Perturbations in the Wild: Leveraging Human-Written Text Perturbations for Realistic Adversarial Attack and Defense

Thai Le Jooyoung Lee Kevin Yen* Yifan Hu* Dongwon Lee

Penn State University {thaile, jfl5838, dongwon}@psu.edu
Yahoo Research* {kevinyen, yifanhu}@yahooinc.com*

Abstract

We proposes a novel algorithm, ANTHRO, that *inductively* extracts over 600K human-written text perturbations in the wild and leverages them for *realistic* adversarial attack. Unlike existing character-based attacks which often deductively hypothesize a set of manipulation strategies, our work is grounded on actual observations from real-world texts. We find that adversarial texts generated by ANTHRO achieve the best trade-off between (1) attack success rate, (2) semantic preservation of the original text, and (3) stealthiness–i.e. indistinguishable from human writings hence harder to be flagged as suspicious. Specifically, our attacks accomplished around 83% and 91% attack success rates on BERT and RoBERTa, respectively. Moreover, it outperformed the *TextBugger* baseline with an increase of 50% and 40% in terms of semantic preservation and stealthiness when evaluated by both layperson and professional human workers. ANTHRO can further enhance a BERT classifier's performance in understanding different variations of human-written toxic texts via adversarial training when compared to the Perspective API. *Source code will be published at* `github.com/lethaiq/perturbations-in-the-wild`.

1 Introduction

Machine learning (ML) models trained to optimize only the prediction performance are often vulnerable to adversarial attacks (Papernot et al., 2016; Wang et al., 2019). In the text domain, especially, a character-based adversarial attacker aims to fool a target ML model by generating an adversarial text x^* from an original text x by manipulating characters of different words in x, such that some properties of x are preserved (Li et al., 2018; Eger et al., 2019; Gao et al., 2018). We characterize strong and practical adversarial attacks as three criteria: (1) *attack performance*, as measured by the ability to flip a target model's predictions, (2) *semantic preservation*, as measured by the ability to preserve the meaning of an original text, and (3) *stealthiness*, as measured by how unlikely it is detected as machine-manipulation and removed by defense systems or human examiners (Figure 1). While the first two criteria are natural derivation from adversarial literature (Papernot et al., 2016), stealthiness is also important to be a practical attack under a mass-manipulation scenario. In fact, adversarial text generation remains a challenging task under practical settings.

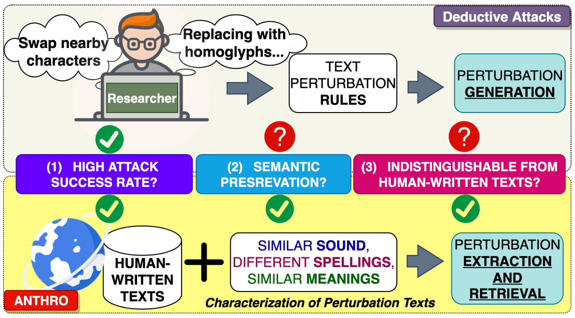

Deductive Attacks
Swap nearby characters
Replacing with homoglyphs...
Researcher
TEXT PERTURBATION RULES
PERTURBATION GENERATION
?
?
(1) HIGH ATTACK SUCCESS RATE?
(2) SEMANTIC PRESREVATION?
(3) INDISTINGUISHABLE FROM HUMAN-WRITTEN TEXTS?
HUMAN-WRITTEN TEXTS
+
SIMILAR SOUND, DIFFERENT SPELLINGS, SIMILAR MEANINGS
PERTURBATION EXTRACTION AND RETRIEVAL
ANTHRO
Characterization of Perturbation Texts

Figure 1: ANTHRO (Bottom) extracts and uses human-written perturbations for adversarial attacks instead of proposing a specific set of manipulation rules (Top).

Previously proposed character-based attacks follow a *deductive* approach where the researchers hypothesize a set of text manipulation strategies that exploit some vulnerabilities of textual ML models (Figure 1). Although these deductively derived techniques can demonstrate superior attack performance, there is no guarantee that they also perform well with regard to semantic preservation and stealthiness. We first analyze why enforcing these properties are challenging especially for character-based attacks.

To preserve the semantic meanings, an attacker can minimize the distance between representative vectors learned from a large pre-trained model–e.g., Universal Sentence Encoder (Cer et al., 2018) of the two sentences. However, this is only applicable in word- or sentence-based attacks, not in

DISCLAIMER! THIS PAPER CONTAINS EXAMPLE TEXTS THAT ARE OFFENSIVE IN NATURE character-based attacks. It is because character-based manipulated tokens are more prone to become out-of-distribution–e.g., morons→mor0ns, from what is observed in a typical training corpus where the correct use of English is often assumed. In fact, existing character-based attacks such as *TextBugger* (Li et al., 2018), *VIPER* (Eger et al., 2019) and *DeepWordBug* (Gao et al., 2018) generally assume that the meaning of the original sentence is preserved without further evaluations.

In addition, a robust ML pipeline is often equipped to detect and remove potential adversarial perturbations either via automatic software (Jayanthi et al., 2020; Pruthi et al., 2019), trapdoors (Le et al., 2021) or human-in-the-loop (Le et al., 2020). Such detection is feasible especially when the perturbed texts are curated using a set of fixed rules that can be easily re-purposed for defense. Thus, attackers such as *VIPER* and *DeepWordBug*, which map each Latin-based character to either non-English accents (e.g., ė, ā, d̃), or homoglyphs (characters of similar shape), fall into this category and can be easily detected under simple normalization techniques (Sec. 4.1). *TextBugger* circumvents this weakness by utilizing a set of more general character-editing strategies–e.g., replacing and swapping nearby characters to synthesize human-written typos and misspellings. Although texts perturbed by such strategies become less likely to be detected, many of them may distort the meaning of the original text (e.g., “garbage"→“gabrage", “dumb"→“dub") and can be easily flagged as machine-generated by human examiners. Therefore, we argue that generating perturbations that both preserve original meanings and are indistinguishable from human-written texts be a critically important yet challenging task.

To overcome these challenges, we introduce ANTHRO, a novel algorithm that *inductively* finds and extracts text perturbations in the wild. As shown in Figure 1, our method relies on human-written sentences in the Web in their raw form. We then use them to develop a character-based adversarial attack that is not only effective and realistic but is also helpful in training ML models that are more robust against a wide variety of human-written perturbations. Distinguished from previous research, our work considers both spellings and phonetic features (how a word sounds), to characterize text perturbations. Furthermore, we conducted user studies to quantitatively evaluate semantic preservation and stealthiness of adversarial texts. Our contributions are as follows.

- ANTHRO extracts over 600K case-sensitive character-based “real" perturbations from human-written texts.
- ANTHRO facilitates black-box adversarial attacks with an average of 82.7% and 90.7% attack success rates on BERT and RoBERTa, and drops the *Perspective API*'s precision to only 12%.
- ANTHRO outperforms the *TextBugger* baseline by over 50% in semantic preservation and 40% in stealthiness in human subject studies.
- ANTHRO combined with adversarial training also enables BERT classifier to achieve 3%–14% improvement in precision over *Perspective API* in understanding human-written perturbations.

2 Perturbations in the Wild

2.1 Machine v.s. Human Perturbations

Perturbations that are neither natural-looking nor resembling human-written texts are more likely to be detected by defense systems (thus not a practical attack from adversaries' perspective). However, some existing character-based perturbation strategies, including *TextBugger*, *VIPER* and *DeepWordBug*, follow a *deductive* approach and their generated texts often do not resemble human-written texts. Qualitatively, however, we find that humans express much more diverse and creative (Tagg, 2011) perturbations (Figure B.1, Appendix) than ones generated by such deductive approaches. For example, humans frequently (1) capitalize and change the parts of a word to emphasize distorted meanings (e.g.,“democrats“→“democRATs", “republicans"→“republiCUNTs"), (2) hyphenate a word (e.g., “depression"→“de-pres-sion"), (3) use emoticons to emphasize meaning (e.g., “shit"→“sh💩t"), (4) repeat particular characters (e.g., “dirty"→“diiirty", “porn"→“pooorn"), or (5) insert phonetically similar characters (e.g., “nigger"→“nighger"). Human-written perturbations do not manifest any fixed rules and often require some context understanding. Moreover, one can generate a new meaningful perturbation simply by repeating a character–e.g., “porn"→“pooorn". Thus, it is challenging to systematically generate all such perturbations, if not impossible. Moreover, it is very difficult for spell-checkers, which usually rely on a fixed set

Attacker	**Reddit Comts.**	**News Comts.**
#texts, #tokens	**»5B, N/A**	**(34M, 11M)**
TextBugger	51.6% (126/244)	7.10% (11K/152K)
VIPER	3.2% (1/31)	0.13% (25/19K)
DeepWordBug	0% (0/31)	0.27% (51/19K)
ANTHRO	**82.4%** (266/323)	**55.7%** (16K/29K)

Table 1: Percentage of offensive perturbed words generated by different attacks that can be observed in real human-written comments on Reddit and online news.

of common spelling mistakes and an edit-distance threshold, to correct and detect all human-written perturbations.

We later show that human examiners rely on personal exposure from Reddit or YouTube comments to decide if a word choice looks natural (Sec. 4.2). Quantitatively, we discover that not all the perturbations generated by deductive methods are observed on the Web (Table 1). To analyze this, we first use each attack to generate all possible perturbations of either (1) a list of over 3K unique offensive words or (2) a set of the top 5 offensive words ("c*nt", "b*tch", "m*therf***er", "bast*rd", "d*ck"). Then, we calculate how many of the perturbed words are present in a dataset of over 34M online news comments or are used by at least 50 unique commentators on Reddit, respectively. Even though *TextBugger* was well-known to simulate human-written typos as adversarial texts, merely 51.6% and 7.1% of its perturbations are observed on Reddit and online news comments, implying *TextBugger*'s generated adversarial texts being "unnatural" and "easily-detectable" by human-in-the-loop defense systems.

2.2 The SMS Property: Similar Sound, Similar Meaning, Different Spelling

The existence of a non-arbitrary relationship between sounds and meanings has been proven by a life-long research establishment (Köhler, 1967; Jared and Seidenberg, 1991; Gough et al., 1972). In fact, Blasi et al. (2016) analyzed over 6K languages and discovered a high correlation between a word's sound and meaning both inter- and intra-cultures. Aryani et al. (2020) found that how a word sounds links to an individual's emotion. This motivates us to hypothesize that words spelled differently yet have the same meanings such as text perturbations will also have similar sounds.

Figure B.1 (Appendix) displays several perturbations that are found from real-life texts. Even though these perturbations are *spelled differently* from the original word, they all preserve *similar meanings* when perceived by humans. Such semantic preservation is feasible because humans perceive these variations *phonetically similar* to the respective original words (Van Orden, 1987). For example, both "republican" and "republikan" sound similar when read by humans. Therefore, given the surrounding context of a perturbed sentence–e.g., "*President Trump is a* republikan", and the phonetic similarity of "republican" and "republikan", end-users are more likely to interpret the perturbed sentence as "*President Trump is a republican*". We call these characteristics of text perturbations the *SMS* property: "*similar Sound, similar Meaning, different Spellings*". Noticeably, the SMS characterization includes a subset of "visually similar" property of perturbations as studied in previous adversarial attacks such as *TextBugger* (e.g., "hello" sounds similar with "he11o"), *VIPER* and *DeepWordBug*. However, two words that look very similar sometimes carry different meanings–e.g., "garbage"→"gabrage". Moreover, our characterization is also distinguished from *homophones* (e.g., "to" and "two") which describe words with similar sound yet *different meaning*.

3 A Realistic Adversarial Attack

Given the above analysis, we now derive our proposed ANTHRO adversarial attack. We first share how to systematically encode the sound–i.e., phonetic feature, of any given words and use it to search for their human-written perturbations that satisfy the SMS property. Then, we introduce an iterative algorithm that utilizes the extracted perturbations to attack textual ML models.

3.1 Mining Perturbations in the Wild

Sound Encoding with SOUNDEX++. To capture the sound of a word, we adopt and extend the case-insensitive SOUNDEX algorithm. SOUNDEX helps index a word based on how it sounds rather than how it is spelled (Stephenson, 1980). Given a word, SOUNDEX first keeps the 1st character. Then, it removes all vowels and matches the remaining characters *one by one* to a digit following a set of predefined rules–e.g., "B", "F"→1, "D", "T"→3 (Stephenson, 1980). For example, "Smith" and "Smyth" are both encoded as S530.

As the SOUNDEX system was designed mainly for encoding surnames, it does not necessarily

Word	SOUNDEX	SOUNDEX++ (Ours)
porn	P650	P650 (**k**=0), PO650 (**k**=1)
p0rn	P065(✗)	(same as above)
lesbian	L215	L245 (**k**=0), LE245 (**k**=1)
lesbbi@n	L21@(✗)	(same as above)
losbian	L215(✗)	L245 (**k**=0), LO245 (**k**=1)

(✗): Incorrect encoding

Table 2: SOUNDEX++ can capture visually similar characters and is more accurate in differentiating between desired (blue) and undesired (red) perturbations.

Key	TH000	DE5263	AR000	DI630	NO300
Value (Set)	the	democrats demokRATs	are arre	dirty dirrrty	not
ANTHRO(democrats,**k**=1,**d**=1)→{democrats, demokRATs}					
ANTHRO(dirty,**k**=1,**d**=2)→{dirty, dirrrty}					

Table 3: Examples of hash table $H_1(k{=}1)$ curated from sentences *"the demokRATs are dirrrty"* and *"the democrats arre not dirty"* and its utilization.

work for texts in the wild. For example, it cannot recognize visually-similar perturbations such as "l"→"1", "a"→"@" and "O"→"0". Moreover, it always fixes the 1st character as part of the final encodes. This rule is too rigid and can result in words that are entirely different yet encoded the same (Table 2). To solve these issues, we propose a new SOUNDEX++ algorithm. SOUNDEX++ is equipped to both recognize visually-similar characters and encode the sound of a word at different hierarchical levels **k** (Table 2). Particularly, at level **k**=0, SOUNDEX++ works similar to SOUNDEX by fixing the first character. At level **k**≥1, SOUNDEX++ instead fixes the first **k**+1 characters and encodes the rest.

Levenshtein Distance d and Phonetic Level k as a Semantic Preservation Proxy. Since SOUNDEX++ is not designed to capture a word's semantic meaning, we utilize both phonetic parameter **k** and *Levenshtein distance* **d** (Levenshtein et al., 1966) as a heuristic approximation to measure the semantic preservation between two words. Intuitively, the higher the phonetic level (**k**≥1) at which two words share the same SOUNDEX++ code and the smaller the Levenshtein distance **d** to transform one word to another, the more likely human associate them with the meaning. In other words, **k** and **d** are hyper-parameters that help control the trade-off between precision and recall when retrieving perturbations of a given word such that they satisfy the SMS property (Figure 2). We will later carry out a human study to evaluate how well our extracted perturbations can preserve the semantic meanings in practice.

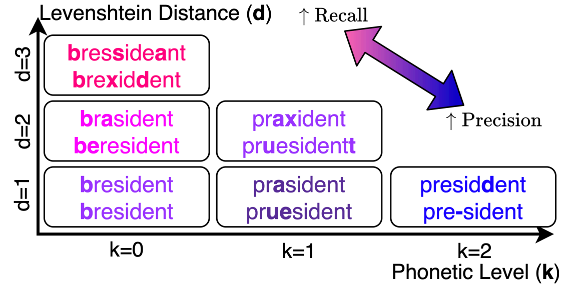


Figure 2: Trade-off between precision and recall of extracted perturbations for the word "president" w.r.t different **k** and **d** values. Higher **k** and lower **d** associate with better preservation of the original meaning.

Mining from the Wild. To mine all human-written perturbations, we first collect a large corpus $\mathcal{D}$ of over 18M sentences written by netizens from 9 different datasets (Table A.1 in Appendix). We select these datasets because they include offensive texts such as hate speech, sensitive search queries, etc., and hence very likely to include text perturbations. Next, for each phonetic level $\mathbf{k}{\leq}K$, we curate different hash tables $\{H\}_0^K$ that maps a unique SOUNDEX++ code **c** to a set of its matching unique *case-sensitive* tokens that share the same encoding **c** as follows:

$$H_\mathbf{k} : \mathbf{c} \mapsto \{w_j | S(w_i, k) = S(w_j, k) = \mathbf{c} \\ \forall w_i, w_j \in \mathcal{D}, w_i \neq w_j\}, \quad (1)$$

where $S(w, \mathbf{k})$ returns the SOUNDEX++ code of token w at phonetic level **k**, K is the largest phonetic level we want to encode. With $\{H\}_0^K$, **k** and **d**, we can now search for the set of perturbations $G_\mathbf{k}^\mathbf{d}(w^*)$ of a specific target token w^* as follows:

$$G_\mathbf{k}^\mathbf{d}(w^*) {\leftarrow} \{w_j | w_j {\in} H_\mathbf{k}[S(w^*, k)], \\ \mathrm{Lev}(w^*, w_j) {\leq} \mathbf{d}\} \quad (2)$$

where $\mathrm{Lev}(w^*, w_j)$ returns the Levenshtein distance between w^* and w^j. Noticeably, we only extract $\{H\}_0^K$ **once** from $\mathcal{D}$ via Eq. (1), then we can use Eq. (2) to retrieve all perturbations for a given word during deployment. We name this method of mining and retrieving human-written text perturbations in the wild as **ANTHRO**, aka *human-like* perturbations:

$$\text{ANTHRO} : w*, \mathbf{k}, \mathbf{d}, \{H\}_0^K \quad \longmapsto \quad G_\mathbf{k}^\mathbf{d}(w^*) \quad (3)$$

Algorithm 1 ANTHRO Attack Algorithm

```
1:  **Input:** $\{H\}_0^K$, **k**, **d**
2:  **Input:** target classifier $f$, original sentence $x$
3:  **Output:** perturbed sentence $x^*$
4:  *Initialize:* $x^* \leftarrow x$
5:  **for** word $x_i$ **in** $x$ **do**:   $s_i \leftarrow \mathrm{Score}(x_i, f)$
6:  $\mathcal{W}_{\mathrm{order}} \leftarrow \mathrm{Sort}(x_1, x_2, ..x_m)$ according to $s_i$
7:  **for** $x_i$ in $\mathcal{W}_{\mathrm{order}}$ **do**:
8:      $\mathcal{P} \leftarrow \mathrm{ANTHRO}(x_i, \mathbf{k}, \mathbf{d}, \{H\}_0^K)$ // Eq.(3)
9:      $x^* \leftarrow$ replace $x_i \in x$ with the best $w \in \mathcal{P}$
10:     **if** $f(x^*) \neq f(x)$ **then** return $x^*$
11: **return** None
```

ANTHRO Attack. To utilize ANTHRO for adversarial attack on model $f(x)$, we propose the ANTHRO attack algorithm (Alg. 1). We use the same iterative mechanism (Ln.9–13) that is common among other black-box attacks. This process replaces the most vulnerable word in sentence x, which is evaluated with the support of $\mathrm{Score})(\cdot)$ function (Ln. 5), with the perturbation that best drops the prediction probability $f(x)$ on the correct label. Unlike the other methods, ANTHRO inclusively draws from perturbations extracted from human-written texts captured in $\{\mathcal{H}\}_0^K$ (Ln. 10). We adopt the $\mathrm{Score}(\cdot)$ from *TextBugger*.

4 Evaluation

We evaluate ANTHRO by: (1) attack performance, (2) semantic preservation, and (3) human-likeness–i.e., how likely an attack message is spotted as machine-generated by human examiners.

4.1 Attack Performance

Setup. We use BERT (*case-insensitive*) (Jin et al., 2019) and RoBERTa (*case-sensitive*) (Liu et al., 2019) as target classifiers to attack. We evaluate on three public tasks, namely detecting toxic comments ((TC) dataset, Kaggle 2018), hate speech ((HS) dataset (Davidson et al.)), and online cyberbullying texts ((CB) dataset (Wulczyn et al., 2017a)). We split each dataset to *train*, *validation* and *test* set with the 8:1:1 ratio. Then, we use the train set to fine-tune BERT and RoBERTa with a maximum of 3 epochs and select the best checkpoint using the validation set. BERT and RoBERTa achieve around 0.85–0.97 in F1 score on the test sets (Table A.2 in Appendix). We evaluate with targeted attack (change positive→negative label) since it is more practical. We randomly sample 200 examples from each test set and use them as initial sentences to attack. We repeat the process 3 times with unique random seeds and report the results. We use the *attack success rate (Atk%)* metric–i.e., the number of examples whose labels are flipped by an attacker over the total number of texts that are correctly predicted pre-attack. We use the 3rd party open-source *OpenAttack* (Zeng et al., 2021) framework to run all evaluations.

Baselines. We compare ANTHRO with three baselines, namely *TextBugger* (Li et al., 2018), *VIPER* (Eger et al., 2019) and *DeepWordBug* (Gao et al., 2018). These attackers utilize different character-based manipulations to craft their adversarial texts as described in Sec. 1. From the analysis in Sec. 3.1 and Figure 2, we set $\mathbf{k} \leftarrow 1$ and $\mathbf{d} \leftarrow 1$ for ANTHRO to achieve a balanced trade-off between precision and recall on the SMS property. We examine all attackers under several combinations of different normalization layers. They are (1) *Accents normalization* (A) and (2) *Homoglyph normalization* [1] (H), which converts non-English accents and homoglyphs to their corresponding ascii characters, (3) *Perturbation normalization* (P), which normalizes potential character-based perturbations using the SOTA misspelling correction model *Neuspell* (Jayanthi et al., 2020). These normalizers are selected as counteracts against the perturbation strategies employed by *VIPER* (uses non-English accents), *DeepWordBug* (uses homoglyphs) and *TextBugger*, ANTHRO (based on misspelling and typos), respectively.

Results. Overall, both ANTHRO and *TextBugger* perform the best. Being case-sensitive, ANTHRO performs significantly better on RoBERTa and is competitive on BERT when compared to *TextBugger* (Table 4). This happens because RoBERTa is case-sensitive (unlike the *base-uncased-bert* BERT model we used) and only ANTHRO is case-sensitive out of all attack baselines. For example, the perturbation “democrats"→“democRATs" is considered as a perturbation for RoBERTa but not for other case-insensitive models. This gives ANTHRO an advantage in practice because many popular commercial API services (e.g., the popular *Perspective API*, the sentiment analysis and text categorization API from Google) are case-sensitive–i.e., “democrats"≠“democRATs". (See more at Table 8).

[1] https://github.com/codebox/homoglyph

Attacker	Normalizer	BERT (*case-insensitive*)			RoBERTa (*case-sensitive*)		
		TC	HS	CB	TC	HS	CB
TextBugger	-	**0.76±0.02**	**0.94±0.01**	**0.78±0.03**	0.77±0.06	0.87±0.01	0.72±0.01
DeepWordBug	-	0.56±0.04	0.68±0.01	0.50±0.02	0.52±0.01	0.42±0.04	0.38±0.04
VIPER	-	0.08±0.03	0.01±0.01	0.13±0.02	**1.00±0.00**	**1.00±0.00**	**0.99±0.01**
ANTHRO	-	0.72±0.02	0.82±0.01	0.71±0.02	0.84±0.00	0.93±0.01	0.78±0.01
TextBugger	A	-	-	-	0.72±0.02	0.92±0.00	0.74±0.02
DeepWordBug	A	-	-	-	0.43±0.02	0.59±0.03	0.43±0.01
VIPER	A	-	-	-	0.09±0.01	0.05±0.01	0.17±0.02
ANTHRO	A	-	-	-	**0.77±0.02**	**0.94±0.02**	**0.84±0.02**
TextBugger	A+H	**0.78±0.03**	**0.85±0.00**	**0.79±0.00**	0.74±0.02	0.93±0.01	0.77±0.03
DeepWordBug	A+H	0.04±0.00	0.06±0.02	0.01±0.01	0.03±0.01	0.01±0.01	0.06±0.02
VIPER	A+H	0.07±0.00	0.01±0.01	0.10±0.00	0.13±0.02	0.07±0.01	0.17±0.01
ANTHRO	A+H	0.76±0.02	0.77±0.03	0.73±0.05	**0.82±0.02**	**0.97±0.00**	**0.82±0.02**
TextBugger	A+H+P	**0.73±0.02**	**0.64±0.06**	**0.70±0.04**	0.68±0.06	0.57±0.03	0.66±0.04
DeepWordBug	A+H+P	0.02±0.01	0.04±0.02	0.01±0.01	0.02±0.01	0.01±0.01	0.02±0.01
VIPER	A+H+P	0.12±0.01	0.04±0.01	0.17±0.03	0.11±0.02	0.05±0.01	0.18±0.01
ANTHRO	A+H+P	0.65±0.04	**0.64±0.01**	0.60±0.05	**0.80±0.02**	**0.91±0.03**	**0.82±0.02**

(-) BERT already has the accents normalization (A normalizer) by default, (Red): Poor performance (Atk%<0.15)

Table 4: Averaged attack success rate (Atk%↑) of different attack methods

VIPER achieves a near perfect score on RoBERTa, yet it is ineffective on BERT because RoBERTa uses the accent Ġ as a part of its byte-level BPE encoding (Liu et al., 2019) while BERT by default removes all such accents. Since *VIPER* exclusively utilizes accents, its attacks can be easily corrected by the *accents normalizer* (Table 4). Similarly, *DeepWordBug* perturbs texts with homoglyph characters, most of which can also be normalized using a 3rd party homoglyph detector (Table 4).

In contrast, even under all normalizers–i.e., A+H+P, *TextBugger* and ANTHRO still achieves 66.3% and 73.7% in Atk% on average across all evaluations. Although *Neuspell* (Jayanthi et al., 2020) drops *TextBugger*'s Atk% 14.7% across all runs, it can only reduce the Atk% of ANTHRO a mere 7.5% on average. This is because *TextBugger* and *Neuspell* or other dictionary-based typo correctors rely on fixed deductive rules–e.g., swapped, replaced by neighbor letters, for attack and defense. However, ANTHRO utilizes human-written perturbations which are greatly varied, hence less likely to be systematically detected. We further discuss the limitation of misspelling correctors such as NeuSpell in Sec. 7.

4.2 Human Evaluation

Since ANTHRO and *TextBugger* are the top two effective attacks, this section will focus on evaluating their ability in semantic preservation and human-likeness. Given an original sentence x and its adversarial text x^* generated by either one of the attacks, we design a human study to *directly compare* ANTHRO with *TextBugger*. Specifically, two alternative hypotheses for our validation are (1) $\mathcal{H}_{\text{Semantics}}$: x^* generated by ANTHRO preserves the original meanings of x *better* than that generated by *TextBugger* and (2) $\mathcal{H}_{\text{Human}}$: x^* generated by ANTHRO is *more likely* to be perceived as a human-written text (and not machine) than that generated by *TextBugger*.

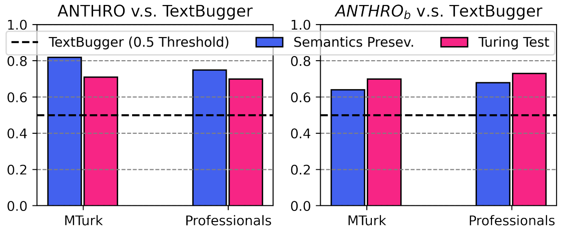


Figure 3: Semantic preservation and human-likeness

Human Study Design. We use the two attackers to generate adversarial texts targeting BERT model on 200 examples sampled from the TC dataset's test set. We then gather examples that are successfully attacked by both ANTHRO and *TextBugger*. Next, we present a pair of texts, one generated by ANTHRO and one by *TextBugger*, together with the original sentence to human subjects. We then ask them to select (1) which text *better* preserves the meaning of the original sentence (Figure B.2 in Appendix) and (2) which text is *more likely* to be written by human (Figure B.3

Attacker	Normalizer	BERT (*case-insensitive*)			RoBERTa (*case-sensitive*)		
		Toxic Comments	HateSpeech	Cyberbullying	Toxic Comments	HateSpeech	Cyberbullying
TextBugger	-	0.76±0.02	0.94±0.01	0.78±0.03	0.77±0.06	0.87±0.01	0.72±0.01
ANTHRO$_\beta$	-	**0.82±0.01**	**0.97±0.01**	**0.88±0.04**	**0.91±0.02**	**0.97±0.01**	**0.89±0.02**
TextBugger	A+H+P	0.73±0.02	0.64±0.06	0.70±0.04	0.68±0.06	0.57±0.03	0.66±0.04
ANTHRO$_\beta$	A+H+P	**0.85±0.04**	**0.79±0.02**	**0.84±0.03**	**0.88±0.04**	**0.93±0.01**	**0.91±0.01**

Table 5: Averaged attack success rate (Atk%↑) of ANTHRO$_\beta$ and *TextBugger*

Reason	Favorable For ANTHRO	Unfavorable For TextBugger
Genuine Typos	stuupid, but, Faoggt	sutpid, burt, Foggat
Intelligible	faiilure	faioure
Sound Preserv.	shytty, crp	shtty, crsp
Meaning Preserv.	ga-y, ashole, dummb	bay, alshose, dub
High Search Results	sodmized, kiills	Smdooized, klils
Personal Exposure	ign0rant, gaarbage	ignorajt, garage
Word Selection	morons→mor0ns	edited→ewited

Table 6: Top reasons in favoring ANTHRO's perturbations as more likely to be written by human.

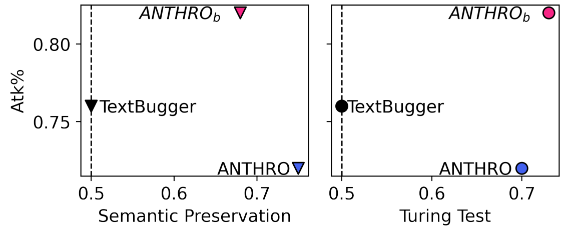


Figure 4: Trade-off among evaluation metrics

in Appendix). To reduce noise and bias, we also provide a *"Cannot decide"* option when quality of both texts are equally good or bad, and present the two questions in two separate tasks. Since the definition of semantic preservation can be subjective, we recruit human subjects as both (1) Amazon Mechanical Turk (MTurk) workers and (2) professional data annotators at a company with extended experience in annotating texts in domain such as toxic and hate speech. Our human subject study with MTurk workers was IRB-approved. We refer the readers to Sec. B.3 (Appendix) for more details on MTurks and study designs.

Quantitative Results. It is statistically significant (*p-value*$\leq$0.05) to reject the null hypotheses of both $\mathcal{H}_{\text{Semantics}}$ and $\mathcal{H}_{\text{Human}}$ (Table A.3). Overall, adversarial texts generated by perturbations mined in the wild are much better at preserving the original semantics and also at resembling human-written texts than those generated by *TextBugger* (Figure 3, Left).

Qualitative Analysis. Table 6 summarizes the top reasons why they favor ANTHRO over *TextBugger* in terms of human-likeness. ANTHRO's perturbations are perceived similar to genuine typos and more intelligible. They also better preserve both meanings and sounds. Moreover, some annotators also rely on personal exposure on Reddit, YouTube comments, or the frequency of word use via the search function on Reddit to decide if a word-choice is human-written.

5 ANTHRO$_\beta$ Attack

ANTHRO$_\beta$. We examine if perturbations inductively extracted from the wild help improve the deductive *TextBugger* attack. Hence, we introduce ANTHRO$_\beta$, which considers the perturbation candidates from both ANTHRO and *TextBugger* in Ln. 10 of Alg. 1. Alg. 1 still selects the perturbation that best flip the target model's prediction.

Attack Performance. Even though ANTHRO comes second after *TextBugger* when attacking BERT model, Table 5 shows that when combined with *TextBugger*–i.e., ANTHRO$_\beta$, it consistently achieves superior performance with an average of 82.7% and 90.7% in Atk% on BERT and RoBERTa even under all normalizers (A+H+P).

Semantic Preservation and Human-Likeness. ANTHRO$_\beta$ improves *TextBugger*'s Atk%, semantic preservation and human-likeness score with an increase of over 8%, 32% and 42% (from 0.5 threshold) on average (Table 5, 3, Right), respectively. The presence of only a few human-like perturbations generated by ANTHRO is sufficient to signal whether or not the whole sentence is written by humans, while only one unreasonable perturbation generated by *TextBugger* can adversely affect its meaning. This explains the performance drop in terms of semantic preservation but not in human-likeness when indirectly comparing ANTHRO$_\beta$ with ANTHRO. Overall, ANTHRO$_\beta$ also has the best trade-off between Atk% and hu-

Model	ANTHRO			ANTHRO$_\beta$		
	TC↓	HS↓	CB↓	TC↓	HS↓	CB↓
BERT	0.72	0.82	0.71	0.82	0.97	0.88
BERT+A+H+P	0.65	0.65	0.60	0.85	0.79	0.84
ADV.TRAIN	0.41	0.30	0.35	**0.72**	**0.72**	**0.67**
SOUNDCNN	**0.14**	**0.02**	**0.15**	0.86	0.84	0.92

Table 7: Averaged Atk% of ANTHRO and ANTHRO$_\beta$ against different defense models.

man evaluation–i.e., positioning at top right corners in Figure 4, with a noticeable superior Atk%.

6 Defend ANTHRO, ANTHRO$_\beta$ Attack

We suggest two countermeasures against ANTHRO attack. They are **(i) Sound-Invariant Model (SOUNDCNN):** When the defender do *not* have access to $\{\mathcal{H}\}_0^K$ learned by the attacker, the defender trains a generic model that encodes not the spellings but the phonetic features of a text for prediction. Here we train a CNN model (Kim, 2014) on top of a embeddings layer for discrete SOUNDEX++ encodings of each token in a sentence; **(ii) Adversarial Training (ADV.TRAIN):** To overcome the lack of access to $\{\mathcal{H}\}_0^K$, the defender extracts his/her perturbations in the wild from a separate corpus $\mathcal{D}^*$ where $\mathcal{D}^* \cap \mathcal{D} = \emptyset$ and uses them to augment the training examples–i.e., via self-attack with ratio 1:1, to fine-tune a more robust BERT model. We use $\mathcal{D}^*$ as a corpus of 34M general comments from online news. We compare the two defenses against BERT and BERT combined with 3 layers of normalization A+H+P. BERT is selected as it is better than RoBERTa at defending against ANTHRO (Table 4).

Results. Table 7 shows that both SOUNDCNN and ADV.TRAIN are robust against ANTHRO attack, while ADV.TRAIN performs best when defending ANTHRO$_\beta$. Since SOUNDCNN is strictly based on phonetic features, it is vulnerable against ANTHRO$_\beta$ whenever *TextBugger*'s perturbations are selected. Table 7 also underscores that ANTHRO$_\beta$ is a strong and practical attack, defense against which is thus an important future direction.

7 Discussion and Analysis

Evaluation with *Perspective API*. We evaluate if ANTHRO and ANTHRO$_\beta$ can successfully attack the popular *Perspective API* [2], which has been adopted in various publishers–e.g., NYTimes, and platforms–e.g., Disqus, Reddit, to detect toxicity. We evaluate on 200 toxic texts randomly sampled from the TC dataset. Figure 5 (Left) shows that the API provides superior performance compared to a self fine-tuned BERT classifier, yet its precision deteriorates quickly from 0.95 to only 0.9 and 0.82 when 25%–50% of a sentence are randomly perturbed using human-written perturbations. However, the ADV.TRAIN (Sec. 6) model achieves fairly consistent precision in the same setting. This shows that ANTHRO is not only a powerful and realistic attack, but also can help develop more robust text classifiers in practice. The API is also vulnerable against both direct (Alg. 1) and transfer ANTHRO attacks through an intermediate BERT classifier, with its precision dropped to only 0.12 when evaluated against ANTHRO$_\beta$.

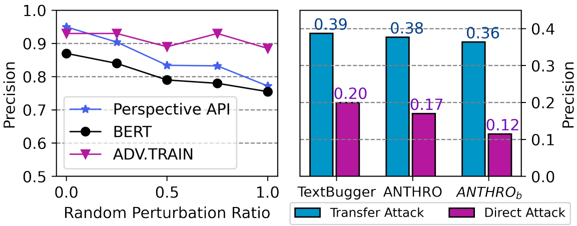


Figure 5: (Left) Precision on human-written perturbed texts synthesized by ANTHRO and (Right) Robustness evaluation of *Perspective API* under different attacks

Task	Sentiment Analysis	Categorization
ANTHRO	0.80	0.93
ANTHRO$_\beta$	0.86	1.00

Table 8: Averaged Atk% of ANTHRO and ANTHRO$_\beta$ in fooling Google Cloud[3]'s sentiment analysis API and text categorization API.

Generalization beyond Offensive Texts. Although ANTHRO extracts perturbations from abusive data, the majority of them are non-abusive texts. Thus, ANTHRO learns perturbations for non-abusive English words–e.g., hilarious->Hi-Larious, shot->sht. We also make no assumption on the task domains that ANTHRO can attack. Evidently, ANTHRO and ANTHRO$_\beta$ achieves 80%, 86% Atk% and 90%, 100% Atk% on fooling the sentiment analysis and text categorization API from Google Cloud (Table 8)

Computational Complexity. The **one-time** extraction of $\{\mathcal{H}\}_0^K$ via Eq. (1) has $\mathcal{O}(|\mathcal{D}|L)$

[2] https://www.perspectiveapi.com/

where $|\mathcal{D}|$, L is the # of tokens and the length of longest token in $\mathcal{D}$ (hash-map operations cost $\mathcal{O}(1)$). Given a word w and $\mathbf{k}, \mathbf{d}$, ANTHRO retrieves a list of perturbation candidates via Eq. (2) with $\mathcal{O}(|w|max(\mathcal{H}_k))$ where $|w|$ is the length of w and $max(\mathcal{H}_k)$ is the size of the largest set of tokens sharing the same SOUNDEX++ encoding in $\mathcal{H}_k$. Since $max(\mathcal{H}_k)$ is constant, the upper-bound then becomes $\mathcal{O}(|w|)$.

Limitation of Misspelling Correctors. Similar to other spell-checkers such as *pyspellchecker* and *symspell*, the SOTA NeuSpell depends on a fixed dictionary of common misspellings, or synthetic misspellings generated by random permutation of characters (Jayanthi et al., 2020). These checkers often assume perturbations are within an edit-distance threshold from the original words. This makes them exclusive since one can easily generate new perturbations by repeating a specific character–e.g., “porn"→“pooorn". Also, due to the iterative attack mechanism (Alg. 1) where each token in a sentence is replaced by many candidates until the correct label’s prediction probability drops, ANTHRO only needs a single good perturbation that is not detected by NeuSpell for a successful replacement. Thus, by formulating perturbations by not only their spellings but also their sounds, ANTHRO is able to mine perturbations that can circumvent NeuSpell.

Limitation of ANTHRO. The perturbation candidate retrieval operation (Eq. (2)) has a higher computational complexity than that of other methods–i.e., $\mathcal{O}(|w|)$ v.s. $\mathcal{O}(1)$ where $|w|$ is the length of an input token w (Please refer to Sec. 7 in the Appendix for detailed computational complexity). This can prolong the running time, especially when attacking long documents. However, we can overcome this by storing all the perturbations (given $\mathbf{k}, \mathbf{d}$) of the top frequently used offensive and non-offensive English words. We can then expect the operation to have an average complexity close to $\mathcal{O}(1)$. The current SOUNDEX++ algorithm is designed for English texts and might not be applicable in other languages. Thus, we plan to extend ANTHRO to a multilingual setting.

8 Conclusion

We propose ANTHRO, a character-based attack algorithm that extracts human-written perturbations in the wild and then utilizes them for adversarial text generation. Our approach yields the best trade-off between attack performance, semantic preservation and stealthiness under both empirical experiments and human studies. A BERT classifier trained with examples augmented by ANTHRO can also better understand human-written texts.

Broad Impact

To the best of our knowledge, ANTHRO is the first work that extracts noisy human-written texts, or text perturbations, online. We further iterate what reviewer pvcD has observed: ANTHRO moves “away from deductively-derived attacks to data-driven inspired attacks". This novel direction is beneficial not only to the adversarial NLP community but also in other NLP tasks that require the understanding of realistic noisy user-generated texts online. Specifically, Sec. 6 and Figure 5 shows that our work enables the training of a BERT model that can understand noisy human-written texts better than the popular *Perspective API*. By extending this to other NLP tasks such as QA and NLI, our work hopes to enable current NLP software to perform well in real life settings, especially on social platforms where user-generated texts are not always in perfect English. Our work also opens a new direction in the use of languages online and how netizens utilize different forms of perturbations for avoiding censorship in this new age of AI.

Ethical Consideration

Similar to previous works in adversarial NLP literature, there are risks that our proposed approach may be unintentionally utilized by malicious actors to attack textual ML systems. To mitigate this, we will not publicly release the full perturbation dictionary that we have extracted and reported in the paper. Instead, we will provide access to our private API on a case-by-case basis with proper security measures. Moreover, we also suggest and discuss two potential approaches that can defend against our proposed attacks (Sec. 6). We believe that the benefits of our work overweight its potential risks. All public secondary datasets used in this paper were either open-sourced or released by the original authors.

Acknowledgement

This research was supported in part by NSF awards #1820609, #1915801, and #2114824.

References

Arash Aryani, Erin S Isbilen, and Morten H Christiansen. 2020. Affective arousal links sound to meaning. *Psychological science*, 31(8):978–986.

Damián E Blasi, Søren Wichmann, Harald Hammarström, Peter F Stadler, and Morten H Christiansen. 2016. Sound–meaning association biases evidenced across thousands of languages. *Proceedings of the National Academy of Sciences*, 113(39):10818–10823.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *EMNLP'18, Demo*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *ICWSM'17*.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. In *NAACL'19*, pages 1634–1647, Minneapolis, Minnesota.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *SPW'18*. IEEE.

Raul Gomez, Jaume Gibert, Lluis Gomez, and Dimosthenis Karatzas. 2020. Exploring hate speech detection in multimodal publications. In *WACV'20*, pages 1470–1478.

Philip B Gough, JF Kavanagh, and IG Mattingly. 1972. One second of reading. *Cambridge: MIT Press*, pages 331–358.

Debra Jared and Mark S Seidenberg. 1991. Does word identification proceed from spelling to sound to meaning? *Journal of Experimental Psychology: General*, 120(4):358.

Sai Muralidhar Jayanthi, Danish Pruthi, and Graham Neubig. 2020. NeuSpell: A neural spelling correction toolkit. In *EMNLP'20, Demo*, Online.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP'14*, Doha, Qatar.

Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. All-in-one: Multi-task learning for rumour verification. In *ACL'18*, Santa Fe, New Mexico, USA. ACL.

Wolfgang Köhler. 1967. Gestalt psychology. *Psychologische Forschung*, 31(1):XVIII–XXX.

Thai Le, Noseong Park, and Dongwon Lee. 2021. A sweet rabbit hole by darcy: Using honeypots to detect universal triggers adversarial attacks. In *ACL'21*.

Thai Le, Suhang Wang, and Dongwon Lee. 2020. Malcom: Generating malicious comments to attack neural fake news detection models. In *ICDM'20*. IEEE.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. TextBugger: Generating Adversarial Text Against Real-world Applications. *NDSS'18*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *EuroS&P'16*, pages 372–387. IEEE.

Danish Pruthi, Bhuwan Dhingra, and Zachary C Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *ACL'19*.

Charles Stephenson. 1980. The methodology of historical census record linkage: A user's guide to the soundex. *Journal of Family History*, 5(1):112–115.

Caroline Tagg. 2011. Wot did he say 01" could u not c him 4 dust?: Written and spoken creativity in text messaging. *Transforming literacies and language: Multimodality and literacy in the new media age*, 223.

Guy C Van Orden. 1987. A rows is a rose: Spelling, sound, and reading. *Memory & cognition*, 15(3):181–198.

Wenqi Wang, Lina Wang, Run Wang, Zhibo Wang, and Aoshuang Ye. 2019. Towards a robust deep neural network in texts: A survey. *arXiv preprint arXiv:1902.07285*.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017a. Ex machina: Personal attacks seen at scale. In *WWW'17*, pages 1391–1399.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017b. Wikipedia talk labels: Personal attacks.

Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021. Openattack: An open-source textual adversarial attack toolkit. In *ACL'21, Demo*, pages 363–371.

Dataset	#Texts	#Tokens
List of Bad Words [4]	1.9K	1.9K
Rumours (Twitter) (Kochkina et al., 2018)	99K	159K
Hate Memes (Twitter) (Gomez et al., 2020)	150K	328K
Personal Atks (Wiki.) (Wulczyn et al., 2017b)	116K	454K
Toxic Comments (Wiki.) (Kaggle, 2019)	2M	1.6M
Malignant Texts (Reddit) (Kaggle, 2021)[5]	313K	857K
Hateful Comments (Reddit) (Kaggle, 2021)[6]	1.7M	1M
Sensitive Query (Search Engine, Private)	1.2M	314K
Hateful Comments (Online News, Private)	12.7M	7M
Total texts used to extract ANTHRO	**18.3M**	-

Table A.1: Real-life datasets that are used to extract adversarial texts in the wild, number of total examples (#Texts) and unique tokens (#Tokens) (case-insensitive)

A Supplementary Materials

A.1 Additional Results and Figures

Below are list of supplementary materials:

- Table A.1: list of datasets we used to curate the corpus $\mathcal{D}$, from which human-written perturbations are extracted (Sec. 3.1). All the datasets are publicly available, except from the two private datasets *Sensitive Query* and *Hateful Comments*.
- Table A.2: list of datasets we used to evaluate the attack performance of all attackers (Sec. 4.1) and the prediction performance of BERT and RoBERTa on the respective test sets. All datasets are publicly available.
- Table A.3: Statistical analysis of the human study results (Sec. 4.2).
- Figure B.1: Word-cloud of extracted human-written perturbations by ANTHRO for some of popular English words.
- Figure B.2, B.3: Interfaces of the human study described in Sec. 4.2.

A.2 Infrastructure and Software

B Implementation Details

B.1 Attackers

We evaluate all the attack baselines using the open-source *OpenAttack* framework (Zeng et al., 2021). We keep all the default parameters for all the attack methods.

Dataset	#Total	BERT	RoBERTa
CB (Wulczyn et al., 2017a)	449K	0.84	0.84
TC (Kaggle, 2018)	160K	0.85	0.85
HS (Davidson et al.)	25K	0.91	0.97

Table A.2: Evaluation datasets Cyberbullying (CB), Toxic Comments (TC) and Hate Speech (HS) and prediction performance in F1 score on their test sets of BERT and RoBERTa.

Alternative Hypothesis	Mean	t-stats	p-value	df
—— AMT Workers as Subjects ——				
$\mathcal{H}_{\text{Semantics}}$: ANTHRO > TB	0.82	5.66	4.1e-7**	48
$\mathcal{H}_{\text{Semantics}}$: ANTHRO$_\beta$ > TB	0.64	1.95	2.9e-2*	46
$\mathcal{H}_{\text{Human}}$: ANTHRO > TB	0.71	3.14	1.5e-3**	47
$\mathcal{H}_{\text{Human}}$: ANTHRO$_\beta$ > TB	0.70	3.00	2.2e-3**	46
—— Professional Annotators as Subjects ——				
$\mathcal{H}_{\text{Semantics}}$: ANTHRO > TB	0.75	3.79	2.4e-4**	44
$\mathcal{H}_{\text{Semantics}}$: ANTHRO$_\beta$ > TB	0.68	2.49	8.6e-3**	41
$\mathcal{H}_{\text{Human}}$: ANTHRO > TB	0.70	3.06	1.82e-3**	50
$\mathcal{H}_{\text{Human}}$: ANTHRO$_\beta$ > TB	0.73	3.53	4.6e-4**	48
Statistical significant **(p-value≤0.01) *(p-value≤0.05)				

Table A.3: It is *statistically significant* (p-value≤0.01) that adversarial texts generated by ANTHRO are better than those generated by TextBugger (TB) at both preserving the semantics of the original sentences ($\mathcal{H}_{\text{Semantics}}$)) and at being perceived as human-written texts ($\mathcal{H}_{\text{Human}}$).

B.2 Defenders

For the (1) *Accents normalization*, we adopt the accents removal code from the *Hugging Face* repository [7]. For (2) *Homoglyph normalization*, we adopt a 3rd party python *Homoglyph* library[8]. For (3) *Perturbation normalization*, we use the state-of-the-art misspelling-based perturbation correction *Neuspell* model (Jayanthi et al., 2020) [9]. For *Perspective API*, we directly use the publicly available API provided by Jigsaw and Google [10].

B.3 Details of Human Study and Experiment Controls

To ensure a high quality response from MTurks, we require a minimum attentions span of 30 seconds for each question. We recruit MTurk workers who are 18 years or older residing in North America. MTurk workers are recruited using the following qualifications provided by AMT, namely (1) recognized as "master" workers by AMT system,

[7] https://huggingface.co

[8] https://github.com/codebox/homoglyph

[9] https://github.com/neuspell/neuspell

[10] https://www.perspectiveapi.com/

(2) have done at least 5K HITs and (3) have historical HITs approval rate of at least 98%. These qualifications are also more conservative than previous human studies we found in previous literature. We pay each worker on average around $10 an hour or higher (federal minimum wage was $7.25 in 2021 when we carried out our study). To limit abusive behaviors, we impose a minimum attention span of 30 seconds for the workers to complete each task.

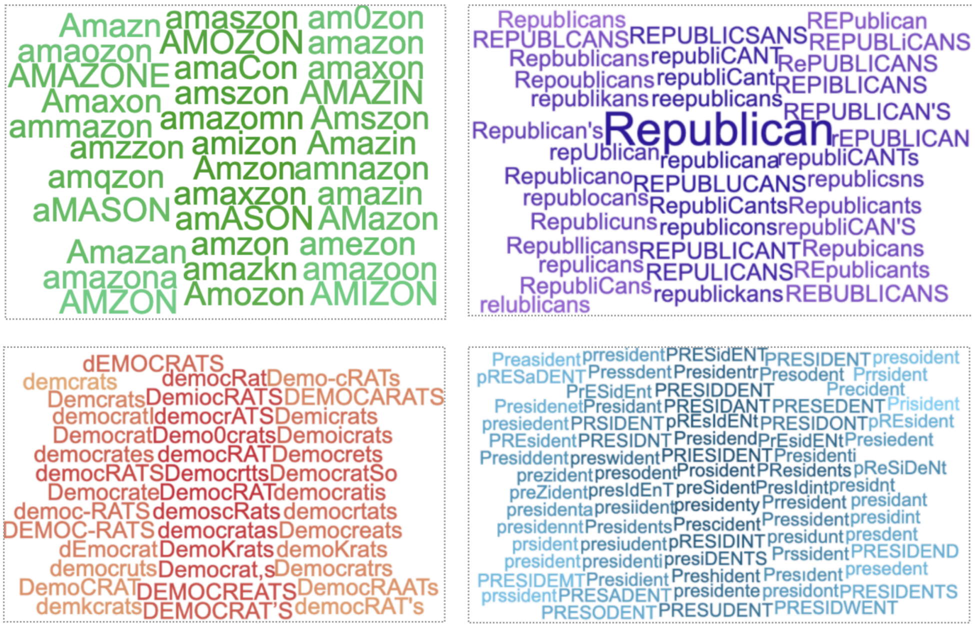
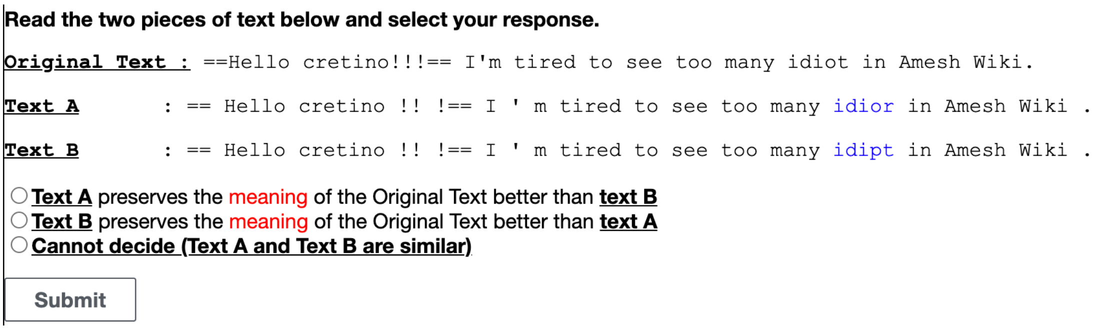

Amazn amaszon am0zon
amaozon AMOZON amazon
AMAZONE amaCon amaxon
Amaxon amszon AMAZIN
ammazon amazomn Amszon
amzzon amizon Amazin
amqzon Amzonamnazon
aMASON amaxzon amazin
amASON AMazon
Amazan amzon amezon
amazona amazkn amazoon
AMZON Amozon AMIZON
Republicans REPublican
REPUBLCANS REPUBLICSANS REPUBLiCANS
Repbublicans republiCANT RePUBLICANS
Repoublicans republiCant REPIBLICANS
republikans reepublicans REPUBLICAN'S
Republican's Republican rEPUBLICAN
repUblican republicana republiCANTs
Republicano REPUBLUCANS republicsns
republocans RepubliCants Republicants
Republicuns republicons republiCAN'S
Republlicans REPUBLICANT Repubicans
repulicans REPULICANS REpublicants
RepubliCans republickans REBUBLICANS
relublicans
dEMOCRATS
demcrats democRat Demo-cRATs
Demcrats DemiocRATS DEMOCARATS
democratl democrATS Demicrats
Democrat Demo0crats Demoicrats
democrates democRAT Democrets
democRATS Democrtts DemocratSo
Democrate DemocRAT democratis
democ-RATS demoscRats democrtats
DEMOC-RATS democratas Democreats
dEmocrat DemoKrats demoKrats
democruts Democrat,s Democratrs
DemoCRAT DEMOCREATS DemocRAATs
demkcrats DEMOCRAT'S democRAT's
Preasident prresident PRESidENT PRESIDENT presoident
pRESaDENT Pressdent Presidentr Presodent Prrsident
PrESidEnt PRESIDDENT Precident
Presidenet Presidant PRESIDANT PRESEDENT Prisident
presiedent PRSIDENT pREsIdENt PRESIDONT pREsident
PREsident PRESIDNT Presidend PrEsidENt Presiedent
Presiddent preswident PRIESIDENT Presidenti
prezident presodent Prosident PResidents pReSiDeNt
preZident presIdEnT preSident PresIdint presidnt
presidenta presiident presidenty Prresident presidant
presidennt Presidents Prescident Pressident presidint
prsident presiudent pRESIDINT presidunt presdent
president presidenti presiDENTS Prssident PRESIDEND
PRESIDEMT Presidient Preshident Presıdent presedent
prssident PRESADENT presidente presidont PRESIDENTS
PRESODENT PRESUDENT PRESIDWENT

Figure B.1: Word-clouds of perturbations in the wild extracted by ANTHRO for the word “amazon”, “republicans”, “democrats” and “president”.



Read the two pieces of text below and select your response.

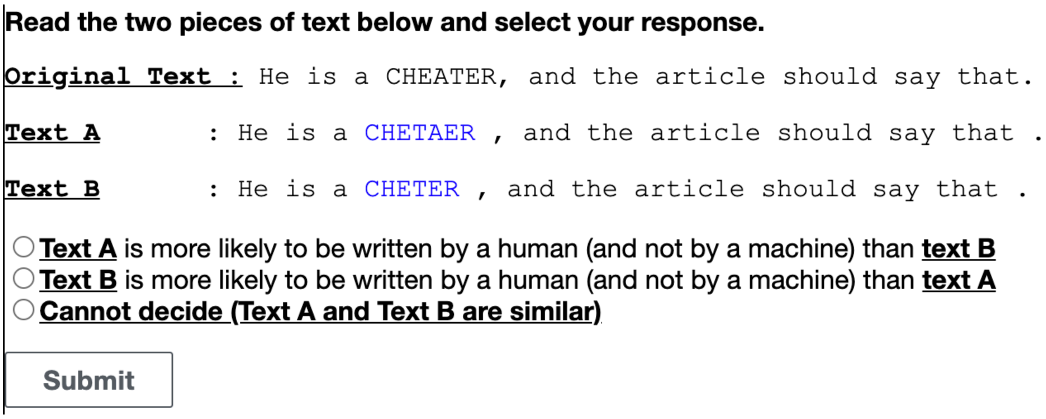
Original Text : `==Hello cretino!!!== I'm tired to see too many idiot in Amesh Wiki.`

Text A : `== Hello cretino !! !== I ' m tired to see too many idior in Amesh Wiki .`

Text B : `== Hello cretino !! !== I ' m tired to see too many idipt in Amesh Wiki .`

○ **Text A** preserves the meaning of the Original Text better than **text B**
○ **Text B** preserves the meaning of the Original Text better than **text A**
○ **Cannot decide (Text A and Text B are similar)**

Figure B.2: User-study design for semantic preservation comparison between ANTHRO, ANTHRO$_\beta$ v.s. *TextBugger*



Read the two pieces of text below and select your response.

Original Text : `He is a CHEATER, and the article should say that.`

Text A : `He is a CHETAER , and the article should say that .`

Text B : `He is a CHETER , and the article should say that .`

○ **Text A** is more likely to be written by a human (and not by a machine) than **text B**
○ **Text B** is more likely to be written by a human (and not by a machine) than **text A**
○ **Cannot decide (Text A and Text B are similar)**

Figure B.3: User-study design for human-likeness comparison between ANTHRO, ANTHRO$_\beta$ v.s. *TextBugger*