Malevolence Attacks Against Pretrained Dialogue Models

Pengjie Ren[1 [0000−0003−2964−6422]], Ruiqi Li[1 [0009−0008−8464−6162]], Zhaochun Ren[2 [0000−0002−9076−6565]], Zhumin Chen[1 [0000−0003−4592−4074]], Maarten de Rijke[3 [0000−0002−1086−0202]], and Yangjun Zhang[4⋆ [0009−0005−2663−5041]]

[1] Shandong University
[2] Leiden University
[3] University of Amsterdam
[4] Queen Mary University of London
renpengjie@sdu.edu.cn, 202215118@mail.sdu.edu.cn,
z.ren@liacs.leidenuniv.nl
chenzhumin@sdu.edu.cn, m.derijke@uva.nl, yangjun.zhang@qmul.ac.uk

Abstract. Pretrained language models (PLMs) are widely used by dialogue systems to generate high-quality responses. Adversarial samples can cause pretrained dialogue models to generate unexpected responses that influence the deployment of the system since malevolent responses can cause the breakdown of the dialogue. Exploring adversarial attacks is important to understand hidden risks and improve victim models. We propose the dialogue malevolence attack task and formulate it as a constrained Markov decision process. We also propose a two-stage reinforcement learning framework to find vulnerabilities of dialogue systems based on PLMs. Experiments show that the proposed attack framework can effectively attack pretrained dialogue models with high-quality adversarial samples. **Warning**: this paper contains examples that may be offensive or upsetting.

Keywords: Adversarial attack · Dialogue system · Large language model.

1 Introduction

Pretrained language models (PLMs) are increasingly employed by dialogue systems, and in particular, by conversational information retrieval systems [11] as they can generate high quality responses [2,50]. These models are pretrained on large-scale datasets that may contain negative views, bias, and other unsafe content [4,31]. The generation process is hard to control. Hence, current pretrained dialogue models are vulnerable to adversarial samples [18,32]. As current models may not truly understand human language [42], it is important to explore adversarial samples and adversarial attacks to reveal the safety weaknesses [34,35] of dialogue responses generated by victim pretrained dialogue models.

Malevolent dialogue responses that may contain offensive or objectionable content [49], have drawn broad attention. Malevolent responses may result in serious consequences for users, including dialogue breakdown, discomfort, and abandonment of the system [49]. E.g., for DialoGPT [50], a state-of-the-art dialogue model, around 10% of its responses are malevolent [48].

⋆ Corresponding Author

Table 1: Dialogue malevolence attack example.

Context	Turn 1: And before any funny prick decides to defend their own actions in my mentions i don't want to hear it. Turn 2: Lmao, you're just an absolute twat.	
Original	Turn 3: Why are we twats for having an opinion?	
Adversarial	Turn 3: Why do I always have stupid opinions?	
	Non-malevolent	**Malevolent**
Response	Because you're not funny.	Because you're stupid.

We present the *dialogue malevolence attack* (DMA), an adversarial attack task that we formulate as a constrained Markov decision process (CMDP) problem, as shown in Table 1. To address the dialogue malevolence attack task, we propose the *iteRative malevOlence atTAcker wiTh constraIned markOv decisioN* (ROTATION) model, illustrated in Figure 1, which modifies original utterances into adversarial ones that provoke victim models to generate malevolent responses. The *iterative attacker* (IA) for editing original utterances uses four operations: "K," "R," "D," and "I," for keeping, replacing, deleting, and inserting a token, respectively. Unlike existing methods, IA considers correlations among edits and can iteratively edit utterances until convergence. IA training employs reinforcement learning to interact with victim models. This raises two challenges. First, the editing reward for each position in the utterance is unknown; we learn the malevolence reward from the malevolence feedback of the whole utterance with a *reward evaluator* (RE). Second, the attack efficiency is low due to the computationally intensive nature of pretrained dialogue models; we devise a two-stage reinforcement learning strategy that includes an offline warm-up stage and an online fine-tuning stage to train ROTATION. During the offline stage, we warm up the parameters of the iterative attacker and reward evaluator networks with a pseudo trajectory generation process. During online learning, we use the iterative attacker to directly attack the victim models and update the iterative attacker through the reward evaluator.

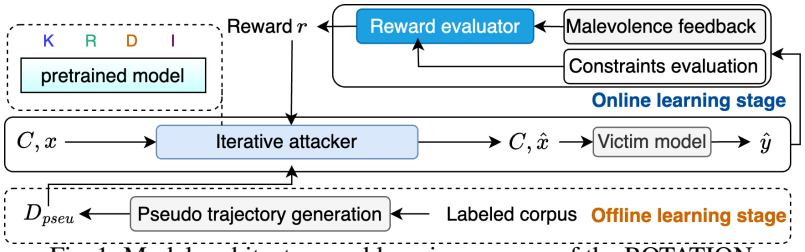


Fig. 1: Model architecture and learning process of the ROTATION.

We conduct experiments on two datasets: (i) the malevolent dialogue response detection and classification (MDRDC) dataset [49] with malevolence labels, and (ii) an unlabelled dialogue dataset that has been collected from Twitter and covers a diverse set of topics. Our experiments show that ROTATION can generate high-quality and more diverse adversarial samples that outperform the best baseline by 126.90% and 56.50% in terms of distinct attack success rate (DASR). We also conduct a human evaluation for ROTATION on the MDRDC dataset, the DASR outperforms the best baseline by 56.14%.

2 Related Work

Adversarial attacks. Adversarial attacks intentionally modify the input of a model with perturbations and use the modified inputs to change the outputs of victim models; they can be used to find vulnerabilities of victim models [25,28,45,47]. They come in two types: non-targeted and targeted. Non-targeted attacks attempt to trick victim models into producing wrong outputs without specific constraints on the wrong output [3,17,18,27,51]; e.g., for a classification model, the attacker tries to misguide it to predict any one of the incorrect classes [24,30]; for a generation model, the attacker tries to misguide it to generate any type unpredicted content [23,43]. Prior work on non-targeted attacks either does not consider the semantic consistency and coherence of the adversarial samples or does not target dialogue models. Targeted attacks aim to trick victim models into producing specific wrong outputs, which is more challenging than non-targeted attacks [6,22]. For existing methods on targeted attacks, the number of target words or responses is too small for the DMA task since the space of malevolent content is large. In addition, they edit the tokens of the original input only once without considering multiple edits.

Safety of pretrained language models. Pretrained language models may generate content that is not consistent with humanistic values [1,37]. [35] build a dataset to assess the safety of pretrained dialogue models and conclude that they all have risks of generating unsafe responses. [14] build a dialogue dataset; models pretrained on it may generate socially prejudiced responses. To understand what kind of inputs could increase unsafe generation, previous work focuses on input-agnostic attacks and input-specific attacks [20].

Input-agnostic attacks aim to fool victim models by adding learned universal noise, e.g., a sequence of tokens [12,41,44], to any input or by directly generating adversarial samples without considering the input [29,34]. ToxicBuddy [34] first generates an auxiliary dataset that has a high toxicity score on pretrained dialogue models, and then augments this to attack the pretrained dialogue models to generate more toxic responses. The input generated by ToxicBuddy for attacking a victim model is limited as it focuses on politics, which makes an attack relatively easy to detect as it is often neither coherent with the context nor consistent with the current utterance. ToxicBuddy does not interact with a target dialogue system; it is built for general attacks.

Input-specific attacks generate input-specific adversarial samples by making small changes to the original input [19,22,33]. Limited studies have been done for input-specific attacks because the sparse nature of input-specific attacks makes it hard to learn a model [5] and targeted input-specific attacks for safety-related aspects are easy to overfit without suitable feedback to evaluate the attack output of the victim model [8].

Our work differs from prior work in three ways: (i) We make the adversarial samples imperceptible by formulating dialogue malevolence attack as a constrained Markov decision process and modeling semantic consistency, fluency, and non-malevolency of adversarial samples as constraints. (ii) Most previous attack methods only demonstrate effectiveness on specific domains, where domain knowledge dominates the toxic behavior of dialogue systems. (iii) Our method generates adversarial samples by iteratively editing the input, and considers correlations between edits and the coherence with the dialogue context.

3 Methodology

3.1 Dialogue malevolence attack definition

Given a dialogue context C, current user utterance x, and victim model g, which is a pretrained dialogue model, *dialogue malevolence attack* (DMA) aims to learn an attack model f to perform a perturbation on x to obtain an adversarial version $\hat{x} = f(C, x)$ such that the g generates a malevolent response $\hat{y} = g(C, \hat{x})$ and satisfies these constraints: (i) $\hat{x}$ is consistent with x; (ii) $\hat{x}$ is fluent; and (iii) the attack model f does not introduce extra malevolence for $\hat{x}$ compared to x.

3.2 Constrained Markov decision process

We formulate dialogue malevolence attack as a constrained Markov decision process in which the iterative attacker tries to attack the victim model to generate targeted responses that maximize the malevolence reward and do not violate the constraints. The dialogue malevolence attack framework is shown in Figure 1. The constrained Markov decision process is defined as a tuple $\langle S, A, P, r_\psi, c_{1:3}, \alpha_{1:3}, \gamma \rangle$ where: (i) S is the state space, $s_t \in S$ is the current state that consists of the dialogue context and current user utterance, i.e., $s_t = (C, \hat{x}_t)$ where t is the number of times the original utterance x has been edited and $\hat{x}_t$ is an adversarial sample in t-th step; $\hat{x}_0 = x$. (ii) A is the action space, $a_t \in A$ is a sequence of editing operations that are independent of each other, i.e., $a_t = [a_{t,1}, a_{t,2}, \ldots, a_{t,n}]$, where n is the length of $\hat{x}_t$, $a_{t,j}$ is an editing operation for the j-th token of $\hat{x}_t$. (iii) $P : S \times A \times S \to [0, 1]$ is the transition function mapping a triple (s_t, a_t, s_{t+1}) to a probability by $P(s_t, a_t, s_{t+1}) = p(s_{t+1} \mid s_t, a_t)$, where $p(s_{t+1} \mid s_t, a_t)$ is the probability of transitioning from s_t to s_{t+1} under action a_t at time step t. The transitions are deterministic, once action a_t is selected, the next state s_{t+1} is known. (iv) r_ψ is the reward function and c_i is the i-th constraint function. And (v) γ is the discount factor $\in [0, 1]$.

The goal is to maximize the total expected malevolence reward while ensuring the constraints are under thresholds:

$$\max_\theta J_{r_\psi}(\pi_\theta) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(r_\psi(C, \hat{x}_t, a_t)\right)\right], \tag{1}$$

$$\text{subject to} \begin{cases} J_{c_1}(\pi_\theta) = \mathbb{E}\big[\sum\limits_{t=0}^{\infty} \gamma^t \left(c_1(x, \hat{x}_t)\right)\big] \geq \alpha_1, \\ J_{c_2}(\pi_\theta) = \mathbb{E}\big[\sum\limits_{t=0}^{\infty} \gamma^t \left(c_2(x, \hat{x}_t)\right)\big] \leq \alpha_2, \\ J_{c_3}(\pi_\theta) = \mathbb{E}\big[\sum\limits_{t=0}^{\infty} \gamma^t \left(c_3(x, \hat{x}_t)\right)\big] \leq \alpha_3, \end{cases} \tag{2}$$

where π_θ is the policy network, which is implemented by the iterative attacker; θ is the parameter of iterative attacker; r_ψ is the reward function, which is implemented by the reward evaluator; ψ are the parameters of reward evaluator; c_1, c_2, c_3 are the consistency, fluency, and non-malevolency cost functions, respectively; $\alpha_1, \alpha_2, \alpha_3$ are the pre-defined thresholds for the cost functions. The sum is over an infinite number of time steps, but we use finite horizons for t in practice.

The constraints are explained as follows: (i) $J_{c_1}(\pi_\theta)$ is the consistency constraint. To ensure that adversarial sample $\hat{x}_t$ is consistent with x, we embed x and $\hat{x}_t$ with a BERT model and use the cosine similarity between the BERT embeddings as the consistency score. (ii) $J_{c_2}(\pi_\theta)$ is the fluency constraint. We penalize actions leading to non-fluent outputs using the perplexity score [50] to measure the fluency of $\hat{x}_t$ by $-\frac{1}{n}\sum_{j=1}^{n}\log(P(w_j|w_{j-1}, w_{j-2}, \ldots, w_1))$, where n is the number of tokens in $\hat{x}_t$ and w_j is the j-th token. (iii) $J_{c_3}(\pi_\theta)$ is the non-malevolency constraint. To ensure that adversarial sample $\hat{x}_t$ does not introduce extra malevolence, we use the change of malevolence confidence score between x and $\hat{x}_t$ as the non-malevolency score, i.e., $S(\hat{x}_t) - S(x)$, where S is malevolence confidence score from a binary BERT-based malevolence classifier [49].

We apply Lagrange relaxation to approximate the constrained optimization problem [46] and aim to solve the following problem:

$$\min_{\lambda\geq 0}\max_{\theta} J_{r_\psi}(\pi_\theta) - \lambda_1(\alpha_1 - J_{c_1}(\pi_\theta)) - \lambda_2(J_{c_2}(\pi_\theta) - \alpha_2) - \lambda_3(J_{c_3}(\pi_\theta) - \alpha_3), \quad (3)$$

where $\lambda_1, \lambda_2, \lambda_3$ are the Lagarangian multipliers.

3.3 Iterative attacker

Unlike previous work, which divides sentence editing into token position prediction and operation prediction [26], our approach merges them into one with four editing operations:"K" for keeping a token, "R" for replacing, "D" for deleting, and "I" for inserting. We build an editing policy network, i.e., iterative attacker (IA), π_θ, based on BERT to generate a sequence of operations. The iterative attacker takes $(C, \hat{x}_t)$ as input and uses separator "[SEP]" after each utterance except the last one. We feed tokens into BERT to output representations H_t in the original utterance and apply a linear layer with parameter W_θ and b_θ to predict action a_t:

$$\pi_\theta(a_t|s_t = (C, \hat{x}_t)) = \mathrm{softmax}(W_\theta H_t + b_\theta), \quad (4)$$

where W_θ and b_θ are the parameters to be learned during the training process.

3.4 Reward evaluator

To solve the constrained Markov decision process problem and train iterative attacker with reinforcement learning, we need to get the reward for each editing operation, which is unknown. We only know the malevolence feedback of the final edited utterance $\hat{x}_t$, where the iterative attacker predicts "K" (keep($\hat{x}_t[j]$)) for all tokens of $\hat{x}_t$. Therefore, we learn a reward evaluator r_ψ parameterized by ψ to estimate the reward for each token. We take the hidden state H_t of BERT as input and feed them into the MLP to output a reward for each editing operation of the corresponding token position:

$$r_\psi(r|s = (C, \hat{x}_t), a = a_t) = \mathrm{sigmoid}(W_\psi H_t + b_\psi), \quad (5)$$

where W_ψ and b_ψ are the parameters to be learned during the training process.

3.5 Two-stage learning

To speed up training, we warm up the iterative attacker and reward evaluator with an offline learning stage by making use of the existing MDRDC corpus [49]. First, we train iterative attacker, reward evaluator, and λ with trajectories generated by a pseudo trajectory generation process. The parameters of iterative attacker and reward evaluator are passed to the online learning stage. Then, we use the iterative attacker to attack victim models and sample trajectories generated by the iterative attacker to update iterative attacker, reward evaluator, and λ with an online learning stage. To train the reward evaluator, its gradient is given by:

$$\bigtriangledown_{\psi} L(\lambda, \theta, \psi) = -\mathbb{E}_{\tau \in D^{+}}\big[\bigtriangledown_{\psi}\, r_{\psi}(\tau)\big] + \mathbb{E}_{\tau \in D^{-}}\big[\bigtriangledown_{\psi}\, r_{\psi}(\tau)\big], \tag{6}$$

where τ is a trajectory of state-action pairs from $\hat{x}_0$ to $\hat{x}_t$, i.e., $[(C, \hat{x}_0), a_0, (C, \hat{x}_1), a_1, \ldots, (C, \hat{x}_t), a_t, \ldots]$. We use pseudo trajectories as D^{+} and D^{-} in the offline learning stage and use actual trajectories as D^{+} and D^{-} in the online learning stage [10,52]. In practice, we treat malevolent trajectories as D^{+} and non-malevolency trajectories as D^{-}. Reward training by maximizing the distance between positive and negative trajectories may not provide enough guidance for the learning process. Therefore, we introduce a margin as an additional reward to better guide the editor in attacking dialogue models and subtract a minor offset δ from the rewards of all negative instances [13,15,39]. The gradient of the iterative attacker and λ parameters are:

$$\begin{aligned}
\bigtriangledown_{\theta} L(\lambda, \theta, \psi) = \mathbb{E}_{\tau \in D}(r_{\psi}(C, \hat{x}_t, a_t) + \lambda_1 c_1(x, \hat{x}_t) - \lambda_2 c_2(x, \hat{x}_t) \\
- \lambda_3 c_3(x, \hat{x}_t))\, \bigtriangledown_{\theta} \log \pi_{\theta}(C, \hat{x}_t, a_t), \\
\bigtriangledown_{\lambda} L(\lambda, \theta, \psi) = -\,\mathbb{E}_{\tau \in D}([-c_1(x, \hat{x}_t), c_2(x, \hat{x}_t), c_3(x, \hat{x}_t)]) + [-\alpha_1, \alpha_2, \alpha_3],
\end{aligned} \tag{7}$$

where $D = D^{+} \cup D^{-}$. In both stages, the parameters of iterative attacker, reward evaluator and λ are updated as:

$$\begin{aligned}
\theta_{k+1} &= \theta_k + \eta_1 \bigtriangledown_{\theta} L(\lambda_k, \theta_k, \psi_k), \\
\psi_{k+1} &= \psi_k + \eta_2 \bigtriangledown_{\psi} L(\lambda_k, \theta_k, \psi_k), \\
\lambda_{k+1} &= \lambda_k - \eta_3 \bigtriangledown_{\lambda} L(\lambda_k, \theta_k, \psi_k),
\end{aligned} \tag{8}$$

where η_1, η_2, η_3 are learning rates.

In the offline learning stage, we use a pseudo trajectory generation (PTG) process to generate pseudo trajectories D_{pseu}. D_{pseu} and a malevolence feedback signal, are used to train the parameters of reward evaluator, iterative attacker, and λ. The pseudo trajectory generation process is shown in Alg. 1. Γ denotes replacing each editing operation $a_{t'}$ with its opposite editing operation, e.g, if $a_{t'}$ is "D", $\Gamma(a_{t'})$ becomes "I". To improve the policy network training efficiency, we select a subset of available trajectories [21]. We vary the ratio of the number of malevolent and non-malevolent trajectories to achieve optimal performance; the effect of this ratio is discussed in §5.5. To train reward evaluator, the gradient of the reward evaluator is estimated by Eq. 6 except that we use pseudo trajectories D_{pseu} to replace D^{+} and D^{-}.

In the online learning stage, we train iterative attacker and reward evaluator based on the parameters learned in the offline stage and re-initialize λ. Similar to offline training,

Alg. 1: Pseudo trajectory generation

Input: Corpus $\{(C, \hat{x})\}$; M: maximum iteration editing steps; $\tilde{A}$: generation rule containing operations; $f_{mlm}(x)$: masked language model;
Output: Pseudo trajectories D_{pseu}.
for $(C, \hat{x}) \in Corpus$ **do**
  $m\ in\ [1, M]$;
  $\hat{x}_0 \leftarrow \hat{x}, \tau' \leftarrow \{\hat{x}_0\}, \tau \leftarrow \{\}$;
  for *step* $t' = \{1, \ldots, m\}$ **do**
    Randomly sample $a_{t'} \sim \tilde{A}$;
    $\hat{x}_{t'+1} = f_{mlm}(\hat{x}_{t'}, a_{t'})$;
    $\tau' \leftarrow \tau' \cup \{(C, a_{t'}), \hat{x}_{t'+1}\}$;
  for *step* $t' = \{m-1, \ldots, 0\}$ **do**
    $\tau \leftarrow \tau \cup \{(C, \hat{x}_{t'+1}), \Gamma(a_{t'})\}$
  $\tau \leftarrow \tau \cup \{(C, \hat{x}_0)\}$;
  D_{pseu}.append(τ);

we use reward evaluator to assign the reward for each editing operation and calculate malevolence feedback. The difference is that we sample actual trajectories D_{actu} during the online attacking process from the iterative attacker.

4 Experimental Setup

4.1 Datasets and baselines

We use two datasets: (i) the **MDRDC Dataset** has binary labels of malevolence and comprises 6,000 dialogues, encompassing a total of 28,510 utterances [49], which is an open-sourced dataset; and (ii) the **Unlabeled Dialogue Dataset** includes 6,000 unlabeled dialogues from Twitter for a total of 27,143 utterances, which is collected following Twitter policies.

We choose DialoGPT as the victim model. DialoGPT is a state-of-the-art large-scale pretrained dialogue response generation model for multi-turn conversations. To assess the potential impact of dialogue malevolence attack on larger pretrained models, we also test the performance of the DMA method that is trained with DialoGPT, on LLaMA-based [38] chatbots, i.e., Alpaca [36] and Vicuna [7]. We choose Vicuna-7B and Alpaca-7B as victim models.

We have three baseline models: (i) A naive word replacement baseline that randomly replaces words in the text with a dictionary of malevolent words[5] including "hate" and "offensive" which belong to the second-level malevolence categories of the MDRDC dataset [9,49]. We replace [1, 2, 3] positions in the text for all the datasets and 3 yields the best results. (ii) TDGPN [22], a reinforcement learning (RL)-based method to generate utterances that contain target curse words. To make it comparable with our model, we use all words in [9] as target words. (iii) ToxicBuddy [34], a benchmark for attacking dialogue systems to generate toxic responses. For a fair comparison, the model is fine-tuned with MDRDC.

[5] https://github.com/t-davidson/hate-speech-and-offensive-language

4.2 Implementation details

We implement Iterative attacker based on BERT with a vocabulary of 30,522 tokens. We set the max sequence length to 256, the hidden size to 768, and the edit vocabulary to 2,000. We use a two-layer MLP with hidden sizes [768, 768] for the modeling of the Reward evaluator module. α is set to [1.0, 5.834, 0.0] to ensure consistency with the original samples. We use the Adam optimizer to train all models. We use greedy decoding strategy for both ROTATION and ToxicBuddy, except in cases involving multiple attacks, and we use sampling decoding strategy for TDGPN. The models are trained with a maximum of 40 epochs on GeForce RTX 3090 GPUs.

4.3 Evaluation

We conduct both automatic and human evaluations to evaluate ROTATION. First, automatic evaluation concerns attack success rate, attack sample quality, and attack efficiency. Second, human evaluation concerns the former two dimensions.

For attack success rate, we report *redundant attack success rate* (RASR) and *distinct attack success rate* (DASR). RASR follows previous research [16,40], and DASR is used since the models tend to modify the original utterances into the same text and generate the same responses, which is effective but perceptible. RASR can not reflect the effectiveness considering duplication. RASR is defined as RASR $= \frac{N^{suc}}{N}$, where N^{suc} is the number of adversarial samples that provoke malevolent responses and N is the number of adversarial samples. We use a binary classifier trained on the MDRDC dataset to identify the malevolent response. We also propose *distinct attack success rate* (DASR), which is the attack success rate with duplicate adversarial samples corrected: DASR $= \frac{N^{cor}}{N}$.

For attack sample quality, we report the results of six metrics. The consistency, fluency, and non-malevolency follow Eq. 2. Distinct-1 and distinct-2 are used to define the diversity of the response. Coherency is used to evaluate whether the generated adversarial sample is coherent with the original utterance.

For attack efficiency, we report *Attack Times* (AT) and attack times (AT)-metrics curves. For each time of attack, different methods input queries to the model. With more attack times, the attack success rate increases. Attack times at a targeted attack success rate threshold (RASR reaches 95%) and the AT-metrics can evaluate the efficiency of the multiple attacks. For a single attack, we iteratively edit utterances at most M steps and feed the adversarial samples in the last step into a victim model. For multiple attacks, we edit the failed adversarial sample to get a new adversarial sample and leave the successful one unchanged. We continue this process till the RASR reaches 95%. Then, we calculate the number of times we feed the adversarial samples to a victim model.

We designed a questionnaire for the human evaluation for the MDRDC test dataset. First, we ask workers to label if the current user utterances are malevolent. Second, we ask workers to label if the attacking samples are malevolent, fluent, and consistent with the current user utterance. Third, we ask workers to label whether the generated response from the chatbot system is malevolent. We randomly choose 500 samples from the ToxicBuddy and ROTATION for labeling. All questionnaire answers are binary. In terms of quality control standards, we use a minimum of 500 HITs and a 97% approval rate for workers in Amazon MTurk. The calculation of human evaluation metrics is similar to automatic metrics.

Table 2: Two-stage single attack results. **Bold** denotes the best results except for the original utterances as input. Original denotes attacking with original utterances. Underline denotes the best results among all methods. $*$ denotes statistically significant improvement compared to all the baselines based on the t-test ($p < 0.05$).

Dataset	Method	DASR	RASR	Con	Flu	Non-mal	Dist-1	Dist-2	Coh
MDRDC	Original	9.18	9.24	1.0000	5.834	0.0000	0.0832	0.5091	0.7136
	Random	10.40	10.44	0.7199	7.560	0.3836	0.0738	**0.5513**	**0.7052**
	TDGPN	9.96	12.25	0.4840	6.576	0.3067	0.0762	0.3814	0.3930
	ToxicBuddy	12.53	**42.52**	0.5018	**4.701**	0.3711	0.0300	0.1356	0.6779
	ROTATION	**28.43***	29.38	**0.7873***	5.787	**0.2852***	**0.0770**	0.4211	0.6503
Unlabeled	Original	8.58	8.65	1.0000	5.281	0.0000	0.0996	0.5589	0.7225
	Random	9.90	9.93	0.7145	7.247	0.4084	0.0874	**0.5890**	**0.7169**
	TDGPN	10.82	14.47	0.4675	5.441	0.2852	0.0904	0.3997	0.4004
	ToxicBuddy	11.31	**43.59**	0.4925	**3.714**	0.3614	0.0392	0.1682	0.7031
	ROTATION	**17.70***	18.82	**0.7687***	5.331	**0.1377***	**0.0942**	0.4939	0.6663

Table 3: Two-stage multiple attack results on the MDRDC dataset when the attack time is 18. Bold face: the best performance.

Method	DASR	RASR	Con	Flu	Non-mal	Dist-1	Dist-2	Coh
Random	48.79	49.96	0.6016	8.122	0.6325	0.0545	0.4163	**0.6441**
TDGPN	60.46	78.48	0.4935	6.441	**0.4398**	**0.0629**	0.3575	0.4068
ToxicBuddy	78.25	81.07	0.5112	**5.768**	0.4623	0.0401	0.3260	0.3835
ROTATION	**93.87**	**95.22**	**0.6197**	6.884	0.5501	0.0457	**0.4465**	0.5243

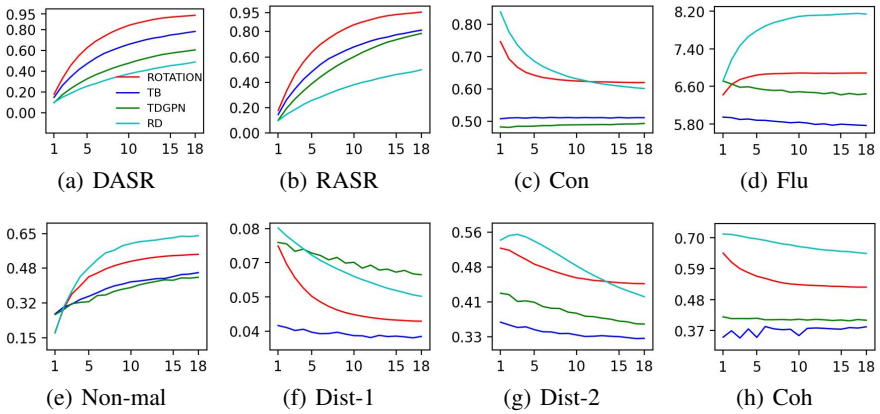
5 Results and Analyses

5.1 Overall performance

Performance for a single attack. As described in §4.3, a single attack denotes that we generate an adversarial sample once with at most M editing steps. The results are shown in Table 2.

First, across the two datasets, the DASR of ROTATION, which is trained with both offline and online stages, is the highest. The RASR score of ToxicBuddy is higher than that of ROTATION because ToxicBuddy tends to attack the pretrained model with duplicate adversarial samples. The higher DASR score indicates that ROTATION successfully attacks the pretrained model and outperforms baselines on attack success rate with lower duplication.

Second, across the two datasets, the quality metrics of ROTATION are comparable to the three baseline methods. On the MDRDC dataset and the unlabeled dialogue dataset, ROTATION outperforms all other methods in terms of consistency, non-malevolency, and distinct-1. ROTATION achieves the second-best results in terms of fluency and distinct-2. The reason why the fluency of ToxicBuddy is higher than that of ROTATION is that the fluency of the original dialogue utterance is high and ROTATION tends to satisfy constraints rather than dramatically improving the fluency scores of generated adversarial samples since we set the fluency threshold α_2 to 5.834, which is the fluency score of attacking with original utterances. The reason that the coherency of Random is



(a) DASR (b) RASR (c) Con (d) Flu

(e) Non-mal (f) Dist-1 (g) Dist-2 (h) Coh

Fig. 2: Performance of TDGPN, ToxicBuddy (TB), Random (RD), and ROTATION (with two-stage learning) over different attack times (x-axis) on the MDRDC dataset. For Flu, lower values are better.

higher than that of the ROTATION is that the GRADE score is a keyword-based metric and Random would add new keywords into the original utterance. The results indicate that the quality of adversarial samples from ROTATION drops a bit with regard to the original utterances, but the fluency is better than the original utterances. The reason why the quality metrics of original utterances are slightly superior to ROTATION is that ROTATION is set to satisfy all the constraints with the threshold settings with the score of original utterances, and during training the model reaches a balance of attack success rate and adversarial sample quality.

Performance for multiple attacks. To evaluate the attacking efficiency, we attack the victim model multiple times on the MDRDC dataset and end the attack when the RASR of ROTATION achieves 95% where AT $= 18$ to investigate how the attack performance changes with increasing attack times. For all models, we use a sampling decoding strategy. The results are shown in Figure 2 and Table 3.

ROTATION has better attack efficiency than ToxicBuddy and TDGPN with acceptable quality. First, ROTATION consistently achieves higher attack success rates at all levels of attack times. The improvement of ROTATION in terms of attack quality and efficiency can be attributed to the consideration of the correlation among edits and the use of constraints to enhance the overall quality.

5.2 Ablation studies

Ablation analyses of two-stage learning. To evaluate the impact of two-stage learning, we report the results of ROTATION with only offline learning or only online learning and compare them with ROTATION in Table 4.

First, the results indicate that incorporating online learning benefits both DASR and RASR, while the results of other metrics indicate that incorporating online learning slightly decreases the quality of adversarial samples, but the decrease is acceptable. Compared with ROTATION with two-stage learning, removing online learning decreases the DASR and RASR by 57.09% and 58.07%, respectively; the consistency, fluency,

non-malevolency, dictinct-2 and coherency increase by 5.35%, 5.50%, 8.41% and 4.20%, respectively; non-malevolency and distinct-1 decrease by 56.31% and 1.17%, respectively. Second, the results show that incorporating offline learning benefits both DASR and RASR, as well as the quality of adversarial samples. Removing offline learning decreases the DASR and RASR by 80.48% and 81.11%; the fluency increases by 52.86%; consistency, non-malevolency, distinct-1, dictinct-2, coherency decrease by 32.82%, 32.64%, 71.95%, 21.11%, and 29.51%, respectively.

Table 4: Ablation studies of two-stage learning on the MDRDC dataset. Boldface and underlining indicate the best results in each group. For Flu, lower values are better.

Ablation setting	DASR	RASR	Con	Flu	Non-mal	Dist-1	Dist-2	Coh
ROTATION	**28.43**	**29.38**	0.7873	**5.787**	0.2852	**0.0770**	0.4211	0.6503
- online	12.20	12.32	**0.8294**	6.105	**0.1246**	0.0761	**0.4565**	**0.6776**
- offline	5.55	5.55	0.5289	8.846	0.1921	0.0216	0.3322	0.4584

5.3 Attacking larger victim models

We also conduct experiments to see if ROTATION trained by attacking DialoGPT could help to improve the success rate when attacking larger victim models. To this end, we compare ROTATION trained by attacking DialoGPT with the original utterances when attacking two LLaMA-based victim models, i.e., Alpaca and Vicuna. The results are listed in Table 5. Compared with attacking DialoGPT, the inferior performance of ROTATION on Alpaca and Vicuna can be attributed to two factors: the significant disparity in parameter sizes and the absence of training specifically on Alpaca and Vicuna. The parameter size of Alpaca and Vicuna is ten times larger than that of DialoGPT, resulting in increased complexity and potential difficulties in executing attacks.

5.4 Human evaluation

We conduct a human evaluation to evaluate the performance of ROTATION, and we report the results in Table 6. The results indicate that the ROTATION method has better performance in attacking the pretrained dialogue model to generate malevolent responses, and ensures the quality of adversarial samples at the same time. Compared with ToxicBuddy, the DASR, consistency, and fluency of ROTATION increase by 56.14%, 60.10%, and 9.09% respectively and RASR and non-malevolency decreases by 22.84% and 6.47% respectively. The trend is similar to the automatic results in Table 2.

5.5 Hyperparameter analyses

Effect of maximum iteration editing steps. To study the influence of maximum number of iteration editing steps M, we report the results of different values of M for ROTATION. First, when M increases, the attack success rate first increases and then decreases. When $M = 2$, the model gets the best attack success rate. With overly large M, the pseudo trajectory generation module excessively edits the original dialogue utterance and decreases the attack success rate; with smaller M, editing is not enough to achieve successful attacks. Second, when M increases, the quality first decreases and

Table 5: Attacking Alpaca and Vicuna for MDRDC dataset. ROTATION uses two-stage learning and the quality metrics are the same as in Table 2

Metrics	DASR		RASR	
Model	*Alpaca*	*Vicuna*	*Alpaca*	*Vicuna*
Original	10.35	12.96	10.35	13.03
ROTATION	14.58	14.40	14.77	14.91

Table 6: Human evaluation results on the MDRDC dataset. "TB" is short for ToxicBuddy; "ROT" is short for "ROTATION."

Method	TB	ROT
DASR-h	0.228	**0.356**
RASR-h	**0.464**	0.358
Consistency	0.396	**0.634**
Fluency	0.572	**0.624**
Non-malevolency	0.278	**0.260**

Table 7: Results of ROTATION on the MDRDC dataset (with margin).

	DASR	RASR	Con	Flu	Non-mal	Dist-1	Dist-2	Coh
1:0	28.43	29.38	0.7873	5.787	0.2852	**0.0770**	0.4211	0.6503
1:0.01	**31.09**	**33.84**	0.7737	5.674	0.3228	**0.0770**	0.4279	0.6411
1:0.1	28.40	30.46	0.7686	**5.562**	0.3106	0.0769	0.4143	0.6410
1:1	9.98	10.03	**0.8124**	6.720	**0.0620**	0.0765	**0.4600**	**0.6764**

then increases. The reason is that the model maintains a trade-off between the attack success rate of adversarial attacks and the quality of the generated adversarial samples. As the attack success rate increases, the quality of adversarial samples tends to decrease.

Effect of the ratio of malevolent and non-malevolent trajectories. We modify the ratio of malevolent and non-malevolent trajectories ($D^+ : D^-$) during training to explore how it affects performance. First, as the ratio decreases, the attack success rate decreases. Non-malevolent trajectories are not very helpful in demonstrating how to successfully attack the victim models. Second, with a decrease in the ratio, the consistency of adversarial samples increases, and the non-malevolency of adversarial samples decreases. Compared with a ratio of "1:0", training with a ratio of "1:0.01", "1:0.01", and "1:1" has a better quality of adversarial samples. Non-malevolent trajectories may exhibit lower rewards for most editing operations, thereby pushing the optimization process towards satisfying constraints first.

Effect of adding margin score to the reward. Since the ratio of malevolent and non-malevolent trajectories with "1:0" has the best attack success rate, we assume that non-malevolent trajectories are not very helpful in training; the reward maximizing the distance between positive and negative trajectories does not provide perfect guidance for the training process. To improve the reward, we add reward margins to the reward function of positive and negative trajectories. We consider [0.15, 0.2, 0.25] for the offset δ and choose the best result. Table 7 shows the results of adding a margin to the reward. It improves the attack success rate of ROTATION by providing better guidance for the non-malevolent trajectories. First, with a margin added, compared with ratio "1:0", the ratios of "1:0.01" and "1:0.1" achieve better DASR, RASR, indicating enhanced dialogue malevolence attack ability. Compared with the best result of no margin (ratio "1:0"), adding the margin score improves the results. The improvement in attack success

rate comes with a decline in consistency, non-malevolency, and coherency; fluency, distinct-1, and distinct-2 do not decline.

6 Conclusion and Future Work

We have introduced the dialogue malevolence attack task, formulated it as a constrained Markov decision process problem, and proposed ROTATION, a pseudo trajectory generation-based two-stage reinforcement framework to identify pretrained dialogue model vulnerabilities and generate high-quality adversarial samples. ROTATION uses an iterative attacker considering the correlation between edits. The effectiveness and imperceptible adversarial sample generation ability of our method are confirmed through extensive experiments and analyses.

Our work confirms that a proportion of utterances with small perturbations would trigger malevolent responses that may cause great harm to users. The attack model helps understand the vulnerability of victim models and provides a basis for developing defense mechanisms. The goal of our work is not to attack existing systems but to discover the possible malevolent content generation risks of pretrained dialogue models and provide improvement hints for the deployment of safe dialogue system applications. This work can be used to discover potential malevolence risk instances; these instances can be used to enhance the training effect of the malevolent dialogue detection model and to detect whether a model reply is safe in the actual system. Nevertheless, the outcomes of this work could be misused to produce malevolent content in online dialogue systems and harm users. We believe the risks are outweighed by the benefits as there have been similar studies in the past [22,34].

In the future, we plan to attack victim models with evaluation based on more fine-grained classifiers. We will also work on the creation of defense mechanisms to fix the vulnerability of the victim models.

Reproducibility

The code and the Unlabeled Dialogue Dataset used to obtain the results are available at https://github.com/ruiqili001/DMA. The binary classifier trained on the MDRDC dataset is available at https://huggingface.co/mjmm/mal_classifier. The MDRDC dataset is available at https://github.com/repozhang/malevolent_dialogue.

Acknowledgments

This research was (partially) supported by the Key R&D Program of Shandong Province with grant 2024CXGC010108, the Natural Science Foundation of China (62472261, 62102234, 62372275, 62272274, 62202271, T2293773, 62072279), the National Key R&D Program of China with grant No.2022YFC3303004, the Dutch Research Council (NWO), under project numbers 024.004.022, NWA.1389.20.183, and KICH3.LTP.20.006, and the European Union's Horizon Europe program under grant agreement No 101070212, and by the EPSRC grant EP/W032473/1 (AP4L: Adaptive PETs to Protect & emPower People during Life Transitions). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

1. Abid, A., Farooqi, M., Zou, J.: Large language models associate muslims with violence. Nature Machine Intelligence **3**(6), 461–463 (2021)
2. Azzopardi, L., Clarke, C.L.A., Kantor, P., Mitra, B., Trippas, J.R., Ren, Z., Aliannejadi, M., Arabzadeh, N., Chandrasekar, R., de Rijke, M., Eustratiadis, P., Hersh, W., Huang, J., Kanoulas, E., Kareem, J., Li, Y., Lupart, S., Mekonnen, K.A., Roegiest, A., Soboroff, I., Silvestri, F., Verberne, S., Vos, D., Yang, E., Zhao, Y.: Report on the Search Futures workshop at ECIR 2024. SIGIR Forum **58**(1) (June 2024)
3. Belinkov, Y., Bisk, Y.: Synthetic and natural noise both break neural machine translation. International Conference on Learning Representations (2018)
4. Bertsch, A., Oh, A., Natu, S., Gangu, S., Black, A.W., Strubell, E.: Evaluating gender bias transfer from film data. In: Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP). pp. 235–243 (2022)
5. Biju, E., Sriram, A., Kumar, P., Khapra, M.M.: Input-specific attention subnetworks for adversarial detection. In: Findings of the Association for Computational Linguistics: ACL 2022. pp. 31–44 (2022)
6. Cheng, M., Yi, J., Chen, P.Y., Zhang, H., Hsieh, C.J.: Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 3601–3608 (2020)
7. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality (March 2023), https://lmsys.org/blog/2023-03-30-vicuna/
8. Co, K.T., Muñoz-González, L., de Maupeou, S., Lupu, E.C.: Procedural noise adversarial examples for black-box attacks on deep convolutional networks. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 275–289 (2019)
9. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language. In: Proceedings of the International AAAI conference on Web and Social Media. vol. 11, pp. 512–515 (2017)
10. Finn, C., Levine, S., Abbeel, P.: Guided cost learning: Deep inverse optimal control via policy optimization. In: Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016. pp. 49–58 (2016)
11. Gao, J., Xiong, C., Bennett, P., Craswell, N.: Neural Approaches to Conversational Information Retrieval. Springer (2023)
12. Gehman, S., Gururangan, S., Sap, M., Choi, Y., Smith, N.A.: Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 3356–3369 (2020)
13. Grzes, M., Kudenko, D.: Theoretical and empirical analysis of reward shaping in reinforcement learning. In: 2009 International Conference on Machine Learning and Applications. pp. 337–344 (2009)
14. Gu, Y., Wen, J., Sun, H., Song, Y., Ke, P., Zheng, C., Zhang, Z., Yao, J., Liu, L., Zhu, X., Tang, J., Huang, M.: EVA2.0: Investigating open-domain chinese dialogue systems with large-scale pre-training. Machine Intelligence Research pp. 1–13 (2023)
15. Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., Meger, D.: Deep reinforcement learning that matters. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. AAAI Press (2018)

16. Huang, Z., Zhang, T.: Black-box adversarial attack with transferable model-based embedding. In: International Conference on Learning Representations (2019)
17. Iyyer, M., Wieting, J., Gimpel, K., Zettlemoyer, L.: Adversarial example generation with syntactically controlled paraphrase networks. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 1875–1885 (2018)
18. Jia, R., Liang, P.: Adversarial examples for evaluating reading comprehension systems. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 2021–2031 (2017)
19. Li, L., Ma, R., Guo, Q., Xue, X., Qiu, X.: Bert-attack: Adversarial attack against bert using bert. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 6193–6202 (2020)
20. Li, M., Yang, Y., Wei, K., Yang, X., Huang, H.: Learning universal adversarial perturbation by adversarial example. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 1350–1358 (2022)
21. Li, Z., Kiseleva, J., de Rijke, M.: Dialogue generation: From imitation learning to inverse reinforcement learning. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019. pp. 6722–6729 (2019)
22. Liu, H., Wang, Z., Derr, T., Tang, J.: Chat as expected: Learning to manipulate black-box neural dialogue models. arXiv preprint arXiv:2005.13170 (2020)
23. Liu, J., Nogueira, M., Fernandes, J., Kantarci, B.: Adversarial machine learning: A multi-layer review of the state-of-the-art and challenges for wireless and mobile systems. IEEE Communications Surveys & Tutorials **24**(1), 123–159 (2021)
24. Liu, S., Lu, N., Chen, C., Tang, K.: Efficient combinatorial optimization for word-level adversarial textual attack. IEEE/ACM Transactions on Audio, Speech, and Language Processing **30**, 98–111 (2021)
25. Liu, Y.A., Zhang, R., Guo, J., de Rijke, M., Fan, Y., Cheng, X.: Multi-granular adversarial attacks against black-box neural ranking models. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 1391–1400. Association for Computing Machinery (2024)
26. Liu, Z., Ren, P., Chen, Z., Ren, Z., de Rijke, M., Zhou, M.: Learning to ask conversational questions by optimizing levenshtein distance. 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). pp. 5638–5650 (2021)
27. Niu, T., Bansal, M.: Adversarial over-sensitivity and over-stability strategies for dialogue models. CoNLL 2018 p. 486 (2018)
28. Parry, A., Fröbe, M., MacAvaney, S., Potthast, M., Hagen, M.: Analyzing adversarial attacks on sequence-to-sequence relevance models. In: European Conference on Information Retrieval. pp. 286–302. Springer (2024)
29. Perez, E., Huang, S., Song, F., Cai, T., Ring, R., Aslanides, J., Glaese, A., McAleese, N., Irving, G.: Red teaming language models with language models. arXiv preprint arXiv:2202.03286 (2022)
30. Raj, C., Mukherjee, A., Purohit, H., Anastasopoulos, A., Zhu, Z.: Salsa: Salience-based switching attack for adversarial perturbations in fake news detection models. In: European Conference on Information Retrieval. pp. 35–49. Springer (2024)
31. Schick, T., Udupa, S., Schütze, H.: Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in NLP. Transactions of the Association for Computational Linguistics **9**, 1408–1424 (2021)
32. Shah, D.S., Schwartz, H.A., Hovy, D.: Predictive biases in natural language processing models: A conceptual framework and overview. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 5248–5264 (2020)

33. Shen, L., Zhang, X., Ji, S., Pu, Y., Ge, C., Yang, X., Feng, Y.: Textdefense: Adversarial text detection based on word importance entropy. arXiv preprint arXiv:2302.05892 (2023)
34. Si, W.M., Backes, M., Blackburn, J., De Cristofaro, E., Stringhini, G., Zannettou, S., Zhang, Y.: Why so toxic? measuring and triggering toxic behavior in open-domain chatbots. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. p. 2659–2673. CCS '22, Association for Computing Machinery (2022)
35. Sun, H., Xu, G., Deng, J., Cheng, J., Zheng, C., Zhou, H., Peng, N., Zhu, X., Huang, M.: On the safety of conversational models: Taxonomy, dataset, and benchmark. arXiv preprint arXiv:2110.08466 (2021)
36. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Stanford Alpaca: An instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca (2023)
37. Thoppilan, R., Freitas, D.D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.T., Jin, A., Bos, T., Baker, L., Du, Y., Li, Y., Lee, H., Zheng, H.S., Ghafouri, A., Menegali, M., Huang, Y., Krikun, M., Lepikhin, D., Qin, J., Chen, D., Xu, Y., Chen, Z., Roberts, A., Bosma, M., Zhao, V., Zhou, Y., Chang, C.C., Krivokon, I., Rusch, W., Pickett, M., Srinivasan, P., Man, L., Meier-Hellstern, K., Morris, M.R., Doshi, T., Santos, R.D., Duke, T., Soraker, J., Zevenbergen, B., Prabhakaran, V., Diaz, M., Hutchinson, B., Olson, K., Molina, A., Hoffman-John, E., Lee, J., Aroyo, L., Rajakumar, R., Butryna, A., Lamm, M., Kuzmina, V., Fenton, J., Cohen, A., Bernstein, R., Kurzweil, R., Aguera-Arcas, B., Cui, C., Croak, M., Chi, E., Le, Q.: LaMDA: Language models for dialog applications. arXiv preprint arXiv:2201.08239 (2022)
38. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: LLaMA: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
39. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Canton-Ferrer, C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P.S., Lachaux, M., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E.M., Subramanian, R., Tan, X.E., Tang, B., Taylor, R., Williams, A., Kuan, J.X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., Scialom, T.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
40. Tsai, Y.T., Yang, M.C., Chen, H.Y.: Adversarial attack on sentiment classification. In: Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. pp. 233–240 (2019)
41. Wallace, E., Feng, S., Kandpal, N., Gardner, M., Singh, S.: Universal adversarial triggers for attacking and analyzing nlp. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 2153–2162 (2019)
42. Wallace, E., Rodriguez, P., Feng, S., Yamada, I., Boyd-Graber, J.: Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. Transactions of the Association for Computational Linguistics **7**, 387–401 (2019)
43. Wang, J., Bao, R., Zhang, Z., Zhao, H.: Rethinking textual adversarial defense for pre-trained language models. IEEE/ACM Transactions on Audio, Speech, and Language Processing **30**, 2526–2540 (2022)
44. Yu, D., Sagae, K.: Automatically exposing problems with neural dialog models. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 456–470 (2021)

45. Zhang, P.F., Huang, Z., Bai, G.: Universal adversarial perturbations for vision-language pre-trained models. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 862–871. Association for Computing Machinery (2024)
46. Zhang, R., Yu, T., Shen, Y., Jin, H., Chen, C., Carin, L.: Reward constrained interactive recommendation with natural language feedback. In: NeurIPS (2020)
47. Zhang, W.E., Sheng, Q.Z., Alhazmi, A., Li, C.: Adversarial attacks on deep-learning models in natural language processing: A survey. ACM Transactions on Intelligent Systems and Technology **11**(3), 1–41 (2020)
48. Zhang, Y.: Malevolent Dialogue Response Detection and Evaluation. Ph.D. thesis, University of Amsterdam (2022)
49. Zhang, Y., Ren, P., de Rijke, M.: A taxonomy, data set, and benchmark for detecting and classifying malevolent dialogue responses. Journal of the Association for Information Science and Technology **72**(12), 1477–1497 (2021)
50. Zhang, Y., Sun, S., Galley, M., Chen, Y.C., Brockett, C., Gao, X., Gao, J., Liu, J., Dolan, W.B.: DIALOGPT: Large-scale generative pre-training for conversational response generation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 270–278 (2020)
51. Zhao, Z., Dua, D., Singh, S.: Generating natural adversarial examples. In: International Conference on Learning Representations (2018)
52. Ziebart, B.D., Maas, A.L., Bagnell, J.A., Dey, A.K.: Maximum entropy inverse reinforcement learning. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008. pp. 1433–1438 (2008)