x82>', '<0x9B>', '<0x20><0xE2>', '<0x82>', '<0x9C>', '<0xE2><0x82>', '<0x95>', '<0xE2><0x82>', '<0x92>', '<0xE1>', '<0xB5>', '<0xA4>', 'g', '<0xE2><0x82>', '<0x95>']	-	64/66
Super-Sub	I am not crazy about the smoothies though.	['<0xE1>', '<0xB4>', '<0xB5>', ' am', ' n', '<0xE2><0x82>', '<0x92>', 't', ' c', '<0xE1>', '<0xB5>', '<0xA3>', 'a', '<0xE1>', '<0xB6>', '<0xBB>', 'y', ' a', '<0xE1>', '<0xB5>', '<0x87>', 'o', '<0xE1>', '<0xB5>', '<0xA4>', 't', ' t', '<0xE2><0x82>', '<0x95>', 'e', ' s', '<0xE2><0x82>', '<0x98>', 'o', '<0xE1>', '<0xB5>', '<0x92>', 't', '<0xE2><0x82>', '<0x95>', 'i', '<0xE1>', '<0xB5>', '<0x89>', 's', ' t', '<0xCA>', '<0xB0>', 'o', '<0xE1>', '<0xB5>', '<0xA4>', 'g', '<0xCA>', '<0xB0>', '.']	-	56/47
SuiggleHooks	I am not crazy about the smoothies though.	['I', '<0xCC>', '<0xA2>', '<0x20><0xE1>', '<0xB6>', '<0x8F>', '<0xE1>', '<0xB8>', '<0xBF>', '<0x20><0xE1>', '<0xBD>', '<0xB5>', '<0xC7>', '<0xAB>', 't', ' crazy', ' about', '<0x20><0xC5>', '<0xA5>', '<0xE2>', '<0xB1>', '<0xA8>', '<0xE1><0xBA>', '<0xBB>', ' s', '<0xE1>', '<0xB6>', '<0x86>', 'o', '<0xC7>', '<0xAB>', 't', '<0xE2>', '<0xB1>', '<0xA8>', 'l', '<0xE1>', '<0xB6>', '<0x92>', 's', '<0x20><0xC5>', '<0xA5>', '<0xE2>', '<0xB1>', '<0xA8>', 'o', '<0xE1>', '<0xB6>', '<0x99>', '<0xC9>', '<0xA0>', '<0xE2>', '<0xB1>', '<0xA8>', '.']	-	55/41
Ransom	I am not crazy about the smoothies though.	['<0xF0><0x9D>', '<0x90>', '<0x88>', '<0x20><0xF0><0x9D>', '<0x90>', '<0x9A>', '<0xF0><0x9D>', '<0x97>', '<0xBA>', '<0x20><0xF0><0x9D>', '<0x93>', '<0x83>', '<0xF0><0x9D>', '<0x98>', '<0xB0>', '<0xF0><0x9D>', '<0x90>', '<0xAD>', ' crazy', ' about', '<0x20><0xF0><0x9D>', '<0x91>', '<0xA1>', '<0xF0><0x9D>', '<0x99>', '<0x9D>', '<0xF0><0x9D>', '<0x92>', '<0x86>', '<0x20><0xF0><0x9D>', '<0x93>', '<0x88>', '<0xF0><0x9D>', '<0x94>', '<0xAA>', '<0xF0><0x9D>', '<0x94>', '<0xAC>', '<0xE2><0x84>', '<0xB4>', '<0xF0><0x9D>', '<0x99>', '<0xA9>', '<0xF0><0x9D>', '<0x96>', '<0x8D>', '<0xF0><0x9D>', '<0x96>', '<0x8E>', '<0xF0><0x9F>', '<0x84>', '<0xB4>', '<0xF0><0x9D>', '<0x99><0xA8>', '<0x20><0xF0><0x9D>', '<0x9A>', '<0x9D>', '<0xF0><0x9F>', '<0x85>', '<0x97>', '<0xF0><0x9D>', '<0x98>', '<0xB0>', '<0xF0><0x9F>', '<0x85>', '<0xA4>', '<0xF0><0x9D>', '<0x94>', '<0xA4>', '<0xF0><0x9D>', '<0x9A>', '<0xA4>', '.']	-	73/50

Figure 24: Tokens generated by GPT and Gemini models.

```
You are an assistant that replaces homoglyphs characters with their Latin counterparts in a given text.

For example you might receive text like:
'''happy at work *conference: right mindset leads to culture-of-development organizations #work #mindset'''
And you should update it to: happy at work *conference: right mindset leads to culture-of-development organizations #work #mindset

Or you may receive text like: '''enjoy your life ..alex s. #day #business #relax #lifestyle #feliz #felicidad #life'''
And you should update it to: enjoy your life ..alex s. #day #business #relax #lifestyle #feliz #felicidad #life

Or you may receive text like: '''@user @user @user absolutely no! at least 3 other guys were more desserving. proves once again league's favoritism toward him'''
And you should update it to: @user @user @user absolutely no! at least 3 other guys were more desserving. proves once again league's favoritism toward him

Or you may receive text like: '''@user , shocked by your ignorance '''
And you should update it to: @user , shocked by your ignorance

Or you may receive text like: '''waterfight. the only way to keep cool this summer! #child #kids #family #smile #instakids'''
And you should update it to: waterfight. the only way to keep cool this summer! #child #kids #family #smile #instakids

Or you may receive text like: '''(advanced value chain videos at ) #valuechain '''
And you should update it to: (advanced value chain videos at ) #valuechain

Or you may receive text like: '''@user i'm an introve person, i talk little, and like to be alone.ą-so ji-sub day cr:junnieuyendong @user htt'''
And you should update it to: @user i'm an introve person, i talk little, and like to be alone.ą-so ji-sub day cr:junnieuyendong @user htt

Or you may receive text like: '''guys! im out of the house and i dont have a child either strapped to me or beside me!! first time in almost three months'''
And you should update it to: guys! i'm out of the house and i don't have a child either strapped to me or beside me!! first time in almost three months

Or you may receive text like: '''checkout today's #trending #gif of the day! , cool, syfy, faceoff, really cool, this is really cool via bit'''
And you should update it to: checkout today's #trending #gif of the day! , cool, syfy, faceoff, really cool, this is really cool via bit

Or you may receive text like: '''#80 #thousand #care # workers in the #iuk are #immigrants and #most #elderly #people #say they cant even #understand #them …so #bloody'''
And you should update it to: #80 #thousand #care # workers in the #iuk are #immigrants and #most #elderly #people #say they cant even #understand #them so #bloody

Replace the homoglyphs in the following text delimited by triple single quotes.
'''{text}'''
```

Figure 25: Prompt used for unicode normalization.