On the Robustness of Offensive Language Classifiers [‡]

Jonathan Rusert
University of Iowa
jonathan-rusert@uiowa.edu

Zubair Shafiq
University of California, Davis
zshafiq@ucdavis.edu

Padmini Srinivasan
University of Iowa
padmini-srinivasan@uiowa.edu

Abstract

Social media platforms are deploying machine learning based offensive language classification systems to combat hateful, racist, and other forms of offensive speech at scale. However, despite their real-world deployment, we do not yet comprehensively understand the extent to which offensive language classifiers are robust against adversarial attacks. Prior work in this space is limited to studying robustness of offensive language classifiers against primitive attacks such as misspellings and extraneous spaces. To address this gap, we systematically analyze the robustness of state-of-the-art offensive language classifiers against more crafty adversarial attacks that leverage greedy- and attention-based word selection and context-aware embeddings for word replacement. Our results on multiple datasets show that these crafty adversarial attacks can degrade the accuracy of offensive language classifiers by more than 50% while also being able to preserve the readability and meaning of the modified text.

1 Introduction

Online social media platforms are dealing with an unprecedented scale of offensive (e.g., hateful, threatening, profane, racist, and xenophobic) language (Twitter; Facebook; Reddit). Given the scale of the problem, online social media platforms now increasingly rely on machine learning based systems to proactively and automatically detect offensive language (Rosen, 2020; Gadde and Derella, 2020; Kastrenakes, 2019; Hutchinson, 2020). The research community is actively working to improve the quality of offensive language classification (Zampieri et al., 2020, 2019b; Liu et al., 2019; Nikolov and Radivchev, 2019; Mahata et al., 2019; Arango et al., 2020; Agrawal and Awekar, 2018; Fortuna and Nunes, 2018). A variety of offensive language classifiers ranging from traditional shallow models (SVM, Random Forest), deep learning models (CNN, LSTM, GRU), to transformer-based models (BERT, GPT-2) have been proposed in prior literature (Liu et al., 2019; Nikolov and Radivchev, 2019; Mahata et al., 2019). Amongst these approaches, BERT-based transformer models have achieved state-of-the-art performance while ensembles of deep learning models also generally perform well (Zampieri et al., 2019b, 2020).

It remains unclear whether the state-of-the-art offensive language classifiers are robust to adversarial attacks. While adversarial attacks are of broad interest in the ML/NLP community (Hsieh et al., 2019; Behjati et al., 2019), they are of particular interest for offensive language classification because malicious users can make subtle perturbations such that the offensive text is still intelligible to humans but evades detection by machine learning classifiers. Prior work on the robustness of text classification is limited to analyzing the impact on classifiers of primitive adversarial changes such as deliberate misspellings (Li et al., 2019), adding extraneous spaces (Gröndahl et al., 2018), or changing words with their synonyms (Jin et al., 2020; Ren et al., 2019; Li et al., 2020). However, the primitive attacks can be easily defended against—a spell checker can fix misspellings and a word segmenter can correctly identify word boundaries even with extra spaces (Rojas-Galeano, 2017; Li et al., 2019). Additionally, a normal synonym substitution will not theoretically hold for offensive language as less offensive language will be substituted and thus meaning will be lost. Crucially, we do not know how effective these text classifiers are against crafty adversarial attacks employing more advanced strategies for text modifications.

To address this gap, we analyze the robustness of offensive language classifiers against an adversary who uses a novel word embedding to identify

[*] This paper examines offensive language as a case study. The reader is cautioned that the paper contains unavoidable strong language given the nature of the research.

[†] Our code and data are available at: https://github.com/JonRusert/RobustnessOfOffensiveClassifiers

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7424 - 7438

word replacements and a surrogate offense classifier in a black-box setting to guide modifications. This embedding is purpose-built to evade offensive language classifiers by leveraging an *evasion collection* that comprises of evasive offensive text gathered from online social media. Using this embedding, the adversary modifies the offensive text while also being able to preserve text readability and semantics. We present a comprehensive evaluation of the state-of-the-art BERT and CNN/LSTM based offensive language classifiers, as well as an offensive lexicon and Google's Perspective API, on two datasets.

We summarize our key contributions below.

• We systematically study the ability of an adversary who uses a novel, crafty strategy to attack and bypass offensive language classifiers. The adversary first builds a new embedding from a special evasion collection, then uses it alongside a surrogate offensive language classifier deployed in black-box mode to launch the attack.

• We explore variations of our adversarial strategy. These include greedy versus attention based selection of text words to replace. These also include two different versions of embeddings for word substitutions.

• We evaluate robustness of state-of-the-art offensive language classifiers, as well as a real-world offensive language classification system on two datasets from Twitter and Reddit. Our results show that 50% of our attacks cause an accuracy drop of $\geq 24\%$ and 69% of attacks cause drops $\geq 20\%$ against classifiers across datasets.

Ethics Statement: We acknowledge that our research demonstrating attacks against offensive language classifiers could be used by bad agents. Our goal is to highlight the vulnerability within offensive language classifiers. We hope our work will inspire further research to improve their robustness against the presented and similar attacks.

2 Target Offensive Language Classifiers

2.1 Threat model

The adversary's goal is to modify his/her offensive post in such a manner as to evade detection by offensive language classifiers while simultaneously preserving semantics and readability for humans. To make suitable modifications, the adversary is assumed to have black-box access to a surrogate offensive language classifier that is different from the one used by the online social media platform. The adversary leverages feedback from this surrogate classifier to guide modifications using a novel approach that we propose. Our goal is to evaluate the extent to which the adversary can evade detection by an unknown offensive language classifier under this threat model.

2.2 Offensive Language Classifiers

We evaluate the following offensive language classifiers under our threat model.

1. **NULI** (Liu et al., 2019) is a BERT (Devlin et al., 2019) based system trained on offensive language. During preprocessing, emojis are converted into English phrases[1] and hashtags are segmented[2]. This was the top-ranked system in OffensEval (Zampieri et al., 2019b).

2. **Vradivchev** (Nikolov and Radivchev, 2019) is also a BERT based system trained on offensive language data. The preprocessing step includes removing symbols "@" and "#", tokenization and lowercasing, splitting hashtags, and removing stopwords. This was the second best system in OffensEval.

3. **MIDAS** (Mahata et al., 2019) is a voting ensemble of three deep learning systems: a CNN, a BLSTM, and a BLSTM fed into a Bidirectional Gated Recurrent Unit (BGRU). This was the top non-BERT system in OffensEval[3].

4. **Offensive Lexicon** (Wiegand et al., 2018) is a simple method that classifies a post as offensive if at least one word is in a lexicon of offensive words. We use their lexicon.

5. **Perspective API** (Perspective) by Google (Jigsaw) provides a toxicity model that classifies whether a post is "rude, disrespectful, or unreasonable." The production model uses a CNN trained with fine-tuned GloVe word embeddings and provides "toxicity" probability. We use 0.5 threshold to classify a post as offensive as in Pavlopoulos et al. (2019).

[1] https://github.com/carpedm20/emoji

[2] https://github.com/grantjenks/python-wordsegment

[3] We implemented NULI, vradivchev and MIDAS with parameters reported in the cited papers. Our accuracies were within 1% of the reported F1 score.

3 Attack Methods

This section describes our adversarial attack method as well as a recent visual adversarial attack (Eger et al., 2019) and a simpler attack (Gröndahl et al., 2018) for baseline comparison.

3.1 Proposed Attack

The adversary's attack involves selecting words to replace in the input text and deciding on suitable replacements.

Selection. There are several ways to approach word selection for replacement. Here we explore a greedy approach (Hsieh et al., 2019) and an approach using attention weights (Xu et al., 2018).

For the greedy approach, we first remove each word one at a time (retaining the rest in the text) and get the drop in classification probability for the text from the surrogate offensive classifier. Words are removed until the offensive label is flipped (according to the classifier). The removed words make up the full list of possible replacements. The adversary then selects the word that causes the largest drop for replacement. If replacing this word is insufficient to bypass the surrogate classifier then the word with the next largest drop is also selected for replacement and so on.

For the attention approach, we leverage a BLSTM with attention which is trained on the target classification task. Note that this BLSTM is different from the one found in MIDAS. To select words, we give the input text to the BLSTM and examine the attention weights estimated during classification. The adversary selects the word with the highest attention weight. If replacing this word is insufficient to bypass the surrogate classifier then the word with the next largest attention weight is also selected for replacement and so on.

The attention approach can potentially find replacements that greedy approach may not. Specifically, the greedy approach may miss instances where the combination of words cause offense rather than single words.

Replacement. Figure 1 depicts our framework for substituting the selected word with another word. First, a candidate list of 20 most similar words (closest vectors) is obtained from an embedding space. Next, we replace the selected word with its most similar word and check the modified text against the surrogate classifier. If the modified text is declared not offensive, then this word is chosen as the replacement. Otherwise, the process continues with the next most similar word. If the candidate list is exhausted without misclassification by the surrogate classifier, we choose the replacement word which causes the largest drop in classification probability.

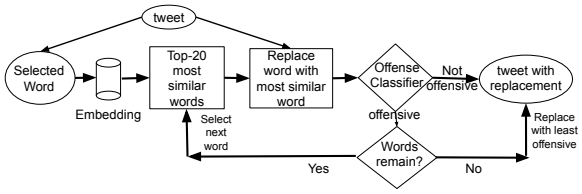

Figure 1: The design of our word replacement approach. First, select the word's 20 most similar words as candidates. Next, replace the word with the most similar candidate and check text against surrogate classifier. If this results in a not offensive classification, the process ends with this word as replacement. Otherwise, continue the process with the next most similar candidate. If no candidates remain, choose the one which causes the greatest drop in classification probability.

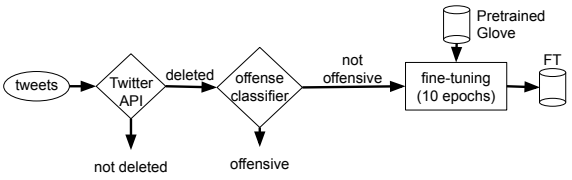

Figure 2: Our approach for creating FT . First, 13 million deleted tweets are identified via retrospective analysis using the Twitter API (Le et al., 2019; Thomas et al., 2011). Next, the tweets are checked against offensive language classifiers to remove those detected as offensive. Finally, for FT, fine-tune the pretrained (Pre) with the remaining tweets for 10 epochs.

Embeddings. The key idea here is to design a context-aware word embedding for crafty replacements. To this end, we first build a text collection of 13 million deleted tweets through retrospective analysis using the Twitter API (Thomas et al., 2011; Le et al., 2019). Next we filter out the tweets from this set that are labeled as offensive by any of the offensive language classifiers in Section 2.2.[4] The remaining set of 8.5 million deleted tweets contains offensive tweets that were likely flagged by users or human moderators.[5] We expect this set of deleted tweets to contain crafty substitutions and expressions that are likely to evade detection by state-of-the-art offensive language classifiers. We refer to this set of deleted tweets as the *evasion* collection and this is the data that the adversary uses to train word embeddings. We explore the following embeddings:

1. GloVe pretrained Twitter embedding (Pre):

[4]Note that Perspective did not participate in this tagging due to query limits of the API.

[5]It is possible that some of these tweets were deleted for other reasons that are unknown to us.

These are pretrained GloVe embeddings on 2 billion tweets. The vocabulary size of this model is 1,193,514 tokens. This represents a baseline off-the-shelf word embedding.

2. GloVe embedding fine-tuned with evasion collection (FT): We use the evasion collection to fine-tune the pretrained GloVe embeddings. Fine tuning is done over 10 epochs. The resulting vocabulary size is 1,312,106 tokens. Figure 2 illustrates this approach.

Insights into the embeddings. Our intuition of crafty substitutions being present in the evasion collection is backed up by examination of the embeddings. Using a set of offensive words as probes we find that on average the position of the first evasive word amongst the 20 most similar words in Pre is 11, while for FT this number is 3, implying that FT is more likely to offer an evasive replacement. We expand on these insights and analysis in Section 6. Furthermore, as fine-tuning the embeddings may introduce garbage words (non english, often meaningless words) as replacements, we add in a filter to the candidates when using the FT embeddings. This filter only allows candidates which have been used in tweets by 3 distinct authors in the *evasion* dataset. Finally, as checking every candidate can be time consuming and inefficient, we apply this filter only when we substitute text words that were not in the original *Pre* embeddings.

3.2 Other Attacks

VIPER. We implement a recent visual adversarial attack called VIPER (Eger et al., 2019) that aims to generate adversarial text for any classification task. VIPER (VIsual PERturber) replaces characters in the text with visually nearest neighbors determined from a visual embedding space. Each character present in the text is selected for replacement with a fixed probability p. VIPER strategically chooses replacements from non standard unicode characters assuming that systems rarely train outside the standard unicode space. As the main comparison, we choose their description-based character embedding space (DCES) in our experiments since it had the best tradeoff between attack success and readability. DCES represents characters by their unicode textual descriptions. The nearest neighbor substitute is the character whose description refers to the same letter in the same case. We also compare with their simpler, easy character embedding space (ECES), which contains only nearest neighbor for character replacement. We used VIPER with p = 0.1 and 0.4, the first for better readability and the second for better likelihood of attack success. Note that higher p values correspond to more changes in the text.

Grondahl. Gröndahl et al. (2018) explored rather simple attack methods such as modifying whitespace and misdirection by adding a misleading word. We implement several of their adversarial attacks. These are: adding a space after every character, removing all spaces between characters, adding the word 'love' to the input text, and finally removing all spaces then adding 'love.' This last attack strategy outperformed others in their evaluation.

4 Experimental Setup

4.1 Datasets

Offensive Language Identification Dataset (OLID). OLID was used in SemEval-6 2019: OffensEval, a shared task on classifying offensive language (Zampieri et al., 2019a). This collection is annotated by experienced annotators to ensure high quality. OLID contains 14,100 English tweets (text only): split into 13,240 (4,400 offensive, 8,840 non-offensive) training tweets and 860 (240 offensive, 620 non-offensive) test tweets.

Semi-Supervised Offensive Language Identification Dataset (SOLID). SOLID is an expansion of OLID used in SemEval 2020: OffensEval, which continued the task of classifying offensive language (Rosenthal et al., 2020). SOLID was constructed from tweets via semi-supervised manner using democratic co-training with OLID as a seed dataset. SOLID contains 9,000,000 tweets as an expansion for training, and 5,993 test tweets, (3,002 offensive, 2,991 non-offensive).

4.2 Evaluation Metrics

Drop in Accuracy:

$$\Delta = \text{Accuracy}_{\text{Original}} - \text{Accuracy}_{\text{Modified}},$$

where $\text{Accuracy}_{\text{Original}}$ is the classifier's accuracy on original text and $\text{Accuracy}_{\text{Modified}}$ is the classifier's accuracy on the modified text. Larger drops imply better evasion of offensive language classifiers by the adversary.

Readability and semantic preservation: We measure readability of the modified text and its semantic preservation through manual evaluation. More specifically, for readability, human reviewers are

asked to examine the modified text and rate it as one of: {'The text is easy to read', 'The text can be read with some difficulty', 'The text is hard to read'}. For semantic preservation, reviewers are given the original texts alongside the modified versions and are asked whether 'text B' (modified text) conveys the same meaning as 'text A' (original text). The choices are {'Yes, Text B conveys the same meaning as Text A', 'Text B conveys partially the meaning of Text A', 'No, Text B does not convey the same meaning as Text A'}.

4.3 Experiment Design

We use the OLID and SOLID test sets to assess the success of our attack strategies. Amongst the several offensive language classifiers considered in this work (see Section 2.2), we make one classifier available to the adversary as a surrogate black-box classifier to guide adversarial modification of each test tweet. Note that we do not use Lexicon as an internal classifier as it does not provide useful feedback (only returning 0 or 1 for positive class probabilities). We then evaluate the drop in classification accuracy (Δ) for each of the remaining classifiers.

5 Results

In this section, we first present the results of our proposed adversarial attack approach and then those of existing approaches from prior literature on the OLID dataset. Evaluation was also performed on the SOLID dataset and the results followed a similar trend. Full results for all attacks are located in the appendix.

5.1 Our adversarial attack

Table 1 presents the results on the OLID dataset. Rows specify the attack strategy. The first column identifies the surrogate offensive language classifier used by the adversary to guide modifications. The remaining columns specify the offensive language classifier whose robustness is being evaluated. Cell values are drops in accuracy after adversarial modification. Accuracy here refers to the percentage of offensive tweets correctly predicted as offensive. Classification accuracy for original text is given in the first row of the table. So for example, the final accuracy for NULI where the adversary uses GS-Pre and MIDAS is 44 (61-17). Blocks of rows labeled with prefix GS stand for results with greedy word selection strategy while AS stand for results with BLSTM-attention based word selection. Note that diagonal entries, where the surrogate classifier is the same as the one being tested for robustness are ignored because the adversary is expected to be quite successful under this condition. We indeed find that the accuracy drops close to 0% in these cases. Additionally, for the Lexicon based method, we find it does not perform as well as the other classifiers, thereby excluding it from the state-of-the-art (SOTA) category.

Offensive language classifiers are susceptible to our adversarial attacks. Table 1 shows that our adversarial attacks are quite successful against crafty offensive language classifiers. For OLID, classifiers see a drop of accuracy in the range of 11–46[6]. In fact, 50% of attacks cause a drop of ≥ 24 and 69% of attacks cause a drop of ≥ 20. This shows the vulnerability of offensive language classifiers and their vulnerability under our threat model.

Greedy select (GS) outperforms Attention select (AS) attacks. Greedy Select achieves higher average drops in accuracy across classifiers. For example, GS - FT achieves an average drop of 26 against NULI while AS - FT achieves only a drop of 17. This holds true for both replacement embeddings. Although lower, AS still achieves strong drops against vradivchev (average of 35). This indicates the strength of a greedy approach, however, attention selection may be more viable in a setting where the number of queries is limited.

***FT* embeddings and *Pre* embeddings see success against different systems.** Comparing to Pre embeddings, FT we see different leads in dropped accuracy depending on the classifier. FT see great success against NULI and vradivchev, while Pre see success against the other three. This indicates that the *evasion* collection can help add power, especially against popular (BERT-based) classifiers.

NULI and vradivchev, BERT based classifiers, are the most and least robust to attacks. Focusing on the GS - FT embedding, NULI has a mean drop in accuracy of 26 (range: 18 - 39), the lowest across SOTA offensive language classifiers. In contrast, vradivchev, performs the best with accuracy of 69 but is also the most vulnerable to our attack model with a mean drop in accuracy of 37

[6]Note: The BLSTM attention classifier used for attention based word selection could also be used as an internal classifier. However, since this strategy did not perform as well as SOTA classifiers so we do not include these results in the main analysis.

		Drop in Classification Accuracy					
		NULI	vradivchev	MIDAS	Perspective	Lexicon	Avg. Drop
No Attack Accuracy %		61	69	66	68	54	
	Surrogate Classifier						
GS - Pre	NULI	-	41	33	34	24	*33*
	vradivchev	28	-	33	28	22	28
	MIDAS	17	35	-	26	19	24
	Perspective	20	36	30	-	17	26
	Average Drop	22	**37**	**32**	**29**	**21**	
GS - FT	NULI	-	46	30	31	19	*32*
	vradivchev	39	-	30	26	18	28
	MIDAS	18	29	-	23	13	21
	Perspective	22	37	28	-	13	25
	Average Drop	**26**	**37**	29	27	16	
AS - Pre	NULI	-	36	19	19	15	22
	vradivchev	22	-	18	19	17	19
	MIDAS	13	34	-	20	15	21
	Perspective	17	37	23	-	16	*23*
	Average Drop	17	36	20	19	16	
AS - FT	NULI	-	39	18	17	15	22
	vradivchev	23	-	17	15	15	18
	MIDAS	11	27	-	17	12	17
	Perspective	17	40	21	-	16	*24*
	Average Drop	17	35	19	16	15	

Table 1: Robustness results on OLID with our attack model. Columns show accuracy drop. The approach is specified as *selection - replacement* where *selection = {Greedy Select (GS), Attention Select (AS)}* and *replacement = {*Pre*, *FT*}*. Note that the BLSTM used for AS can be used as an internal classifier but performed poorly so was not included. The adversarial, surrogate classifier is indicated in column 1. The first row presents baseline classification accuracies (%) before attacks. Therefore the resulting accuracies can be calculated by subtracting the drop from the original accuracy.

(range: 29 - 46), the highest drop of any offensive language classifier. This mean is 27 for Perspective and 29 for MIDAS. The stark difference between the two BERT systems' robustness most likely stems from the preprocessing step. BERT is a context-aware system. While NULI's preprocessing helps add context (e.g. converting emojis to text), vradivchev's hinders it. Specifically, vradivchev removes stop words. This could be a problem as removing this additional information causes the system to miss out on context during training. Then, as the attack is more likely to focus on changing non-stop words, vradivchev then loses both contextual information (via stop word removal) as well as offense indicating tokens (the main information it focused on during training).

NULI is the most effective surrogate classifier for the adversary while MIDAS the least effective. Again, focusing on GS-FT , NULI helps the adversary as the surrogate classifier the most by causing an average accuracy drop of 32 (range: 19 - 46), compared to vradivchev (avg: 28, range: 18 - 39), Perspective (avg: 25, range: 13 - 37), and MIDAS (avg: 21, range: 13 - 29). This again emphasizes BERT based methods' ability to understand context and use it effectively in attacks, also seen in previous research (Li et al., 2020).

Replication of Results. We replicate our results on a Reddit dataset in the appendix.

5.2 Other attacks

Grondahl. Table 2 shows the results when methods proposed by Gröndahl et al. (2018) are used to obfuscate. Note that this approach does not use a surrogate classifier. The simpler whitespace and 'love' word based attacks proposed by Gröndahl et al. (2018) have little to no effect on offensive language classifiers which contain a word segmentation pre-processing step. These classifiers include NULI (average drop: -3), vradivchev (average drop: 11), and Lexicon (average drop:0). MIDAS being ill equipped in this regard sees a drop of 64 when all spaces are removed and 'love' is added to the text. However, when we add a simple word segmentation step during pre-processing the attack loses effectiveness. For example, the "Remove space, Add 'love"' attack is reduced to a drop of 33 with this shielding pre-processing step, compared to 64 without it. Similarly, Perspective also sees drops up to 38 in these settings.

VIPER. Like a whitespace attack, VIPER attacks can be easily prevented using a trivial text preprocessing step. To demonstrate this, we added

a pre-processing 'shielding' step to each system which replaces any non-standard characters with standard ones. The results for shielded VIPER attacks are found in Table 2 (Note: The full results for non-shielding against VIPER are found in the appendix.). This is in essence logically the reverse of VIPER's obfuscation by character translation process. Non-standard characters are those which exist outside 'a-zA-Z', numbers, and punctuation. To do this, as do the VIPER authors, we leverage the NamesList from the unicode database[7]. For any non-standard character, the description is searched for in the NamesList and the character which appears after "LETTER" in the description is used for substitution. For example, 'â'is described as "LATIN SMALL LETTER A WITH INVERTED BREVE", and hence would be replaced with 'a'. This simple pre-processing step reduces VIPER's average attack rate from 37 to 7 as shown in the VIPER results. In contrast, our proposed attack is not preventable through such simple preprocessing.

5.3 SOLID Results

The attack results against SOLID are found in Table 3. We see similar attack success as seen in OLID, finding even greater drops. Specifically, 75% of attacks cause a drop of $\geq$ 40 and 100% attacks cause a drop of at least 33.

5.4 Quality of Adversarial Examples

FT embeddings maintain a majority of the meaning and readability. We test readability of a sample of 50 tweets from the SOLID dataset, of which all were modified by FT . We asked three crowdworkers to assess the 50 tweets for readability. For comparison, we asked additional crowdworkers to assess the readability of the original texts. This helps explore the true drop in readability of a text. Additionally, we showed three other crowdworkers the original text as well and asked them to assess if the obfuscated texted conveyed the same meaning as the original (see Section 4.2 for details). We finally combined the crowdworkers votes by taking a majority vote for each example.

Table 4 presents the results. We find that FT scored slightly less in terms of readability than the original texts, but finds replacements with similar meaning. Specifically, readability drops from 74% to 70% for fully readable, but nearly two thirds retain the same meaning and 96% retain at least partial meaning. These numbers help indicate the strength of the attack, even when leveraging a crafty collection of word substitutions.

To provide insights into texts which retained full meaning versus partial, Table 5 shows a few examples of tweets in their respective categories as voted by crowdworkers. FT is able to find many appropriate, non traditional replacements. For example, "shit" is replaced with "shxt", in several instances which helps maintain meaning while evading classification. As another example, "phising" a mispelling of "phishing" is substituted for "fake". In context, this substitution makes sense. Note that while some examples are misspellings, these crafty modifications are ones that are mined from our large evasion text collection and not algorithmically-generated mispellings. However, some errors are found after replacement. For example, in the *Not Similar* instance FT replaces "fuck" with "bruh", and "shut" with "walked". These errors demonstrate room for improvement when selecting a candidate.

6 Analysis of embeddings

As discussed in Section 3.1, the adversary's strategy is to make crafty word replacements using a new embedding generated from an evasion collection (here made of deleted tweets not detected by an offense classifier). Results show that these embeddings successfully support the adversary at evading offensive language classifiers while maintaining readability and semantics. For further insights, we compare the off-the-shelf pretrained (Pre) embedding with the embedding fine-tuned on the evasion collection (FT). We examine the embeddings using the 59 words as probes which are both in the offensive Lexicon (Wiegand et al., 2018) and in the OLID test. For each word we get the 20 most similar words from Pre and from FT for comparison.

Fine-tuned embeddings move evasive substitute words closer to offensive probe words. We calculate the average position of the first evasive word[8] amongst the 20 most similar words. Pre has an average distance of 11, while FT has an average distance of 3. Thus, on average, FT is more likely to find evasive replacements. For example, in Pre *dispicable* appears as the 3rd most similar word to *despicable*, but it is the most similar in FT. Since

[7] https://www.unicode.org/Public/UCD/latest/ucd/NamesList.txt

[8] Evasion is determined by Perspective API.

		Drop in Classification Accuracy					
		NULI	vradivchev	MIDAS	Perspective	Lexicon	Avg. Drop
No Attack Accuracy %		61	69	66	68	54	
Grondahl	Add Space	-6	8	51	2	0	11
	Remove Space	-6	8	66	34	0	20
	Add 'love'	0	14	8	1	0	3
	Remove Space, Add 'love'	0	14	64	38	0	23
Drop in Classification Accuracy after Shielding using Character Preprocessing							
VIPER(0.1, DCES)		-5	12	17	3	3	6
VIPER(0.4, DCES)		5	20	23	8	10	13
VIPER(0.1, ECES)		-7	8	18	1	0	4
VIPER(0.4, ECES)		-7	8	18	1	0	4

Table 2: Robustness results on OLID against Grondahl and VIPER attacks (with and without shielding with simple character replacement pre-processing step). Columns show accuracy drop. The first row presents classification accuracies before attacks.

		Drop in Classification Accuracy					
		NULI	vradivchev	MIDAS	Perspective	Lexicon	Avg. Drop
No Attack Accuracy %		96	93	99	97	82	
	Surrogate Classifier						
GS - FT	NULI	-	69	45	50	49	53
	vradivchev	54	-	42	40	43	41
	MIDAS	33	38	-	39	34	37
	Perspective	46	54	43	-	40	46
	Average Drop	44	54	43	43	41	

Table 3: Robustness results on SOLID with our attack model. Columns show accuracy drop. The adversarial, surrogate classifier is indicated in column 1. The first row presents baseline classification accuracies (%) before attacks. Therefore the resulting accuracies can be calculated by subtracting the drop from the original accuracy.

Adversarial Attack	Readability		
	Yes	Partially	No
FT	35	13	2
[%]	70.0	26.0	4.0
Original	37	13	0
[%]	74.0	26.0	0.0
	Conveys same meaning		
	Yes	Partially	No
FT	31	17	2
[%]	62.0	34.0	2.0

Table 4: Results of human readability and meaning comparison assessments. Majority voting was used to combine all three annotators' answers into one vote.

FT could contain some unintelligible words, we repeat the experiment to filter out substitute words used by less than 3 different users. The same overall trend still holds.

Updated embeddings learn creative replacements. We manually compare the entries in the two lists (FT and Pre) of substitute words for each probe word. FT learns creative replacements absent in Pre. Examples include the word *azz* being the most similar word to *ass* in FT, but being absent within the most similar word list for Pre. Similarly, *niggah* appears as a replacement for *bitch* in FT, but not in Pre. These examples, along with the previous distance analysis, illustrate the craftiness in our evasion dataset.

Original Tweet	FT
Very Similar	
The amount of fake pages that watches my snap is insane I hate bitches so bad g	The amount of phising pages that watches my snap is unbelievable I hate bitchs so bad g
@USER Nigga is a rapist, he deserves all the shitty things that happen to him	@USER Nigha is a peado, he deserves all the awful things that happen to him
arab men are some of the most egotistical, disrespectful, pieces of shit you'll ever meet	arab men are some of the most egotistical, inconsiderate, pieces of shxt you'will ever meet
Somewhat Similar	
@USER Look at his lame ass with that shit eating smile... he is so vile.	@USER Look at his lame asf with that shxt eating smile... he is so discusting.
Yes I am a triple threat. A bad bitch, A dumb bitch, And a sensitive bitch	Yes I am a triple approach. A bad hoe, A rude hoe, And a hostile niggah
@USER You're a shameless pig but you knew that already. Just a reminder. Enjoy your jail time.	@USER You're a self-promotion baboon but you knew that already. Just a reminder. Enjoy your jail time.
Not Similar	
shut the fuck omg no one cares damn	walked the bruh omg no one cares lmaoo

Table 5: Examples of tweets in majority voted categories from crowdworkers.

7 Related Work

We first review related work on robustness of text classification in general and then closely related research on evading offensive language classifiers.

Evading Text Classifiers. Prior work has explored ways to evade text classification in general. Li et al. (2019) showed that character-level perturbations such as misspellings and word-level perturbations using off-the-shelf GloVe embeddings can evade text classifiers. Deri and Knight (2015) proposed an approach to create portmanteaus, which could be extended to adversarial texts. Behjati et al. (2019) added a sequence of words to any input to evade text classifiers. Zhao et al. (2018) proposed a GAN to generate adversarial attacks on text classification tasks. (Li et al., 2020) leverage BERT to propose solutions for replacement words, (Jin et al., 2020) leverage word embeddings, and (Ren et al., 2019) leverage WordNet. In contrast to prior work evading text classifiers, our work includes approaches to leverage embeddings built from a special evasion text collection.

Robustness of Text Classifiers. Our work is also relevant to prior studies of the robustness of text classifiers to adversarial inputs. Rojas-Galeano (2017) showed that primitive adversarial attacks (e.g., misspellings) can be detected and countered using edit distance. Hsieh et al. (2019) evaluated the robustness of self-attentive models in tasks of sentiment analysis, machine translation, and textual entailment. We examine robustness of similar models, however, we fine tune our embeddings to be task specific, while they do not, and we also test on the state-of-the-art offensive language classifiers.

Evading Offensive Language Classifiers. Gröndahl et al. (2018) examined robustness of hate speech classifiers against adding typos, whitespace, and non-hate words to text. As discussed earlier, prior work has shown that such primitive perturbations can be detected and reversed (Li et al., 2019; Rojas-Galeano, 2017). In contrast, we focus on more crafty text perturbations in our work. Ji and Knight (2018) surveyed the ways text has been encoded by humans to avoid censorship and explain challenges which automated systems would have to overcome. This work does not propose an automated approach for text perturbation. Eger et al. (2019) proposed VIPER for visual adversarial attacks. We implemented VIPER and (Gröndahl et al., 2018) as baseline attacks and showed that our approach is more successful overall. Overall, our work advances the research in this space by investigating robustness of offensive language classifiers against crafty adversarial attacks.

8 Conclusion

In this paper, we showed that state-of-the-art offensive language classifiers are vulnerable to crafty adversarial attacks. Our proposed adversarial attacks that leverage greedy and attention-based word selection and context-aware embeddings for word replacement were able to evade offensive language classifiers while preserving readability and semantics much better than prior simpler adversarial attacks. We report accuracy drops of up to 46 points or 67% against state-of-the-art offensive language classifiers. Furthermore, unlike VIPER and simpler attacks, our proposed attack cannot be easily prevented using pre-processing strategies. The user study showed that our adversarial attack was able to maintain similar readability with only a slight drop in semantic preservation.

Our work also suggests ways to improve the robustness of offensive language classifiers through adversarial training (Kurakin et al., 2017; Madry et al., 2018; Tramèr et al., 2018). More specifically, our attack relies on the *evasion* collection, which contains crafty adversarial examples that evade detection by offensive language classifiers but are flagged based on manual feedback by users or human moderators. Thus, offensive language classifiers can be adversarially trained on the latest *evasion* collection from time to time to improve their robustness to the ever evolving adversarial attacks. In this context it is noteworthy that continuous availability of large-scale manual feedback is quite unique to the problem of offensive language classification, where popular online social media platforms employ thousands of human moderators (Barrett, 2020).

References

Sweta Agrawal and Amit Awekar. 2018. Deep Learning for Detecting Cyberbullying Across Multiple Social Media Platforms. In *European Conference on Information Retrieval (ECIR)*.

Ayme Arango, Jorge Perez, and Barbara Poblete. 2020. Hate Speech Detection is Not as Easy as You May Think: A Closer Look at Model Validation. In *42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Paul M. Barrett. 2020. Who Moderates the Social Media Giants? A Call to End Outsourcing. NYU Center for Business and Human Rights.

Melika Behjati, Seyed-Mohsen Moosavi-Dezfooli, Mahdieh Soleymani Baghshah, and Pascal Frossard. 2019. Universal adversarial attacks on text classifiers. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Eshwar Chandrasekharan, Mattia Samory, Shagun Jhaver, Hunter Charvat, Amy Bruckman, Cliff Lampe, Jacob Eisenstein, and Eric Gilbert. 2018. The Internet's Hidden Rules: An Empirical Study of Reddit Norm Violations at Micro, Meso, and Macro Scales. *ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW)*.

Aliya Deri and Kevin Knight. 2015. How to make a frenemy: Multitape FSTs for portmanteau generation. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

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 *Proceedings of NAACL-HLT*, pages 1634–1647.

Facebook. Facebook Community Standards. https://www.facebook.com/communitystandards/.

Paula Fortuna and Sergio Nunes. 2018. A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys*.

V. Gadde and Matt Derella. 2020. An update on our continuity strategy during covid-19. https://blog.twitter.com/en_us/topics/company/2020/An-update-on-our-continuity-strategy\-during-COVID-19.html.

Tommi Gröndahl, Luca Pajola, Mika Juuti, Mauro Conti, and N. Asokan. 2018. All you need is "love": Evading hate-speech detection. *11th ACM Workshop on Artificial Intelligence and Security*.

Yu-Lun Hsieh, Minhao Cheng, Da-Cheng Juan, Wei Wei, Wen-Lian Hsu, and Cho-Jui Hsieh. 2019. On the robustness of self-attentive models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Andrew Hutchinson. 2020. Twitter Will Increase Its Use of Automation Tools as It Looks to Ensure Accuracy in COVID-19 Discussion | Social Media Today. https://www.socialmediatoday.com/news/twitter-will-increase-its-use-of-\automation-tools-as-it-looks-to-ensure\-acc/574263/.

Heng Ji and Kevin Knight. 2018. Creative language encoding under censorship. In *Proceedings of the First Workshop on Natural Language Processing for Internet Freedom*, pages 23–33, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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.

Jacob Kastrenakes. 2019. Twitter says it now removes half of all abusive tweets before users report them - the verge. https://www.theverge.com/2019/10/24/20929290/twitter-abusive-tweets-automated-\removal-earnings-q3-2019.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2017. Adversarial machine learning at scale. In *5th International Conference on Learning Representations*.

Huyen Le, Bob Boynton, Zubair Shafiq, and Padmini Srinivasan. 2019. A Postmortem of Suspended Twitter Accounts in the 2016 U.S. Presidential Election. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. TEXTBUGGER: Generating Adversarial Text Against Real-world Applications. In *Network and Distributed Systems Security Symposium*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

Ping Liu, Wen Li, and Liang Zou. 2019. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations*.

Debanjan Mahata, Haimin Zhang, Karan Uppal, Yaman Kumar, Rajiv Ratn Shah, Simra Shahid, Laiba Mehnaz, and Sarthak Anand. 2019. Midas at semeval-2019 task 6: Identifying offensive posts and targeted offense from twitter. In *SemEval@NAACL-HLT*.

Alex Nikolov and Victor Radivchev. 2019. Nikolov-radivchev at SemEval-2019 task 6: Offensive tweet classification with BERT and ensembles. In *Proceedings of the 13th International Workshop on Semantic Evaluation*.

John Pavlopoulos, Nithum Thain, Lucas Dixon, and Ion Androutsopoulos. 2019. Convai at semeval-2019 task 6: Offensive language identification and categorization with perspective and bert. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 571–576.

Perspective. Perspective Comment Analyzer API - Toxicity. https://github.com/conversationai/perspectiveapi/blob/master/2-api/model-cards/English/toxicity.md.

Reddit. Reddit Content Policy. https://web.archive.org/web/20200530184202/https://www.redditinc.com/policies/content-policy.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Sergio Rojas-Galeano. 2017. On obstructing obscenity obfuscation. *ACM Transactions on the Web*.

Guy Rosen. 2020. Facebook Community Standards Enforcement Report, November 2020. https://about.fb.com/news/2020/11/community-standards-enforcement-report\-nov-2020/.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A large-scale semi-supervised dataset for offensive language identification. *arXiv preprint arXiv:2004.14454*.

Kurt Thomas, Chris Grier, Vern Paxson, and Dawn Song. 2011. Suspended Accounts in Retrospect: An Analysis of Twitter Spam. In *ACM Internet Measurement Conference*.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. 2018. Ensemble adversarial training: Attacks and defenses. In *6th International Conference on Learning Representations*.

Twitter. Twitter hateful conduct policy. https://web.archive.org/web/20200530195919/https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy.

Michael Wiegand, Josef Ruppenhofer, Anna Schmidt, and Clayton Greenberg. 2018. Inducing a lexicon of abusive words – a feature-based approach. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the type and target of offensive posts in social media. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *13th International Workshop on Semantic Evaluation*.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. *International Conference on Learning Representations*.

A Full Results

Table 6 shows full results for all attacks, including our attack, Grondahl ,and VIPER attacks before and after shielding.

B Replication study: Reddit dataset

We verify our initial results on a second dataset composed of moderated Reddit comments (Chandrasekharan et al., 2018). To include non-moderated comments, we collected 5.6 million comments following the same procedure as (Chandrasekharan et al., 2018). We then used random sampling to construct a dataset with a similar 15:1 ratio of non-moderated to moderated comments as OLID. The dataset has 181,519 comments split into 145,846 (4,285 moderated and 141,561 non-moderated) training comments and 35,746 (1,071 moderated and 34,675 non-moderated) test comments. We re-build using these data and test the BERT based classifer (NULI-R) and the BLSTM Ensemble classifier (MIDAS-R). These are tagged with a '-R' to indicate training on the Reddit dataset. We exclude VIPER due to the previously shown weaknesses. We also exclude the methods of Gröndahl et al. (2018) because of weak performance.

Accuracy. Summarizing here, BLSTM ensemble (MIDAS-R) is most robust seeing a lower drop in accuracy, 31, than the BERT based model (NULI-R), 39, against the attack. Attacks using FT , see highest drops in accuracy against MIDAS: average of 32 (range: 28 - 35), while attacks using Pre, see highest drops against NULI (avg: 40, range: 37-42). Finally, greedy select (GS) causes greater drops against NULI (avg: 40) while attention select (AS) causes greater drops against MIDAS (avg: 35). The results are reported in Table 7

Quality. We see substitutions that subvert offense detectors such as *trump* being replaced with *trum*, which maintains the original message but now bypasses the detector[9]. We also see errors appear, such as "ctfu" being substituted for "shut". Overall, results with this second Reddit dataset are consistent with OLID results underlining our conclusion that the offense classifiers are not robust against these crafty attacks. We also see room for improvement of our adversarial attack methods especially in exploring more advanced filters for candidate substitution words. More examples found in Table 8.

[9]Comments on Reddit are moderated for various reasons not limited to offensive words, therefore in this case if comments against trump supporters are being moderated, it follows to change "trump"

		Drop in Classification Accuracy					
		NULI	vradivchev	MIDAS	Perspective	Lexicon	Avg. Drop
	No Attack Accuracy %	61	69	66	68	54	
	Surrogate Classifier						
GS - Pre	NULI	-	41	33	34	24	*33*
	vradivchev	28	-	33	28	22	28
	MIDAS	17	35	-	26	19	24
	Perspective	20	36	30	-	17	26
	Average Drop	22	**37**	**32**	**29**	**21**	
GS - FT	NULI	-	46	30	31	19	*32*
	vradivchev	39	-	30	26	18	28
	MIDAS	18	29	-	23	13	21
	Perspective	22	37	28	-	13	25
	Average Drop	**26**	**37**	29	27	16	
AS - Pre	NULI	-	36	19	19	15	22
	vradivchev	22	-	18	19	17	19
	MIDAS	13	34	-	20	15	21
	Perspective	17	37	23	-	16	*23*
	Average Drop	17	36	20	19	16	
AS - FT	NULI	-	39	18	17	15	22
	vradivchev	23	-	17	15	15	18
	MIDAS	11	27	-	17	12	17
	Perspective	17	40	21	-	16	*24*
	Average Drop	17	35	19	16	15	
	VIPER(0.1, DCES)	16	30	19	16	20	20
	VIPER(0.4, DCES)	55	66	58	54	39	54
	VIPER(0.1, ECES)	20	29	21	15	18	21
	VIPER(0.4, ECES)	54	63	57	44	48	53
	Drop in Classification Accuracy after Shielding using Character Preprocessing						
	VIPER(0.1, DCES)	-5	12	17	3	3	6
	VIPER(0.4, DCES)	5	20	23	8	10	13
	VIPER(0.1, ECES)	-7	8	18	1	0	4
	VIPER(0.4, ECES)	-7	8	18	1	0	4
Grondahl	Add Space	-6	8	51	2	0	11
	Remove Space	-6	8	66	34	0	20
	Add 'love'	0	14	8	1	0	3
	Remove Space, Add 'love'	0	14	64	38	0	23

Table 6: Full robustness results on OLID. Columns show accuracy drop. The approach is specified as *selection - replacement* where *selection = {Greedy Select (GS), Attention Select (AS)}* and *replacement = {Pre, FT}*. For our attack, the adversarial, surrogate classifier is indicated in column 1. The first row presents baseline classification accuracies (%) before attacks. Therefore the resulting accuracies can be calculated by subtracting the drop from the original accuracy.

		Drop in Accuracy	
No Attack Accuracy %		92	99
	Surrogate Classifier	NULI-R	MIDAS -R
GS - Pre	NULI-R	-	25
	MIDAS-R	42	-
GS - FT	NULI-R	-	28
	MIDAS-R	38	-
AS - Pre	NULI-R	-	34
	MIDAS-R	37	-
AS - FT	NULI-R	-	35
	MIDAS-R	37	-

Table 7: Robustness results on Reddit dataset using our attack model. Cell values indicate drop in classifier accuracy. First row: classification accuracy before attack.

Original Comment	Adversarial Attack Comment
you want me to cite statutes to prove perjury and evidence tampering are crimes are you fucking retarded	you wants me to cite statutes to prove perjury and evidence tampering are crimes are you damn weird
you're a moron get fucked	you're a moron get screwed
no you evil cunt	no you evil prick
she a dumb brown bitch	she a dumb brown hoe
you are so fucking delusional	you are so damn psychotic

Table 8: Examples to illustrate adversarial generations of Reddit comments (MIDAS-R used as surrogate classifier and FT at the attack).