Build it Break it Fix it for Dialogue Safety: Robustness from Adversarial Human Attack

Emily Dinan
Facebook AI Research
edinan@fb.com

Samuel Humeau
Facebook AI Research
samuel.humeau@gmail.com

Bharath Chintagunta
Virginia Tech
jaic4@vt.edu

Jason Weston
Facebook AI Research
jase@fb.com

Abstract

The detection of offensive language in the context of a dialogue has become an increasingly important application of natural language processing. The detection of trolls in public forums (Galán-García et al., 2016), and the deployment of chatbots in the public domain (Wolf et al., 2017) are two examples that show the necessity of guarding against adversarially offensive behavior on the part of humans. In this work, we develop a training scheme for a model to become robust to such human attacks by an iterative build it, break it, fix it strategy with humans and models in the loop. In detailed experiments we show this approach is considerably more robust than previous systems. Further, we show that offensive language used within a conversation critically depends on the dialogue context, and cannot be viewed as a single sentence offensive detection task as in most previous work. Our newly collected tasks and methods are all made open source and publicly available.

1 Introduction

The detection of offensive language has become an important topic as the online community has grown, as so too have the number of bad actors (Cheng et al., 2017). Such behavior includes, but is not limited to, trolling in public discussion forums (Herring et al., 2002) and via social media (Silva et al., 2016; Davidson et al., 2017), employing hate speech that expresses prejudice against a particular group, or offensive language specifically targeting an individual. Such actions can be motivated to cause harm from which the bad actor derives enjoyment, despite negative consequences to others (Bishop, 2014). As such, some bad actors go to great lengths to both avoid detection and to achieve their goals (Shachaf and Hara, 2010). In that context, any attempt to automatically detect this behavior can be expected to be adversarially attacked by looking for weaknesses in the detection system, which currently can easily be exploited as shown in (Hosseini et al., 2017; Gröndahl et al., 2018). A further example, relevant to the natural langauge processing community, is the exploitation of weaknesses in machine learning models that *generate* text, to force them to emit offensive language. Adversarial attacks on the Tay chatbot led to the developers shutting down the system (Wolf et al., 2017).

In this work, we study the detection of offensive language in dialogue with models that are robust to adversarial attack. We develop an automatic approach to the "Build it Break it Fix it" strategy originally adopted for writing secure programs (Ruef et al., 2016), and the "Build it Break it" approach consequently adapting it for NLP (Ettinger et al., 2017). In the latter work, two teams of researchers, "builders" and "breakers" were used to first create sentiment and semantic role-labeling systems and then construct examples that find their faults. In this work we instead fully automate such an approach using crowdworkers as the humans-in-the-loop, and also apply a fixing stage where models are retrained to improve them. Finally, we repeat the whole build, break, and fix sequence over a number of iterations.

We show that such an approach provides more and more robust systems over the fixing iterations. Analysis of the type of data collected in the iterations of the break it phase shows clear distribution changes, moving away from simple use of profanity and other obvious offensive words to utterances that require understanding of world knowledge, figurative language, and use of negation to detect if they are offensive or not. Further, data collected in the context of a dialogue rather than a sentence without context provides more sophisticated attacks. We show that model architectures that use the dialogue context efficiently perform much bet-

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 4537–4546,
Hong Kong, China, November 3–7, 2019.

ter than systems that do not, where the latter has been the main focus of existing research (Wulczyn et al., 2017; Davidson et al., 2017; Zampieri et al., 2019).

Code for our entire build it, break it, fix it algorithm is made open source, complete with model training code and crowdsourcing interface for humans. Our data and trained models will also be made available for the community via ParlAI[1].

2 Related Work

The task of detecting offensive language has been studied across a variety of content classes. Perhaps the most commonly studied class is hate speech, but work has also covered bullying, aggression, and toxic comments (Zampieri et al., 2019).

To this end, various datasets have been created to benchmark progress in the field. In hate speech detection, recently Davidson et al. (2017) compiled and released a dataset of over 24,000 tweets labeled as containing hate speech, offensive language, or neither. In order to benchmark toxic comment detection, The Wikipedia Toxic Comments dataset (which we study in this work) was collected and extracted from Wikipedia Talk pages and featured in a Kaggle competition (Wulczyn et al., 2017; Google, 2018). Each of these benchmarks examine only single-turn utterances, outside of the context in which the language appeared. In this work we recommend that future systems should move beyond classification of singular utterances and use contextual information to help identify offensive language.

Many approaches have been taken to solve these tasks – from linear regression and SVMs to deep learning (Noever, 2018). The best performing systems in each of the competitions mentioned above (for aggression and toxic comment classification) used deep learning approaches such as LSTMs and CNNs (Kumar et al., 2018; Google, 2018). In this work we consider a large-pretrained transformer model which has been shown to perform well on many downstream NLP tasks (Devlin et al., 2018).

The broad class of adversarial training is currently a hot topic in machine learning (Goodfellow et al., 2014). Use cases include training image generators (Brock et al., 2018) as well as image classifiers to be robust to adversarial examples (Liu et al., 2019). These methods find the breaking examples algorithmically, rather than by using humans breakers as we do. Applying the same approaches to NLP tends to be more challenging because, unlike for images, even small changes to a sentence can cause a large change in the meaning of that sentence, which a human can detect but a lower quality model cannot. Nevertheless algorithmic approaches have been attempted, for example in text classification (Ebrahimi et al., 2018), machine translation (Belinkov and Bisk, 2018), dialogue generation tasks (Li et al., 2017) and reading comprehension (Jia and Liang, 2017). The latter was particularly effective at proposing a more difficult version of the popular SQuAD dataset.

As mentioned in the introduction, our approach takes inspiration from "Build it Break it" approaches which have been successfully tried in other domains (Ruef et al., 2016; Ettinger et al., 2017). Those approaches advocate finding faults in systems by having humans look for insecurities (in software) or prediction failures (in models), but do not advocate an automated approach as we do here. Our work is also closely connected to the "Mechanical Turker Descent" algorithm detailed in (Yang et al., 2018) where language to action pairs were collected from crowdworkers by incentivizing them with a game-with-a-purpose technique: a crowdworker receives a bonus if their contribution results in better models than another crowdworker. We did not gamify our approach in this way, but still our approach has commonalities in the round-based improvement of models through crowdworker interaction.

3 Baselines: Wikipedia Toxic Comments

In this section we describe the publicly available data that we have used to bootstrap our *build it break it fix it* approach. We also compare our model choices with existing work and clarify the metrics chosen to report our results.

Wikipedia Toxic Comments The Wikipedia Toxic Comments dataset (WTC) has been collected in a common effort from the Wikimedia Foundation and Jigsaw (Wulczyn et al., 2017) to identify personal attacks online. The data has been extracted from the Wikipedia Talk pages, discussion pages where editors can discuss improvements to articles or other Wikipedia pages. We considered the version of the dataset that corresponds to the Kaggle competition: "Toxic Comment Classification Challenge" (Google, 2018)

[1] https://parl.ai/projects/dialogue_safety/

which features 7 classes of toxicity: toxic, severe toxic, obscene, threat, insult, identity hate and non-toxic. In the same way as in (Khatri et al., 2018), every label except non-toxic is grouped into a class OFFENSIVE while the non-toxic class is kept as the SAFE class. In order to compare our results to (Khatri et al., 2018), we similarly split this dataset to dedicate 10% as a test set. 80% are dedicated to train set while the remaining 10% is used for validation. Statistics on the dataset are shown in Table 1.

Models We establish baselines using two models. The first one is a binary classifier built on top of a large pre-trained transformer model. We use the same architecture as in BERT (Devlin et al., 2018). We add a linear layer to the output of the first token ([CLS]) to produce a final binary classification. We initialize the model using the weights provided by (Devlin et al., 2018) corresponding to "BERT-base". The transformer is composed of 12 layers with hidden size of 768 and 12 attention heads. We fine-tune the whole network on the classification task. We also compare it the fastText classifier (Joulin et al., 2017) for which a given sentence is encoded as the average of individual word vectors that are pre-trained on a large corpus issued from Wikipedia. A linear layer is then applied on top to yield a binary classification.

Experiments We compare the two aforementioned models with (Khatri et al., 2018) who conducted their experiments with a BiLSTM with GloVe pre-trained word vectors (Pennington et al., 2014). Results are listed in Table 2 and we compare them using the weighted-F1, i.e. the sum of F1 score of each class weighted by their frequency in the dataset. We also report the F1 of the OFFENSIVE-class which is the metric we favor within this work, although we report both. (Note that throughout the paper, the notation F1 is always referring to OFFENSIVE-class F1.) Indeed, in the case of an imbalanced dataset such as Wikipedia Toxic Comments where most samples are SAFE, the weighted-F1 is closer to the F1 score of the SAFE class while we focus on detecting OFFENSIVE content. Our BERT-based model outperforms the method from Khatri et al. (2018); throughout the rest of the paper, we use the BERT-based architecture in our experiments. In particular, we used this baseline trained on WTC to bootstrap our approach, to be described subsequently.

	Train	Valid	Test
SAFE	89.8%	89.7%	90.1%
OFFENSIVE	10.2%	10.3%	9.1%
Total	114656	15958	15957

Table 1: Dataset statistics for our splits of Wikipedia Toxic Comments.

	OFFENSIVE F1	Weighted F1
fastText	71.4%	94.8%
BERT-based	83.4%	96.7%
(Khatri et al., 2018)	-	95.4%

Table 2: Comparison between our models based on fastText and BERT with the BiLSTM used by (Khatri et al., 2018) on Wikipedia Toxic Comments.

4 Build it Break it Fix it Method

In order to train models that are robust to adversarial behavior, we posit that it is crucial collect and train on data that was collected in an adversarial manner. We propose the following automated build it, break it, fix it algorithm:

1. **Build it:** Build a model capable of detecting OFFENSIVE messages. This is our best-performing BERT-based model trained on the Wikipedia Toxic Comments dataset described in the previous section. We refer to this model throughout as A_0.

2. **Break it:** Ask crowdworkers to try to "beat the system" by submitting messages that our system (A_0) marks as SAFE but that the worker considers to be OFFENSIVE.

3. **Fix it:** Train a new model on these collected examples in order to be more robust to these adversarial attacks.

4. **Repeat:** Repeat, deploying the newly trained model in the **break it** phase, then **fix it** again.

See Figure 1 for a visualization of this process.

4.1 Break it Details

Definition of OFFENSIVE Throughout data collection, we characterize OFFENSIVE messages for users as messages that would not be "ok to send in a friendly conversation with someone you just met online." We use this specific language in an attempt to capture various classes of content that would be considered unacceptable in a friendly

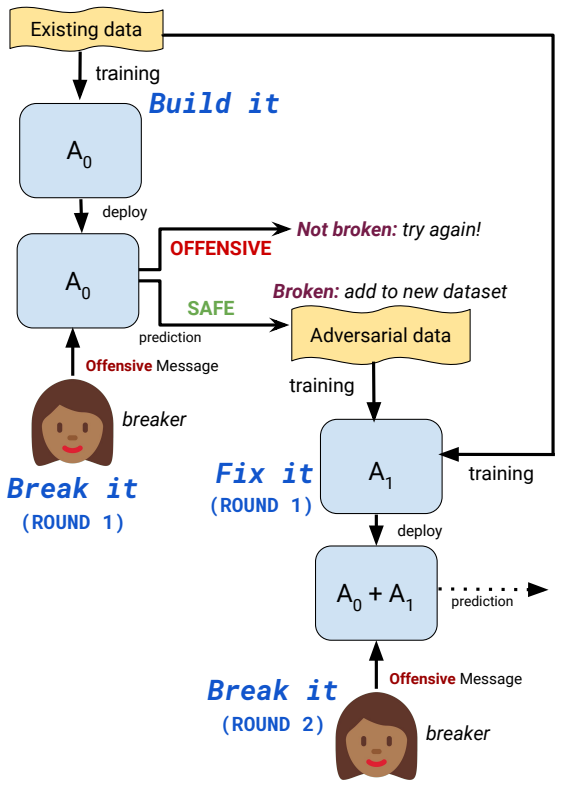

Figure 1: The build it, break it, fix it algorithm we use to iteratively train better models $A_0, \ldots, A_N$. In experiments we perform $N = 3$ iterations of the break it, fix it loop for the single-turn utterance detection task, and a further iteration for the multi-turn task in a dialogue context setting.

conversation, without imposing our own definitions of what that means. The phrase "with someone you just met online" was meant to mimic the setting of a public forum.

Crowderworker Task We ask crowdworkers to try to "beat the system" by submitting messages that our system marks as SAFE but that the worker considers to be OFFENSIVE. For a given round, workers earn a "game" point each time they are able to "beat the system," or in other words, trick the model by submitting OFFENSIVE messages that the model marks as SAFE. Workers earn up to 5 points each round, and have two tries for each point: we allow multiple attempts per point so that workers can get feedback from the models and better understand their weaknesses. The points serve to indicate success to the crowdworker and motivate to achieve high scores, but have no other meaning (e.g. no monetary value as in (Yang et al., 2018)). More details regarding the user interface and instructions can be found in Appendix B.

Models to Break During round 1, workers try to break the baseline model A_0, trained on Wikipedia Toxic Comments. For rounds i, $i > 1$, workers must break both the baseline model and the model from the previous "fix it" round, which we refer to as A_{i-1}. In that case, the worker must submit messages that both A_0 and A_{i-1} mark as SAFE but which the worker considers to be OFFENSIVE.

4.2 Fix it Details

During the "fix it" round, we update the models with the newly collected adversarial data from the "break it" round. The training data consists of all previous rounds of data, so that model A_i is trained on all rounds n for $n \leq i$, as well as the Wikipedia Toxic Comments data. We split each round of data into train, validation, and test partitions. The validation set is used for hyperparameter selection. The test sets are used to measure how robust we are to new adversarial attacks. With increasing round i, A_i should become more robust to increasingly complex human adversarial attacks.

5 Single-Turn Task

We first consider a single-turn set-up, i.e. detection of offensive language in one utterance, with no dialogue context or conversational history.

5.1 Data Collection

Adversarial Collection We collected three rounds of data with the build it, break it, fix it algorithm described in the previous section. Each round of data consisted of 1000 examples, leading to 3000 single-turn adversarial examples in total. For the remainder of the paper, we refer to this method of data collection as the *adversarial method*.

Standard Collection In addition to the *adversarial method*, we also collected data in a non-adversarial manner in order to directly compare the two set-ups. In this method – which we refer to as the *standard method*, we simply ask crowdworkers to submit messages that they consider to be OFFENSIVE. There is no model to break. Instructions are otherwise the same.

In this set-up, there is no real notion of "rounds", but for the sake of comparison we refer to each subsequent 1000 examples collected in this manner as a "round". We collect 3000 examples – or three rounds of data. We refer to a model trained on rounds $n \leq i$ of the *standard* data as S_i.

Single-Turn Adversarial (Round 1) and Standard Task OFFENSIVE Examples

	contains profanity	non-profane offending words	contains negation	contains figurative language	requires world knowledge	contains sarcasm
Standard	13%	12%	12%	11%	8%	3%
Adversarial	0%	5%	23%	19%	14%	6%

Table 3: Language analysis of the single-turn *standard* and *adversarial* (round 1) tasks by human annotation of various language properties. Standard collection examples contain more words found in an offensive words list, while *adversarial* examples require more sophisticated language understanding.

5.1.1 Task Formulation Details

Since all of the collected examples are labeled as OFFENSIVE, to make this task a binary classification problem, we will also add SAFE examples to it. The "safe data" is comprised of utterances from the ConvAI2 chit-chat task (Dinan et al., 2019; Zhang et al., 2018) which consists of pairs of humans getting to know each other by discussing their interests. Each utterance we used was reviewed by two independent crowdworkers and labeled as SAFE, with the same characterization of SAFE as described before. For each partition (train, validation, test), the final task has a ratio of 9:1 SAFE to OFFENSIVE examples, mimicking the division of the Wikipedia Toxic Comments dataset used for training our baseline models. Dataset statistics for the final task can be found in Table 5. We refer to these tasks – with both SAFE and OFFENSIVE examples – as the *adversarial* and *standard* tasks.

5.1.2 Model Training Details

Using the BERT-based model architecture described in Section 3, we trained models on each round of the *standard* and *adversarial* tasks, multi-tasking with the Wikipedia Toxic Comments task. We weight the multi-tasking with a mixing parameter which is also tuned on the validation set. Finally, after training weights with the cross entropy loss, we adjust the final bias also using the validation set. We optimize for the sensitive class (i.e. OFFENSIVE-class) F1 metric on the *standard* and *adversarial* validation sets respectively.

For each task (*standard* and *adversarial*), on round i, we train on data from all rounds n for $n \leq i$ and optimize for performance on the validation sets $n \leq i$.

5.2 Experimental Results

We conduct experiments comparing the *adversarial* and *standard* methods. We break down the results into "break it" results comparing the data collected and "fix it" results comparing the models obtained.

	% with profanity	% with "not"	avg. # chars	avg. # tokens
Std. (Rnds 1-3)	18.2	2.8	48.6	9.4
Adv. Rnd 1	2.6	5.8	53.7	10.3
Adv. Rnd 2	1.5	5.5	44.5	9
Adv. Rnd 3	1.2	9.8	45.7	9.3
Multi-turn Adv.	1.6	4.9	36.6	7.8

Table 4: Percent of OFFENSIVE examples in each task containing profanity, the token "not", as well as the average number of characters and tokens in each example. Rows 1-4 are the single-turn task, and the last row is the multi-turn task. Later rounds have less profanity and more use of negation as human breakers have to find more sophisticated language to adversarially attack our models.

Rounds {1, 2 and 3}	Train	Valid	Test
SAFE **Examples**	21,600	2700	2700
OFFENSIVE **Examples**	2400	300	300
Total Examples	24,000	3,000	3,000

Table 5: Dataset statistics for the single-turn rounds of the *adversarial* task data collection. There are three rounds in total all of identical size, hence the numbers above can be divided for individual statistics. The *standard* task is an additional dataset of exactly the same size as above.

5.2.1 Break it Phase

Examples obtained from both the *adversarial* and *standard* collection methods were found to be clearly offensive, but we note several differences in the distribution of examples from each task, shown in Table 4. First, examples from the *standard* task tend to contain more profanity. Using a list of common English obscenities and otherwise bad words[2], in Table 4 we calculate the percentage of examples in each task containing such obscenities, and see that the *standard* examples contain

[2]https://github.com/LDNOOBW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words

		WTC Baseline	*Standard* models			*Adversarial* models		
Task Type	Task Round	A_0	S_1	S_2	S_3	A_1	A_2	A_3
WTC	-	**83.3**	80.6	81.1	82.1	81.3	78.9	78.0
Standard Task	All (1-3)	68.1	83.3	85.8	**88.0**	83.0	85.3	83.7
Adversarial Task	1	0.0	51.7	69.3	68.6	71.8	**79.0**	78.2
	2	0.0	10.8	26.4	31.8	0.0	**64.4**	62.1
	3	0.0	12.3	17.1	13.7	32.1	0.0	**59.9**
	All (1-3)	0.0	27.4	41.7	41.8	40.6	55.5	**67.6**

Table 6: Test performance of best standard models trained on *standard* task rounds (models S_i for each round i) and best *adversarial* models trained on adversarial task rounds (models A_i). All models are evaluated using OFFENSIVE-class F1 on each round of both the *standard* task and *adversarial* task. A_0 is the baseline model trained on the existing Wiki Toxic Comments (WTC) dataset. Adversarial models prove to be more robust than standard ones against attack (Adversarial Task 1-3), while still performing reasonably on Standard and WTC tasks.

at least seven times as many as each round of the *adversarial* task. Additionally, in previous works, authors have observed that classifiers struggle with negations (Hosseini et al., 2017). This is borne out by our data: examples from the single-turn *adversarial* task more often contain the token "not" than examples from the *standard* task, indicating that users are easily able to fool the classifier with negations.

We also anecdotally see figurative language such as "snakes hiding in the grass" in the adversarial data, which contain no individually offensive words, the offensive nature is captured by reading the entire sentence. Other examples require sophisticated world knowledge such as that many cultures consider eating cats to be offensive. To quantify these differences, we performed a blind human annotation of a sample of the data, 100 examples of standard and 100 examples of adversarial round 1. Results are shown in Table 3. Adversarial data was indeed found to contain less profanity, fewer non-profane but offending words (such as "idiot"), more figurative language, and to require more world knowledge.

We note that, as anticipated, the task becomes more challenging for the crowdworkers with each round, indicated by the decreasing average scores in Table 7. In round 1, workers are able to get past A_0 most of the time – earning an average score of 4.56 out of 5 points per round – showcasing how susceptible this baseline is to adversarial attack despite its relatively strong performance on the Wikipedia Toxic Comments task. By round 3, however, workers struggle to trick the system, earning an average score of only 1.6 out of 5. A finer-grained assessment of the worker scores can be found in Table 11 in the appendix.

	Single-Turn			Multi
Round	1	2	3	("4")
Avg. score (0-5)	4.56	2.56	1.6	2.89

Table 7: Adversarial data collection worker scores. Workers received a score out of 5 indicating how often (out of 5 rounds) they were able to get past our classifiers within two tries. In later single-turn rounds it is harder to defeat our models, but switching to multi-turn makes this easier again as new attacks can be found by using the dialogue context.

5.2.2 Fix it Phase

Results comparing the performance of models trained on the *adversarial* (A_i) and *standard* (S_i) tasks are summarized in Table 6, with further results in Table 13 in Appendix A.2. The adversarially trained models A_i prove to be more robust to adversarial attack: on each round of adversarial testing they outperform standard models S_i.

Further, note that the *adversarial* task becomes harder with each subsequent round. In particular, the performance of the standard models S_i rapidly deteriorates between round 1 and round 2 of the *adversarial* task. This is a clear indication that models need to train on adversarially-collected data to be robust to adversarial behavior.

Standard models (S_i), trained on the *standard* data, tend to perform similarly to the *adversarial* models (A_i) as measured on the *standard* test sets, with the exception of training round 3, in which A_3 fails to improve on this task, likely due to being too optimized for adversarial tasks. The *standard* models S_i, on the other hand, are improving

with subsequent rounds as they have more training data of the same distribution as the evaluation set. Similarly, our baseline model performs best on its own test set, but other models are not far behind.

Finally, we remark that all scores of 0 in Table 6 are by design, as for round i of the *adversarial* task, both A_0 and A_{i-1} classified each example as SAFE during the 'break it' data collection phase.

6 Multi-Turn Task

In most real-world applications, we find that adversarial behavior occurs in context – whether it is in the context of a one-on-one conversation, a comment thread, or even an image. In this work we focus on offensive utterances within the context of two-person dialogues. For dialogue safety we posit it is important to move beyond classifying single utterances, as it may be the case that an utterance is entirely innocuous on its own but extremely offensive in the context of the previous dialogue history. For instance, "Yes, you should definitely do it!" is a rather inoffensive message by itself, but most would agree that it is a hurtful response to the question "Should I hurt myself?"

6.1 Task Implementation

To this end, we collect data by asking crowdworkers to try to "beat" our best single-turn classifier (using the model that performed best on rounds 1-3 of the *adversarial* task, i.e., A_3), in addition to our baseline classifier A_0. The workers are shown truncated pieces of a conversation from the ConvAI2 chit-chat task, and asked to continue the conversation with OFFENSIVE responses that our classifier marks as SAFE. The resulting conversations – including the newly provided OFFENSIVE responses – are between 3 and 6 turns long. As before, workers have two attempts per conversation to try to get past the classifier and are shown five conversations per round. They are given a score (out of five) at the end of each round indicating the number of times they successfully fooled the classifier.

We collected 3000 offensive examples in this manner. As in the single-turn set up, we combine this data with SAFE examples with a ratio of 9:1 SAFE to OFFENSIVE for classifier training. The safe examples are dialogue examples from ConvAI2 for which the responses were reviewed by two independent crowdworkers and labeled as SAFE, as in the s single-turn task set-up. We refer to this overall task as the *multi-turn adversarial* task. Dataset statistics are given in Table 9.

6.2 Models

To measure the impact of the context, we train models on this dataset with and without the given context. We use the fastText and the BERT-based model described in Section 3. In addition, we build a BERT-based model variant that splits the last utterance (to be classified) and the rest of the history into two dialogue segments. Each segment is assigned an embedding and the input provided to the transformer is the sum of word embedding and segment embedding, replicating the setup of the Next Sentence Prediction that is used in the training of BERT (Devlin et al., 2018).

6.3 Experimental Results

6.3.1 Break it Phase

During data collection, we observed that workers had an easier time bypassing the classifiers than in the single-turn set-up. See Table 7. In the single-turn set-up, the task at hand gets harder with each round – the average score of the crowdworkers decreases from 4.56 in round 1 to 1.6 in round 3. Despite the fact that we are using our best single-turn classifier in the multi-turn set-up (A_3), the task becomes easier: the average score per round is 2.89. This is because the workers are often able to use contextual information to suggest something offensive rather than say something offensive outright. See examples of submitted messages in Table 8. Having context also allows one to express something offensive more efficiently: the messages supplied by workers in the multi-turn setting were significantly shorter on average, see Table 4.

6.3.2 Fix it Phase

During training, we multi-tasked the *multi-turn adversarial* task with the Wikipedia Toxic Comments task as well as the single-turn *adversarial* and *standard* tasks. We average the results of our best models from five different training runs. The results of these experiments are given in Table 10.

As we observed during the training of our baselines in Section 3, the fastText model architecture is ill-equipped for this task relative to our BERT-based architectures. The fastText model performs worse given the dialogue context (an average of 23.56 OFFENSIVE-class F1 relative to 37.1) than without, likely because its bag-of-

Multi-Turn Adversarial Task Examples		
Context	*[P1:]*	greetings and salutations my friend
	[P2:]	hi how are you doing?
	[P1:]	i'm well. enjoying some r r from the navy station
	[P2:]	i love to cook. do you cook?
	[P1:]	i do on occasion. i love to read more though.
OFFENSIVE **Response**	*[P2:]*	**reading the menu doesnt count**
Context	*[P1:]*	hey how are you
	[P2:]	good but i keep thinking about the time i almost played for nba
	[P1:]	aww why did not you play
	[P2:]	lost my nerve i guess. what do you do?
	[P1:]	oh that's to bad. i work at a gas station
OFFENSIVE **Response**	*[P2:]*	**i need to follow my dream to not work in a gas station**

Table 8: Examples from the *multi-turn adversarial* task. Responses can be offensive only in context.

Multi-Turn Adversarial Task	Train	Valid	Test
SAFE **Examples**	21,600	2,700	2,700
OFFENSIVE **Examples**	2,400	300	300
Total Examples	24,000	3,000	3,000

Table 9: *Multi-turn adversarial* task data statistics.

Multi-Turn Adversarial Task Results		
	F1	Weighted-F1
fastText		
with context	23.6 ± 1.9	85.9 ± 0.5
without context	37.1 ± 2.6	88.8 ± 0.6
BERT-based *(no segments)*		
with context	60.5 ± 1.3	92.2 ± 0.3
without context	56.8 ± 1.6	90.6 ± 0.7
BERT-based *(dialogue segments)*		
with context	66.4 ± 2.2	93.2 ± 0.4
without context	59.0 ± 2.5	91.2 ± 0.8

Table 10: Results of experiments on the multi-turn *adversarial* task. We denote the average and one standard deviation from the results of five runs. Models that use the context as input ("with context") perform better. Encoding this in the architecture as well (via BERT *dialogue segment* features) gives us the best results.

embeddings representation is too simple to take the context into account.

We see the opposite with our BERT-based models, indicating that more complex models are able to effectively use the contextual information to detect whether the response is SAFE or OFFENSIVE. With the simple BERT-based architecture (that does not split the context and the utterance into separate segments), we observe an average of a 3.7 point increase in OFFENSIVE-class F1 with the addition of context. When we use segments to separate the context from the utterance we are trying to classify, we observe an average of a 7.4 point increase in OFFENSIVE-class F1. Thus, it appears that the use of contextual information to identify OFFENSIVE language is critical to making these systems robust, and improving the model architecture to take account of this has large impact.

7 Conclusion

We have presented an approach to build more robust offensive language detection systems in the context of a dialogue. We proposed a build it, break it, fix it, and then repeat strategy, whereby humans attempt to break the models we built, and we use the broken examples to fix the models. We show this results in far more nuanced language than in existing datasets. The adversarial data includes less profanity, which existing classifiers can pick up on, and is instead offensive due to figurative language, negation, and by requiring more world knowledge, which all make current classifiers fail. Similarly, offensive language in the context of a *dialogue* is also more nuanced than standalone offensive utterances. We show that classifiers that learn from these more complex examples are indeed more robust to attack, and that using the dialogue context gives improved performance if the model architecture takes it into account.

In this work we considered a binary problem (offensive or safe). Future work could consider classes of offensive language separately (Zampieri et al., 2019), or explore other dialogue tasks, e.g. from social media or forums. Another interesting direction is to explore how our build it, break it, fix it strategy would similarly apply to make neural generative models safe (Henderson et al., 2018).

Acknowledgments

Thank you to Stephen Roller for providing insights and ideas for this project. The emoji image in Figure 1 is by Twemoji[3], and is licensed under CC BY-4.0.

References

2018. *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In (DBL, 2018).

Jonathan Bishop. 2014. Representations of trolls in mass media communication: a review of media-texts and moral panics relating to internet trolling. *International Journal of Web Based Communities*, 10(1):7–24.

Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Justin Cheng, Cristian Danescu-Niculescu-Mizil, Jure Leskovec, and Michael Bernstein. 2017. Anyone can become a troll: Causes of trolling behavior in online discussions. *American Scientist*, 105(3):152.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh International AAAI Conference on Web and Social Media*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. 2019. The second conversational intelligence challenge (convai2). *arXiv preprint arXiv:1902.00098*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 31–36. Association for Computational Linguistics.

[3] https://github.com/twitter/twemoji

Allyson Ettinger, Sudha Rao, Hal Daumé III, and Emily M Bender. 2017. Towards linguistically generalizable nlp systems: A workshop and shared task. *arXiv preprint arXiv:1711.01505*.

Patxi Galán-García, José Gaviria de la Puerta, Carlos Laorden Gómez, Igor Santos, and Pablo García Bringas. 2016. Supervised machine learning for the detection of troll profiles in twitter social network: Application to a real case of cyberbullying. *Logic Journal of the IGPL*, 24(1):42–53.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Google. 2018. Toxic comment classification challenge.

Tommi Gröndahl, Luca Pajola, Mika Juuti, Mauro Conti, and N Asokan. 2018. All you need is" love": Evading hate-speech detection. *arXiv preprint arXiv:1808.09115*.

Peter Henderson, Koustuv Sinha, Nicolas Angelard-Gontier, Nan Rosemary Ke, Genevieve Fried, Ryan Lowe, and Joelle Pineau. 2018. Ethical challenges in data-driven dialogue systems. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, pages 123–129, New York, NY, USA. ACM.

Susan Herring, Kirk Job-Sluder, Rebecca Scheckler, and Sasha Barab. 2002. Searching for safety online: Managing" trolling" in a feminist forum. *The information society*, 18(5):371–384.

Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving google's perspective api built for detecting toxic comments. *arXiv preprint arXiv:1702.08138*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In (Palmer et al., 2017), pages 2021–2031.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Chandra Khatri, Behnam Hedayatnia, Rahul Goel, Anushree Venkatesh, Raefer Gabriel, and Arindam Mandal. 2018. Detecting offensive content in open-domain conversations using two stage semi-supervision. *CoRR*, abs/1811.12900.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11, Santa Fe, New

Mexico, USA. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In (Palmer et al., 2017), pages 2157–2169.

Aishan Liu, Xianglong Liu, Jiaxin Fan, Yuqing Ma, Anlan Zhang, Huiyuan Xie, and Dacheng Tao. 2019. Perceptual-sensitive gan for generating adversarial patches.

David Noever. 2018. Machine learning suites for online toxicity detection. *arXiv preprint arXiv:1810.01869*.

Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors. 2017. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Andrew Ruef, Michael Hicks, James Parker, Dave Levin, Michelle L Mazurek, and Piotr Mardziel. 2016. Build it, break it, fix it: Contesting secure development. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 690–703. ACM.

Pnina Shachaf and Noriko Hara. 2010. Beyond vandalism: Wikipedia trolls. *Journal of Information Science*, 36(3):357–370.

Leandro Silva, Mainack Mondal, Denzil Correa, Fabrício Benevenuto, and Ingmar Weber. 2016. Analyzing the targets of hate in online social media. In *Tenth International AAAI Conference on Web and Social Media*.

Marty J Wolf, K Miller, and Frances S Grodzinsky. 2017. Why we should have seen that coming: comments on microsoft's tay experiment, and wider implications. *ACM SIGCAS Computers and Society*, 47(3):54–64.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1391–1399. ACM.

Zhilin Yang, Saizheng Zhang, Jack Urbanek, Will Feng, Alexander H. Miller, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Mastering the dungeon: Grounded language learning by mechanical turker descent. In (DBL, 2018).

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2204–2213. Association for Computational Linguistics.