# NaturalAdversaries: Can Naturalistic Adversaries Be as Effective as Artificial Adversaries?

**Saadia Gabriel**[♠] **Hamid Palangi**[♡] **Yejin Choi**[♠◇]
[♠]Paul G. Allen School of Computer Science & Engineering, University of Washington
[♡]Microsoft Research
[◇]Allen Institute for Artificial Intelligence
{skgabrie, yejin}@cs.washington.edu , hpalangi@microsoft.com

## Abstract

While a substantial body of prior work has explored adversarial example generation for natural language understanding tasks, these examples are often unrealistic and diverge from the real-world data distributions. In this work, we introduce a two-stage adversarial example generation framework (NaturalAdversaries), for designing adversaries that are effective at fooling a given classifier and demonstrate natural-looking failure cases that could plausibly occur during in-the-wild deployment of the models.

At the first stage a token attribution method is used to summarize a given classifier's behaviour as a function of the key tokens in the input. In the second stage a generative model is conditioned on the key tokens from the first stage. NaturalAdversaries is adaptable to both black-box and white-box adversarial attacks based on the level of access to the model parameters. Our results indicate these adversaries generalize across domains, and offer insights for future research on improving robustness of neural text classification models.

## 1 Introduction

Transformer models have gained prominence in NLP research due to their powerful performances on leaderboards. However, numerous studies have shown these neural models are brittle, frequently taking shortcuts to reach decisions rather than reasoning about the underlying semantics correctly (Geirhos et al.; Bras et al., 2020) or failing when exposed to adversarial perturbations of inputs (e.g, Goodfellow et al., 2015; Szegedy et al., 2015; Jia and Liang, 2017; Glockner et al., 2018; Dinan et al., 2019). Due to the opaque nature of neural modeling, methods for adversarial example generation may also steer algorithms towards generating unlikely examples that exhibit unrealistic properties (Zhao et al., 2018).

In this work, we pose the question, *"what does it really mean for an adversarial attack to be effective and can naturalistic adversaries match artificial ones?"* We argue that effectiveness should be dependent not only on attacking accuracy, but on usefulness of adversaries for improving robustness under realistic conditions (e.g identifying social biases learned by neural models, Buolamwini and Gebru, 2018; Sheng et al., 2019; Stanovsky et al., 2019; Sap et al., 2019; Ross et al., 2021).

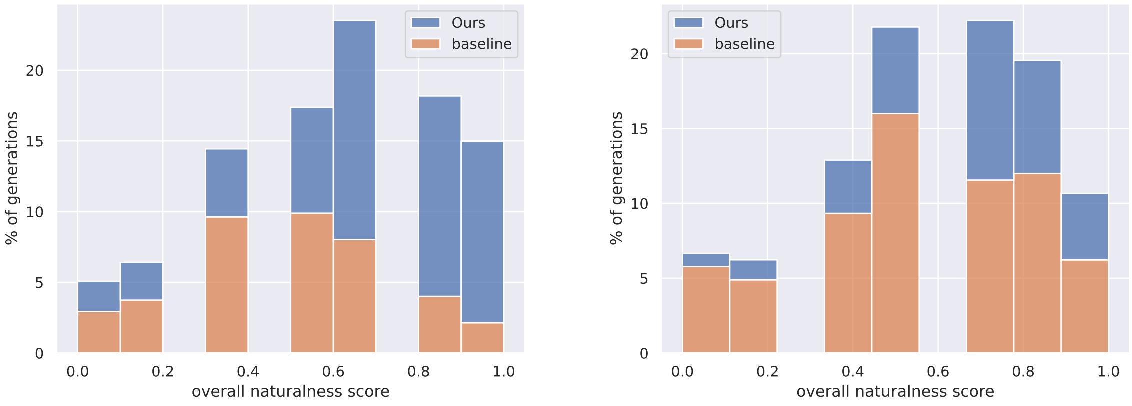
We propose a framework NaturalAdversaries[1] for generating convincingly naturalistic adversaries. We first approximate the behavior of a given classifier decision function $F_c(x)$ and then train a generative model $F_g(x)$ to mimic this behavior. As shown in Figure 2, we condition generative models on influential tokens extracted using black box or white box explainability methods (Ribeiro et al., 2016; Sundararajan et al., 2017), and a desired label (e.g. "*entailment*" or "*contradiction*"), to produce new examples that match a distribution learned from $F_c(x)$ through sampling.

Our results on two different tasks (*hate speech detection* and *natural language inference*) show that our approach leads to adversaries that are perceived by annotators as considerably more natural. While naturalistic adversaries are often less adversarial than artificial adversaries (following prior literature, e.g. Morris et al., 2020a), we find this depends on the evaluation setting and they can be better defenses.

## 2 Defining Naturalness

We first define what it means for machine-generated adversaries to have the quality of "naturalness." Prior work on text generation has defined this property in terms of linguistic competence (Novikova et al., 2018; Lau et al., 2020), as well as enumerating undesirable characteristics that lower perceived naturalness like self-contradiction (Dou et al., 2022). In our work, we ask human

[1] Code and data can be found here: https://github.com/skgabriel/NaturalAdversaries.



(a) Distribution for ineffective examples.

(b) Distribution for effective examples.

Figure 1: Averaged naturalness scores from human evaluation. The adversarial examples were generated from the DynaHate test set (Vidgen et al., 2021a) using two common baselines (baseline) (Ebrahimi et al., 2018; Jin et al., 2020) as well as NaturalAdversaries (Ours). We show the distribution of scores for examples that are effective or ineffective respectively at fooling a RoBERTa toxicity classification model (Zhou et al., 2021). This shows that not only does NaturalAdversaries generate more natural examples, but naturalistic examples can also be effective at demonstrating adversarial behavior.

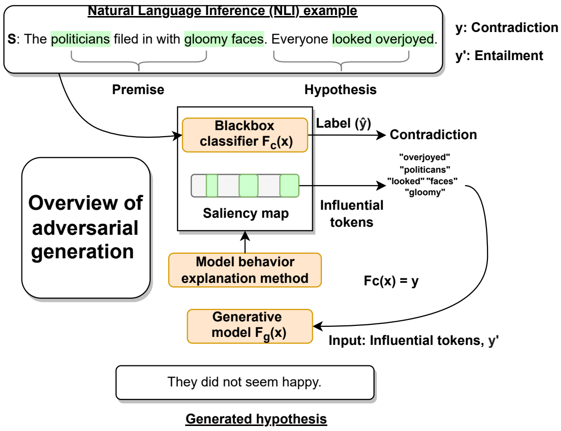

Natural Language Inference (NLI) example
S: The politicians filed in with gloomy faces. Everyone looked overjoyed.
y: Contradiction
y': Entailment
Premise
Hypothesis
Blackbox classifier F_c(x)
Label (ŷ)
Contradiction
"overjoyed"
"politicans"
"looked" "faces"
"gloomy"
Saliency map
Influential tokens
Overview of adversarial generation
Model behavior explanation method
Fc(x) = y
Generative model F_g(x)
Input: Influential tokens, y'
They did not seem happy.
Generated hypothesis

Figure 2: Our proposed framework for testing robustness of models using machine-generated adversarial examples (NaturalAdversaries). In the initial step, we probe the behavior of a black-box classifier (e.g. RoBERTa) using an explainability method like integrated gradients to find tokens with high attribution weights (influential tokens). We then use a generative model (e.g. GPT-2) to produce new adversarial examples conditioned on these tokens.

evaluators to judge naturalness in terms of whether generated text fragments are *coherent*, *well-formed* and *likely to be human-written*.

## 3 Description of NaturalAdversaries

Our overall framework consists of two stages - (1) a probing stage where we identify the influential (high attribution) tokens and (2) an adversarial generative stage where we generate unseen challenging examples by conditioning on the extracted tokens and a reversed label (§3.2). We focus on two types of explainability methods as a means of summarizing model behavior through sampling - (1) LIME local linear explanation models (Ribeiro et al., 2016) and (2) gradient attribution scores (Sundararajan et al., 2017). Using these methods as part of our proposed method is advantageous since it doesn't require curation of cherry-picked examples to probe model behavior, and is agnostic to the specific internal structure of a given classifier. Given a sequence of text tokens $S = [s_0, s_1, \ldots, s_i, s_{i+1}, \ldots]$ with classifier label $\hat{y}$, each of these methods define a scoring function $F_{attr}(s_i)$ which we use for assigning attribution scores to each token $s_i$ which represents its overall contribution to the classifier's decision. We separate these approaches based on whether $F_{attr}$ is conditioned on the model parameters or not. If it is, this demonstrates a white-box attack that can adapt to the vulnerabilities of a specific classifier given complete access to the model (e.g., using gradient attribution scores, see §A.2 in the Appendix for details). In the black-box setting (e.g. using LIME attribution scores), the underlying architecture and parameters are not known, and only sampled predictions are used to approximate the model behavior (See §A.1 in the Appendix).

| Dataset | Taxonomy | Example | Classifier(s) |
|---|---|---|---|
| DynaHate (Vidgen et al., 2021a) | hate / nothate | I say I like women, but I don't | RoBERTa, BERT |
| ANLI (Nie et al., 2020) | contradiction / neutral / entailment | **P**: P-17 is a mixed use skyscraper proposed for construction in Dubai... The design is for a 379 m tall building, comprising 78 floors. **H**: P-17 is designed to have 78 floors and be over 500 meters tall. | DeBERTa, BERT |

Table 1: Description of considered datasets. For ANLI, each example comes with a premise (**P**) providing context and a hypothesis statement (**H**).

### 3.1 Domains

### 3.2 Adversarial Generation

Given a generative autoregressive model $F_g$ and training set $D_1$ with triples $(S, y, F_{attr}(S))$, we construct the following input sequence

$$x = [attr, z, label, y', text, S, eos] \quad (1)$$

where $z$ is a sequence of influential tokens sampled from $S$ using the attribution weights defined by $F_{attr}(S)$, $attr$ is a special token indicating the start of this sequence, $y'$ is the desired classification label, $label$ and $text$ are special tokens indicating the start of $y'$ and $S$ respectively, and $eos$ is a special token indicating where the full input sequence ends. At training time, $y' = y$ as the generated model is trained to mimic the behavior of the classifier model and generate examples with a given label $y'$ based on the classifier's observed behavior. At decoding time, we encourage adversarial behavior by reversing the label (e.g. setting $y' = 1 - y$). The model is prompted using only the influential tokens $z$ and $y'$. For example, given a natural language inference (NLI) premise and hypothesis pair (*"It was sunny outside"*, *"it was too dark to see anything outside"*) where the gold label is contradiction, at training time we use ($y'$="contradiction";$z$="influentialWord1", "influentialWord2", "influentialWord3", $S$=*"It was sunny outside. It was too dark to see anything outside."*). At decoding time we would use ($y'$="entailment";$z$="influentialWord1", "influentialWord2", "influentialWord3") and predict $S$.

We minimize cross-entropy loss during training time:

$$\mathcal{L}_{CE} = -\frac{1}{|S|}\sum_{i=1}^{|S|} logP(S_i|S_1, ..., S_{i-1}). \quad (2)$$

## 4 Experimental Setup

In this section we first introduce the domains we test on (§3.1) and then methods used for baseline comparison (§4.1). All adversarial generators are based on the GPT-2 124M parameter model. We describe evaluation setups (§4.2.1), as well as out-of-distribution evaluation (§4.2.2).

An advantage of proposed generative method is that we can automatically extend human-in-the-loop adversarial generation methods like Adversarial NLI (ANLI, Nie et al., 2020), which are costly and time-consuming to curate. Given this motivation, we focus on particularly challenging human-in-the-loop examples rather than cases that are already well solved by existing benchmarks. To study effectiveness of the proposed approach, we conducted experiments on the hate speech detection (DynaHate (Vidgen et al., 2021b)) and natural language inference (NLI). For DynaHate we use a RoBERTa classifier trained on tweets (Founta et al., 2018; Zhou et al., 2021). We test generalization across both model architectures and (non-adversarial) data domains using a BERT model (Devlin et al., 2019) trained on the HateXplain dataset (Mathew et al., 2021). For NLI we use DeBERTa (He et al., 2021) trained on MNLI (Williams et al., 2018). We test generalization using BERT trained on the QNLI dataset (Wang et al., 2019). Further details are provided in Table 1 and Appendix B.

| Dataset | Model | Natural$_H$ (%) | Adv1 (%) | Adv2 (%) | HateCheck (F1) |
|---|---|---|---|---|---|
| DynaHate | Original | - | - | - | 55.01 |
| | TF | 53 | **69** | **59** | 55.59 |
| | HF | 27 | 30 | 55 | 55.92 |
| | NA-LIME | 67 | 30 | 55 | 55.90 |
| | NA-IG | **73** | 21 | 36 | **56.69** |
| Dataset | Model | Natural$_H$ (%) | Adv1 (%) | Adv2 (%) | SNLI-Hard (F1) |
| ANLI | Original | - | - | - | 76.95 |
| | TF | 57 | **57** | **46** | 76.82 |
| | HF | 64 | 33 | 38 | **76.98** |
| | NA-LIME | 73 | 31 | 43 | **76.98** |
| | NA-IG | **89** | 27 | 42 | 76.97 |

Table 2: Human evaluation (Natural$_H$) of naturalness, along with adversarial performance against the original target classifier $Adv1$ and an unseen classifier $Adv2$. In the last topright column we show macro-averaged F1 performance on HateCheck (Röttger et al., 2021) after finetuning RoBERTa on 150 adversarial examples, compared to the original performance. We conduct a similar experiment for NLI using the SNLI-Hard evaluation set (Gururangan et al., 2018) with results in the last bottomright column. We bold the best-performing model and underline the second best model.

### 4.1 Baselines

For automatic adversarial example construction, we compare against several common adversarial example generation approaches which are designed for either **black-box** (model-agnostic) or **white-box** (model-dependent) attacks. Baselines are implemented using TextAttack (Morris et al., 2020b).

**Black-Box Baselines** We use the TextFooler (Jin et al., 2020) algorithm for generating coherent adversaries by replacing high-importance words in original examples with words that preserve semantic similarity.

**White-Box Baselines** Since our approach has an advantage over other baselines in the white-box setting of utilizing knowledge about model parameters, we also compare against a word-level version of the widely used HotFlip gradient-based approach (Ebrahimi et al., 2018).

### 4.2 Evaluation Metrics

#### 4.2.1 Human Evaluation

To compare effectiveness of automatic methods, adversarial examples are manually validated to determine the true label. We also assess *naturalness* of examples, i.e. whether they are perceived as realistic examples that could be written by humans. We use 156 crowd-source workers from Amazon Mechanical Turk (MTurk) with prior experience validating hate speech (Sap et al., 2020) and 79 workers with experience validating NLI data (Liu et al., 2022). We sample 150 examples using each approach (some approaches may impose constraints that are unsatisfied by all candidate transformed sentences, we also filter out examples that are already adversarial to avoid conflating adversarialness of original examples with effects of the transformation). Each example is judged by 3 different workers. For hate speech, we classify an example as toxic if at least one annotator considers it so. We achieve moderate inter-annotator agreement of Fleiss' $\kappa = .51$ for hate speech and $\kappa = .52$ for NLI.

#### 4.2.2 Out-of-Distribution Performance

Here we frame domain adaptation as a few-shot learning problem, where the adversarial evaluation set represents training examples from outside the seen domain of the classifier. To test out-of-distribution (OOD) performance on hate speech data, we use the HateCheck test suite (Röttger et al., 2021), which consists of test cases for 29 model functionalities relating to real-world concerns of stakeholders. For NLI we check OOD performance on the SNLI-Hard dataset (Gururangan et al., 2018), which assesses common model vulnerabilities.

## 5 Results

We discuss results for the TextFooler (TF) and HotFlip (HF) baselines along with our two model variations (NA-LIME and NA-IG).

**Quality and effectiveness of adversarial generations.** From Table 2, we can see that examples generated using NaturalAdversaries are perceived as more natural across domains (20% more for hate speech and 25% more for NLI). While attacking accuracy is generally lower than artificial adversaries, we also find that our black-box approach generalizes well to classifiers other than the original target model, sometimes matching or beating the performance of artificial baselines (notably, NA-LIME does 5% better on NLI for $Adv2$ than HotFlip).

**OOD performance.** Although NLI model performance is relatively unaffected by finetuning, when we assess the RoBERTa classifier using HateCheck, we find that the target model exhibits concerning vulnerabilities. Finetuning generally improves performance, though our NA-IG model leads to the most improvement (1.68 F1 over base performance). Given the small size of our evaluation set (150 examples), this indicates naturalistic adversarial examples may address classifier weaknesses, with minimal need for manual annotation. We provide examples of generations in Appendix D.

## 6 Related Work

**Adversarial Attacks** Prior work on adversarial attacks focus primarily on time-consuming and costly manual annotation (Kiela et al., 2021), or automatic example construction that relies upon a predefined type of attack (e.g. testing robustness to syntactic and lexical errors, Belinkov and Bisk, 2018; McCoy et al., 2019; Gabriel et al., 2021; Wu et al., 2021). The effect of complex adversarial attacks in the hate speech domain is also relatively unexplored. While Rusert et al. (2022) recently address this, they do not consider naturalness.

## 7 Conclusion

We introduce a framework for generation of naturalistic adversaries that is effective for multiple neural classifiers and across domains. We encourage further work on how naturalistic adversaries may improve robustness in real-world settings.

## 8 Limitations

While it is well-known that transformer-based language models suffer from lexical biases (Gururangan et al., 2018), it may be an oversimplification to say that a single keyword is independently the cause of a particular classification decision. It has been shown that language models may consider compositionality to some degree (Shwartz and Dagan, 2019; Baroni, 2019), and future work may explore explainability methods that take this into consideration (e.g., Ye et al., 2021). Another limitation is that generative approaches are highly dependent on the decoding method of choice (Holtzman et al., 2020), and while this provides us more flexibility, it also leads to more variability in performance.

## 9 Ethics & Broader Impact Statement

**General Statement** While there is a risk of any technologies aimed at mimicking natural language being used for malicious purposes, our work has wide-ranging potential societal benefit by improving fairness and real-world robustness of neural classifiers. Increasingly it has become clear that pretrained neural language models do not operate from a neutral perspective, and implicitly learn behaviors that pose real harm to users from training data (Jernite et al., 2022). We demonstrate that our framework is effective at generating adversaries that uncover model vulnerabilities for two well-studied domains (hate speech and NLI), and it is hypothetically extensible to other domains like automated fact-checking. Given the sensitive nature of toxic language and hate speech detection in particular, we strongly emphasize that the work is intended only for research purposes or improving robustness of automated systems. For data and code release, we include detailed model and data cards (Bender and Friedman, 2018; Mitchell et al., 2019).

**Annotation** Based on time estimates, the annotator wage is approx. $10-$16 per hour. All annotators were required to click a consent button before working on the tasks. For hate speech annotation, annotators were cautioned about the possibly disturbing nature of the content before being shown examples. We also provided crisis hot-line information in case of emotional distress.

## References

Marco Baroni. 2019. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B*, 375.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. *ICLR*.

Emily M. Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *ICML*.

Joy Buolamwini and Timnit Gebru. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *FAT*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. In *EMNLP*.

Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A. Smith, and Yejin Choi. 2022. Is gpt-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text. In *ACL*.

J. Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *ACL*.

Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. *Proceedings of the International AAAI Conference on Web and Social Media*, 12(1).

Saadia Gabriel, Asli Çelikyilmaz, Rahul Jha, Yejin Choi, and Jianfeng Gao. 2021. Go figure: A meta evaluation of factuality in summarization. In *ACL Findings*.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix Wichmann. Shortcut learning in deep neural networks. *Nat Mach Intell 2, 665–673*.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *NAACL*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. *ArXiv*, abs/2006.03654.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. *ICLR*.

Yacine Jernite, Huu Nguyen, Stella Biderman, Anna Rogers, Maraim Masoud, Valentin Danchev, Samson Tan, Alexandra Sasha Luccioni, Nishant Subramani, Isaac Johnson, Gerard Dupont, Jesse Dodge, Kyle Lo, Zeerak Talat, Dragomir Radev, Aaron Gokaslan, Somaieh Nikpoor, Peter Henderson, Rishi Bommasani, and Margaret Mitchell. 2022. Data governance in the age of large-scale data-driven language technology. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 2206–2222, New York, NY, USA. Association for Computing Machinery.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. 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. In *AAAI*.

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

Jey Han Lau, Carlos Armendariz, Shalom Lappin, Matthew Purver, and Chang Shu. 2020. How Furiously Can Colorless Green Ideas Sleep? Sentence Acceptability in Context. *Transactions of the Association for Computational Linguistics*, 8:296–310.

Alisa Liu, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. 2022. Wanli: Worker and ai collaboration for natural language inference dataset creation. In *Findings of EMNLP*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. In *AAAI*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 220–229, New York, NY, USA. Association for Computing Machinery.

John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020a. Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3829–3839, Online. Association for Computational Linguistics.

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

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1896–1906, Melbourne, Australia. Association for Computational Linguistics.

Subhabrata Mukherjee, Xiaodong Liu, Guoqing Zheng, Saghar Hosseini, Hao Cheng, Greg Yang, Christopher Meek, Ahmed Hassan Awadallah, and Jianfeng Gao. 2021. Clues: Few-shot learning evaluation in natural language understanding. *NeurIPS*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2018. RankME: Reliable human ratings for natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 72–78, New Orleans, Louisiana. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Candace Ross, Boris Katz, and Andrei Barbu. 2021. Measuring social biases in grounded vision and language embeddings. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 998–1008, Online. Association for Computational Linguistics.

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. 2021. HateCheck: Functional tests for hate speech detection models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 41–58, Online. Association for Computational Linguistics.

Jonathan Rusert, Zubair Shafiq, and Padmini Srinivasan. 2022. On the robustness of offensive language classifiers. In *ACL*.

Maarten Sap, D. Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *ACL*.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *ACL*.

Emily Sheng, Kai-Wei Chang, P. Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. *EMNLP*.

Vered Shwartz and Ido Dagan. 2019. Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, 7:403–419.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, D. Erhan, Ian J. Goodfellow, and Rob Fergus. 2015. Intriguing properties of neural networks. *IClR*.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021a. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1667–1682, Online. Association for Computational Linguistics.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021b. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *ACL/IJCNLP*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. *ICLR*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Tongshuang (Sherry) Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S. Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *ACL/IJCNLP*.

Xi Ye, Rohan Nair, and Greg Durrett. 2021. Connecting attributions and QA model behavior on realistic counterfactuals. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5496–5512, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. *ICLR*.

Xuhui Zhou, Maarten Sap, Swabha Swayamdipta, Yejin Choi, and Noah A. Smith. 2021. Challenges in automated debiasing for toxic language detection. In *EACL*.

# A Explainability Method Details

## A.1 Black-Box Setting

Given a classifier $F_c(x)$ and a set of initial seed examples $D$ with ground truth labels $y$, we predict classifier labels $\hat{y}$ and measure the contribution of each token $s_i$ in a given sequence $S \in D$ to classifier's prediction using LIME (Ribeiro et al., 2016). The LIME algorithm defines a local neighborhood around a point x representing S, $N(x)$, using a proximity measure $\pi_x$, and optimizes linear models $g \in G$ to jointly minimize the distance of decision functions $g$ and $F_c$ for $\tilde{x} \in N(x)$ and also the complexity of $g$. In this case:

$$\mathcal{L}(F_c, g, \pi_x) = \sum_{\tilde{x}, \tilde{x}' \in N(x)} \pi_x(\tilde{x})(f(\tilde{x}) - g(\tilde{x}'))^2 \tag{3}$$

$$F_{attr}(S) = \text{argmin}_{g \in G} \mathcal{L}(F_c, g, \pi_x) + \Omega(g) \tag{4}$$

where $\mathcal{L}(F_c, g, \pi_x)$ is the locality-aware loss, $\Omega(g)$ is model complexity, and $\pi_x$ is an exponential kernel defined using cosine distance[2]. We also tested Shapley additive explanation values (Lundberg and Lee, 2017) in early experimentation, but found that the results were less promising than LIME.

## A.2 White-Box Setting

Given a classifier $F_c(x)$ and a set of initial seed examples $D$ with labels $y$, we predict classifier labels $\hat{y}$ and measure the contribution of each token $S_i$ in an example sequence $S \in D$ to this final output decision using the following computation:

$$F_{attr}(S_i) = (x_i - x_i') \times \int_{\alpha=0}^{1} f(\alpha) d\alpha$$
$$f(\alpha) = \frac{\partial F_c(x' + \alpha \times (x - x'))}{\partial x_i}. \tag{5}$$

Here $x_i$ is the embedding of $S_i$. Following (Mudrakarta et al., 2018), the baseline input embedding $(x')$ is defined by a sequence of pad tokens that is the same length as the input, since the embedded pad tokens should not be informative. $\frac{\partial F(x)}{\partial x_i}$ is the gradient of $F_c(x)$ with respect to $x_i$ (Sundararajan et al., 2017). After identifying contribution of each token, we can partition $D$ into two sets $D_1$ and $D_2$ based on the model behavior, where $D_1$ forms a subset representing the space of correctly predicted examples and $D_2$ consists of incorrect predictions. We use the examples and attribution weights from $D_1$ as training data for a generative model $F_g(x)$.

# B Domains

## B.1 Hate Speech Detection

For hate speech detection, we train the AdversarialGen generation model on the DynaHate benchmark. (Vidgen et al., 2021b). We use a RoBERTa classifier trained on the twitter hate speech dataset (Founta et al., 2018; Zhou et al., 2021). DynaHate benchmark was chosen for training in our experiments since it was constructed using a human-and-machine-in-the-loop setup designed to reduce dataset biases and improve model generalizability. It also includes examples of implicit hate, which rely less on lexical cues. For this task, the input to the classifier model is a text document like the one shown below

x = [CLS] *all I want is to not be treated like a second-class citizen* [SEP].

Here $[CLS]$ and $[SEP]$ denote classifier-specific special tokens. The output is a binary label (benign or harmful).

## B.2 Natural Language Inference

For natural language inference, we train the generation model on the ANLI dataset (Nie et al., 2020), which was constructed similarly to DynaHate with an iterative human-and-machine-in-the-loop process. We test a DeBERTa-base classifier (He et al., 2021) trained on the Multi-Genre NLI (MNLI) corpus (Williams et al., 2018), which is a featured task in the CLUES and GLUE evaluation benchmarks (Mukherjee et al., 2021; Wang et al., 2019). The MNLI corpus consists of ∼433k diverse sentence pairs, however recent work has shown that MNLI-trained models are highly susceptible to adversarial attacks from crowd-source workers (Nie et al., 2020). For this task, the input to the classifier model is a premise sentence $s_p$ and hypothesis sentence $s_h$ like the ones shown below

$$\text{x = [CLS]} \underbrace{\textit{she walks behind me}}_{s_p} \text{[SEP]}$$
$$\underbrace{\textit{she walks in front of me}}_{s_h} \text{ [SEP]}.$$

The output is a label specifying whether the premise *contradicts* the hypothesis, *entails* the hypothesis or is *neutral* (the hypothesis could either be true or false given the premise).

[2] S is embedded using a simplified one-hot feature vector for g(x). In perturbed examples randomly selected tokens are masked.

# C Additional Implementation Details

For all explainability methods we took the top 20% of tokens with the highest attribution scores. For LIME, we set the maximum number of features to 20 and generate 2,000 samples for training.

All models are trained on a single Quadro RTX 8000 GPU. Training time averages 1 hour per epoch with a batch size of 32 for NLI data and 1 minute per epoch with a batch size of 16 for hate speech data. Inference time is approx. 10 minutes. We use the 124M parameter GPT-2 model for all generators. Generations are sampled after one training and evaluation run. We used a learning rate of 2e-4 for all models, determined by a hyperparameter search tuned on the dev. loss with the DynaHate trained model and potential learning rates in the range of $lr \in [.2, 2\text{e-}6]$. For the hate speech domain we use top-$k$ decoding with $k = 10$, while for natural language inference (NLI) we use beam search decoding with a beam size of 10 (we tested $k$ and beam size values from 3-10). We also use a max length context size of 150 tokens for hate speech and 500 for NLI. For both domains, we use a learning rate of $lr = .0004$ and train until early stopping on the dev. set.

We train the hate speech domain model using round 1 of DynaHate, and combine test sets from all rounds. We use round 3 for the NLI model and sample from all rounds for the test set.

## C.1 Data Processing

For human evaluation, we remove non-ASCII characters and special tokens like "<|endoftext|>." We also noticed that the HotFlip baseline tends to generate extra spaces, and removed unncessary spaces. Newline characters were replaced by "<linebreak>." All the data is in English.

## C.2 Annotator Instructions

**Hate speech** *Read a text fragment (potentially generated by an AI system) and tell us whether it is potentially toxic/harmful to any reader. Note that there is no right or wrong answer, and we are interested in your opinion. We use a broad definition of harmful speech, including but not limited to use of inappropriate language, insults, threats, stereotyping, hate speech, microaggressions, derogatory speech and obscene content.*

*Please look carefully at the text fragment and consider whether or not it is harmful. During each HIT you will be asked to answer several questions, listed and explained below.*

*1) Is the text fragment harmful/toxic? Our purpose is to understand how harm/toxicity can show up in language, we are not making statements about the actual content of the posts. Try to answer this question while considering a broad set of people from different backgrounds, not just your own.*

*2) Does the text fragment read naturally (Is it coherent, well-formed and likely human-written)?*

**NLI** *Read a premise text fragment and a hypothesis text fragment (potentially generated by an AI system) and tell us what type of relationship there is between the two text fragments (entailment, contradiction, neutral). We provide detailed definitions and examples for these three types of relationships below.*

*If you do not understand the text fragments, please indicate so using the given checkbox, but still answer the questions to the best of your abilities.*

***Entailment**: The two text fragments have this type of relationship if the information in the hypothesis follows from the premise (and no information contradicts the premise), e.g. the hypothesis "the woman is playing basketball" is entailed by the premise "the woman and the man are playing basketball in the park."*

***Contradiction**: The two text fragments have this type of relationship if at least some information in the hypothesis is contradicted by the premise, e.g. "the woman is playing soccer" is contradicted by the premise "the woman and the man are playing basketball in the park."*

***Neutral**: The two text fragments have this type of relationship if the information in the hypothesis is neither entailed by or contradicted by the premise (the two text fragments may be completely unrelated), e.g. the relationship between "the woman and the man are playing basketball in the park" and "the woman loves basketball" is neutral.*

***Questions:***

*1) What is the relationship (entailment, contradiction or neutral) between the two text fragments?*

*2) Do the premise and hypothesis text fragments read naturally (are they coherent, well-formed and likely human-written)?*

# D Generation Examples

We provide examples of generated examples in Table 3.

| **Example** | **Domain** | **Gold Label** |
|---|---|---|
| Kobo Town is a Canadian Caribbean music group, led by Trinidadian Canadian singer and songwriter Drew Gonsalves. Based in Toronto, Ontario... [SEP] **Kobo Town is based in Canada.** | **NLI** | 2 (Entailment) |
| Snake Handler<br>Clark went to church with his second cousin. Clark was shocked when the preacher held a poisonous snake...[SEP] **Clark is named after a preacher.** | **NLI** | 1 (Neutral) |
| "How to ask about someoneś disability<br>Consider whether the information is relevant. If itś not really relevant to you, then asking is probably rude...[SEP] **People with disabilities are generally brave.** | **NLI** | 1 (Neutral) |
| Iḿ glad that the vaccine is so popular, but I wonder if this would be a good thing for our democracy. | **Hate Speech** | 1 (Toxic/Hate) |
| I've always hated the word "feminists" because they are basically feminists, but I hate them for their views on women. | **Hate Speech** | 1 (Toxic/Hate) |

Table 3: Examples of NaturalAdversaries generations that were misclassified. For NLI only the hypothesis is generated, which we highlight in bold.