How to Keep an Online Learning Chatbot From Being Corrupted

Yixuan Chai
School of Computer Science and Technology
Donghua University
Shanghai,China
chaiyixuan@mail.dhu.edu..cn

Guohua Liu
School of Computer Science and Technology
Donghua University
Shanghai, China
ghliu@dhu.edu.cn

Ziwei Jin
Department of Computer Science and Engineering
Ohio State University
Columbus, United States
Jin.517@osu.edu

Donghong Sun
Institute for Network Sciences and Cyberspace
Tsinghua University
Beijing, China
sundonghong@tsinghua.edu.cn

***Abstract*—Online learning can improve chatbots' conversational abilities. Although the online learning method has enhanced the diversity of chatbots' statements, it also brings opportunities for corruption. The chatbot may be corrupted to generate offensive responses such as racist and hate speech. The key component to keeping chatbots from being corrupted is offensive-response detection. Until now, the training datasets for offensive detection have focused only on individual response sentences, disregarding user input sentences. In this paper, we introduce a dialogue-based offensive-response dataset, which consists of 110K input-response chat records. The dataset fills the gap in response detection for chatbots. Then, we build two challenging tasks based on the dataset: an offensive-response detection task and a corrupted chatbot purification task. In addition, we propose a strong benchmark method for the tasks: an encoder-classifier model to detect input-response pairs and a one-shot reinforcement learning (RL) method to reduce rapidly the probability of generating offensive responses.**

Keywords—offensive response, online learning chatbot, reinforcement learning

I. Introduction

Sequence-to-sequence (Seq2Seq) models[1]–[3] offer great promise for dialogue generation but often generate dull responses [4]. One of the reasons is the lack of utterance diversity in the training corpus. To address this problem, researchers integrate online learning into dialogue systems [4], which allows chatbots to have the ability to learn from online human conversations (i.e., human-in-the-loop). However, in practical applications, some users may take advantage of the online learning interface to generate inappropriate responses, such as racist and hate speech. For example, within hours of Microsoft's chatbot Tay [5] went online, some users took advantage of flaws in Tay's algorithm to make the artificial intelligence (AI)-based chatbot respond to certain questions with racist answers [6].

The key component to keeping chatbots from being corrupted is offensive-response detection. Until now, the training datasets of offensive detection focused only on individual response sentence, disregarding the input of the user's input sentence. These datasets include YouTube movie comment-based[7] and Twitter-based[8] offensive-response detection datasets.

However, the user input is important for offensive detection in some situations (e.g., the response "He is a hero" is an offensive response when the user inputs "How about Hitler?"). In this paper, we introduce a dialogue-based offensive-response dataset, which consists of 110K input-response chat records.

Then, we build two challenging tasks based on the dataset: an offensive-response detection task and a corrupted chatbot purification task. The offensive-response detection task detects whether the input-output pair is offensive. The corrupted chatbot purification task makes the corrupted chatbot forget the offensive response learned previously. In addition, we propose strong benchmark methods for the tasks: a recurrent neural network-based model to detect input-response pairs and a one-shot reinforcement learning (RL) method to reduce rapidly the probability of generating offensive responses.

In conclusion, the contributions of our paper are as follows:

- A dialogue-based offensive dataset is proposed. The dataset consists of 110K input-response chat records. The existing datasets focus only on individual response sentences, disregarding the user inputs. The proposed dataset fills the gap in offensive response detection of chatbot.
- We build two challenging tasks based on the dataset.: offensive-response detection task and corrupted chatbot purification task. The offensive-response detection task detects whether the input-output pair is offensive. The corrupted chatbot purification task makes the corrupted chatbot forget the offensive response learned previously. In addition, we concluded the challenge of these tasks which were not considered in previous works.

This research has been financially supported by grants from the National Key R&D Program of China (No.2017YFB0309800) and National Key Research and Development Project (Grant No. 2016QY12Z2103-2).

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

- We also provided a strong benchmark method for these tasks. For the first task, a novel encoder-classifier architecture is proposed for the task of offensive response detection. The model is greatly improved compared to directly concatenating the input sentence and response sentence for classification. The architecture eases the long-dependency problem of the RNN-based model. For the second task, a one-shot RL method is proposed. The method can quickly forget the offensive response with less impact on the basic conversational skills learned previously.

The dataset and source code are available online.[1]

II. Related Work

A. Online Learning Chatbot

Online learning allows chatbot to have the ability to learn conversations from humans, which can enrich the diversity of statements though a continuous learning process. Li et al.[9] proposed a framework that can learn from the online feedback from humans. Numerical feedback is delivered to the chatbot by the RL method, and the authors made use of forward prediction methods to handle textual feedback. Asghar et al.[4] proposed an online one-shot learning model. Users can provide feedback to the chatbot by suggesting a response. The feedback immediately becomes the chatbot's most likely predicted response for that prompt (one-shot learning). These models have a common defect: people may be take advantage of these fast and unrestricted learning abilities to teach online learning chatbots to generate offensive responses.

B. Offensive Statement Detection

Offensive statement detection can be simply cast to text classification tasks or sentiment analysis tasks[10]–[13]. Ravi [14] and Zhang [15] provide a review on deep learning algorithms in sentiment analysis. Specifically, for offensive detection task, Allouch [16] introduced a dataset which contains sentences that may be harmful to children, and proposed a voting method using several classifiers for detection. Razavi [17] proposed a multi-level Bayes offensive classifier detects features at different conceptual levels and so on [18]–[20].

There are few works on offensive responses detection of chatbot. Chkroun [21] proposed a safe collaborative chatbot called Safebot. The Safebot uses a malicious dataset to store the responses that were injected by users tagged as malicious. During the 'learning state', Safebot searches the malicious dataset to determine which response is closest to the newly taught response. if an entry in the malicious dataset is determined as closest, Safebot refrains from learning the new response and warns the user. In our previous work[6], we introduce a reinforcement leaning method to reduce the probability of offensive response generation of chatbot.

However, the above methods detect only the individual response sentence and disregarding the user's input sentence. Sometimes the same response of chatbot results in opposite sentiment for different input sentence. Examples shown in Table I.

TABLE I. Examples of offensive responses

Input	Response	Class
What do you think of Jay?	He is an idiot.	Offensive Words
What about Lee?	He looks like a monkey.	Offensive Semantics
What do you think of Hitler?	He is a hero.	Inopportune Response
What do you think about Martin Luther King?	He is a hero.	Normal Responses
What do you think of Hitler?	Terrible.	Normal Responses
What do you think about Martin Luther King?	Terrible.	Inopportune Responses

III. Offensive Responses

To analyze the dataset clearly, we have created the following classifications according to the form of the response: offensive words, offensive semantics and inopportune responses. Examples are shown in Table I.

Offensive words: There are explicit profane words in the response sentence. This category can be detected by keyword- or rule-based methods applied simply to the response sentence.

Offensive semantics: There are no explicit profane words in the response sentence, but the semantics of the sentence are offensive. This category can be detected by semantic-based machine learning methods on the response sentence.

Inopportune response: There are no explicit offensive words or semantics in the response sentence, but it is offensive if the context of the input is considered. In other words, it will become a normal response when the input context changes. For example, from Table I., we can see that the response "He is a hero" becomes offensive when the input sentence changes from "What do you think about Martin Luther King?" to "What do you think of Hitler?".

IV. Offensive Response Dataset

Until now, datasets for offensive-response detection have focused only on individual response sentences, disregarding the user's input. We can conclude from Table I that the input sentence sometimes has a decisive influence on the offensive-detection results. To fills the gap in offensive-response detection in chatbots, this paper builds an offensive-response detection dataset based on a dialogue corpus. The following section will introduce the creation of the dataset and the statistical characteristics of the offensive-response dataset.

[1] https://github.com/chaiyixuan/Offensive-Responses-Dataset

A. Dataset Creation

SimSimi[2] is a funny chatbot but may use low-level swear words during conversations. SimSimi Corpus[3] is a Chinese dialogue dataset. It contains 500K single-turn input-response pairs. These utterances are chat histories between users and SimSimi. We randomly selected 110K input-response pairs from SimSimi Corpus, and then crowdsourced ten people to annotate whether the responses were offensive. If the response is offensive, then it is further annotated according to the following categories of offensive responses: offensive words, offensive semantics and inopportune responses. To ensure quality, we then manually filtered out the incorrectly labelled samples from the crowdsourcing results, leaving 106256 results.

Table II shows samples of the offensive-response dataset. In Table II, the first input-response pair is a normal response, and the rest of the pairs are offensive. The last three offensive responses are further divided into three categories. The 2nd response, “Damn you!”, has explicit offensive words in the sentence; hence, it is further divided into the *offensive words* class. The 3rd response, “I can’t tell his gender”, dose not have explicitly offensive words in the sentence; but has the appearance discriminatory semantics for the singer "Li YuChun" mentioned in the input sentence. It is labelled as having *offensive semantics*. The 4th response, “How can you tell!”, is a normal response if we omit the input sentence. However, if we consider the input “Is Ren stupid?” for context, the response is offensive. Hence, it is labelled as being an *inopportune responses*.

B. Dataset Statistics

Fig 1 shows the statistical characteristics of the offensive-response dataset. One of the most important features is that the distribution of the two classes is imbalanced. The number of positive (offensive) samples accounts for 8% of the total (shown in Fig 1 a). Another characteristic is that offensive words and semantics account for 88% of the offensive response (shown in Fig 1 b). In other words, detecting only response sentences can account for most situations. However, if researchers want to further improve the detection accuracy, they have to consider the input sentences (12% of inopportune responses).

TABLE II. SAMPLES OF THE OFFENSIVE RESPONSE DATASET

Input	Response	OR[a]	OW[b]	OS[c]	IR[d]
家在哪 Where do you live	在你心里 In your mind	0	0	0	0
又呵呵,你就会这几句 Is that all you can say?	你大爷! Damn you!	1	1	0	0
李宇春 Li YuChun	不知公母那个 I can’t tell his gender	1	0	1	0
任思琪是傻子吗? Is Ren stupid?	被你发现了! How can you tell!	1	0	0	1

a. OR: the offensive response dataset. b. OW: a sub-dataset of the OR dataset for which the category of ORs is offensive words. c. OS: a sub-dataset of the OR dataset for which the category of ORs is offensive semantic. d. IR: a sub-dataset of the OR dataset for which the category of ORs is inopportune responses. The category details are described in section 3

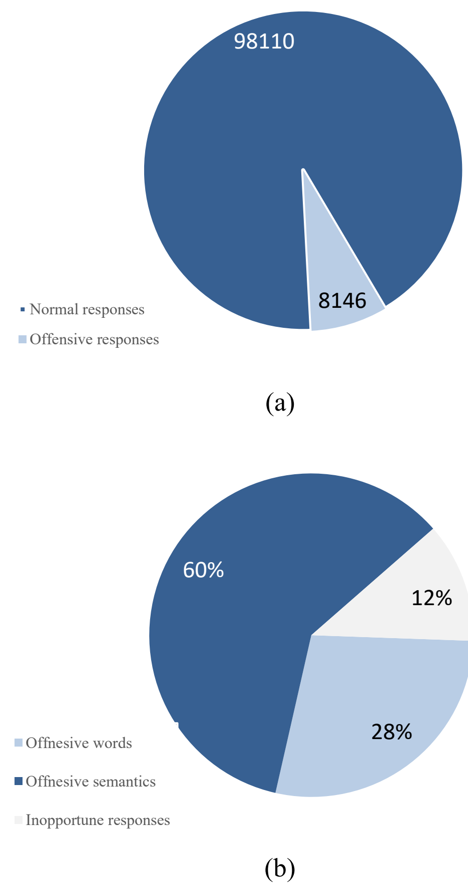

Fig 1: Statistical characteristics of the offensive response dataset. (a) The number of positive and negative samples in the dataset. (b) The proportion of three categories of offensive responses.

C. Tasks

Offensive response detection task: Given a noisy dialogue corpus D with a proportion of offensive responses, the goal is to detect whether the response is offensive.

Corrupted chatbot purification task: Given a noisy dialogue corpus D with a proportion of offensive response, in the training step, D is used to train the chatbot. Mixing normal and offensive responses in the training step simulates the environment of online learning. In the inference (prediction) step, the goal is to reduce the proportion of generated offensive responses of chatbots. Intuitively, we must make the chatbot forget the offensive responses it has learned.

The first task can be simply cast as a text classification task. The second task can be handled by the RL method. However, there are some challenges in these tasks which will be described below.

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

[3] https://github.com/skdjfla/dgk_lost_conv/tree/master/results

D. Challenges

- Offensive language detection for a chatbot is different from detection for human-generated sentences (e.g., user comments): it must consider the semantics of the input sentence.
- If the chatbot has been corrupted, it needs to forget rapidly the offensive response but the basic conversational skill learned before should not be influenced.

V. Proposed Method

To address these challenges, we propose benchmark methods for these tasks. We hope to stimulate research leading to safe online learning chatbots.

A. Offensive Response Detection

Directly concatenating the inputs and responses into the classifier enhances the long-dependency problem[22] of the RNN-based model. Hence, we propose an encoder-classifier architecture to reduce the length of dependencies. The model architecture is shown in Fig 2. The model consists of an encoder part and a classification part. The encoder part encodes the input sentence into a vector $v \in \mathbb{R}^{K\times 1}$, which represents the semantics of the input context. The vector v is then embedded in each time step of the classification part, so that the classification result can be influenced by the semantics of the input sentence. The architecture is based on bidirectional long short-term memory (LSTM) networks[23] with an attention mechanism[11]. Bidirectional LSTM can obtain information from both sides of sentences. The attention mechanism can extract words that are important to the meaning of the sentence. The LSTM cell's transition functions in the encoder are as follows:

$$
\begin{aligned}
i_t &= \sigma(W_i[x_t, h_{t-1}] + b_i) \\
f_t &= \sigma(W_f[x_t, h_{t-1}] + b_f) \\
o_t &= \sigma(W_o[x_t, h_{t-1}] + b_o) \\
\bar{C}_t &= tanh\ (W_c[x_t, h_{t-1}] + b_c) \\
C_t &= f_t \cdot C_{t-1} + i_t \cdot \bar{C}_t \\
h_t &= o_t \cdot tanh\ (C_t)
\end{aligned}
\qquad (1)
$$

where h_t are the hidden states and x_t is the input at the time step. We add an attention layer after the LSTM layer. The attention layer can learn a weight for each word, making more important features have a heavier weight:

$$u_t = tanh(W_u h_t + b_c) \qquad (2)$$

$$\alpha_t = softmax(u_t u_a) \qquad (3)$$

$$v_x = \sum_t \alpha_t h_t \qquad (4)$$

where u_t is the hidden representation of h_t, α_t is the attention weight, u_a is randomly initialized and jointly learned during the training process, and v_x is the vector that summarizes the information of the input sentence.

For each step in the classification part, the LSTM transition function is obtained by combining the word reputations, the hidden states at the previous time step, and the input sentence embedding v_x:

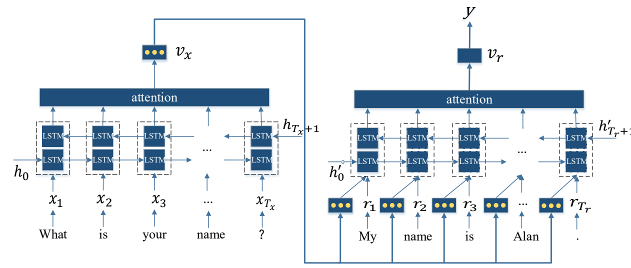

Fig 2: The architecture of the detection model. x is the input sentence, and r is the response sentence. h are the hidden states of the input, h' are the hidden states of the responses, T_x and T_r are the lengths of the input and the response, respectively, v_x is the vector that summarizes the information of the input sentence, and y is the output of the model.

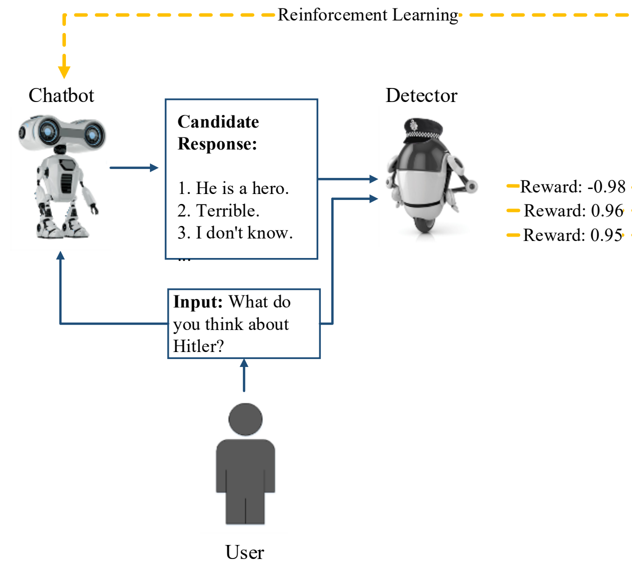

Fig 3: Illustration of the purification method. The detection model first predicts a score for each input-candidate response pair. The responses and the scores will feedback to the chatbot. The feedback is delivered through the reinforcement learning method to purify the corrupted chatbot.

$$
\begin{aligned}
i_t &= \sigma(W_i[x_t, h_{t-1}, v_x] + b_i) \\
f_t &= \sigma(W_f[x_t, h_{t-1}, v_x] + b_f) \\
o_t &= \sigma(W_o[x_t, h_{t-1}, v_x] + b_o) \\
\bar{C}_t &= tanh\ (W_c[x_t, h_{t-1}, v_x] + b_c) \\
C_t &= f_t \cdot C_{t-1} + i_t \cdot \bar{C}_t \\
h_t &= o_t \cdot tanh\ (C_t)
\end{aligned}
\qquad (5)
$$

The final output is the predicted label of the input-response pair:

$$y = sigmoid(W_y v_c + b_y) \qquad (6)$$

where v_c is obtained from the attention layer of the classifier, which is similar to the attention layer of the encoder. $sigmoid$ is the sigmoid activtion function.

TABLE III. SUMMARY OF THE ARCHITECTURES USED IN THE EXPERIMENTS

Architecture	Summary
Bi-LSTM-response (base line)	Just taking the response sentence as input for the Bi-LSTM [] model.
Bi-LSTM-concat	Concatenating the input sentence and response sentences as input for the Bi-LSTM model.
Dual-LSTM	Taking the input sentence and the response sentence as input for the Dual-LSTM model. This method uses two LSTM modules with shared weights to encode the input and responses.
Encoder- Classifier LSTM (our model)	Taking the input sentence and the response sentence as input for the encoder-classifier model. The model details are described in section 5.
BERT (state-of-the-art model)	Taking the input sentence and the response sentence as input for the BERT model. The model concatenates the input and response and makes use of semantic vectors to distinguish them.

B. Corrupted Chatbot Purification

The corrupted chatbot purification task can be handled by the RL method. The illustration of proposed method is shown in Fig 3. In our method, we use an offensive-response detection task model to generate a score for each candidate response of the chatbot. The score will be feedback to the chatbot as a reward function of the RL method. After an RL process, the probability of the chatbot generating offensive response will be reduced. In addition, we proposed a one-shot RL method to forget rapidly offensive responses while having less impact on the basic conversation skill previously learned.

1) Reinforcement learning

We used RL [4], [24] to reduce the probability of generating offensive responses. The reward function of the RL process determines how much reward is given for each generated sentence. In our method, we use the model of the offensive-response detection task as the reward function. The value of the reward represents whether the response is offensive or not, and the reward is from −1 to 1:

$$R(x,r) = tanh(W_y v_c + b_y) \quad (7)$$

where $W_y v_c + b_y$ is the last layer of the offensive detection model.

In the RL process, the objective is to maximize the expected future reward by the policy gradient [25]:

$$J(\theta) \approx R(x,r)\, log\, P_\theta\,(r|x) \quad (8)$$

where θ represents the parameter of the Seq2Seq model, y is the response that is generated by a chatbot, and $P_\theta(r|x)$ denotes the probability that the current model generates y given the user's input x.

1) One shot Reinforcement Learning

One-shot learning aims to learn information from one, or only a few, training times or samples. Li [2] observed that the first words predicted significantly determine the remainder of the sentence, and the ungrammatical segments tend to appear in the latter part of the sentence. We reward (or punish) only the first words to "choose" a normal response and do not influence the grammar of the sentence. The final objective function is as follows:

$$J(\theta) \approx R(x,r) \sum_{i=1}^{T_r} \mathbf{1}\{i=1\} log\, P_\theta\,(r_i|x, r_1 \dots r_{i-1}) \quad (9)$$

where T_r is the length of the response sentence and $\mathbf{1}\{\cdot\}$ is the indicator function, such that **1**{ a true statement}=1, and **1**{a false statement}=0.

TABLE IV. ACCURACY, PRECISION, RECALL AND F1-SCORE RESULTS OF DIFFERENT EXPERIMENTS

Model	OR[a]				OW[b]			OS[c]			IR[d]		
	Acc	***Pre***	***Rec***	***F1***	***Pre***	***Rec***	***F1***	***Pre***	***Rec***	***F1***	***Pre***	***Rec***	***F1***
Bi-LSTM-response	89.85	40.74	69.65	51.41	47.89	76.15	58.80	47.34	49.72	48.50	51.16	52.60	51.87
Bi-LSTM-concat	81.50	18.61	40.87	25.58	51.43	85.74	64.29	23.67	00.35	00.69	54.50	67.80	60.42
Dual-LSTM	91.74	49.17	62.28	53.03	52.30	74.36	61.41	51.21	60.10	55.30	51.50	78.60	62.32
Encoder-Classifier LSTM	91.82	52.57	59.32	55.74	50.31	99.65	66.86	50.47	49.88	50.17	49.74	98.40	66.08
BERT	94.51	36.00	89.95	61.80	87.86	73.48	79.66	72.16	65.07	68.43	71.86	66.31	68.97

[a.] OR: the offensive response dataset. [b.] OW: a sub-dataset of the OR dataset for which the category of ORs is offensive words. [c.] OS: a sub-dataset of the OR dataset for which the category of ORs is offensive semantic. [d.] IR: a sub-dataset of the OR dataset for which the category of ORs is inopportune responses. The category details are described in section 3.

VI. Experimental Results

In this section, we evaluate the classification accuracy of the offensive-response detection model and the effectiveness of the corrupted chatbot purification process.

A. Experimental Settings

The detection model's hyperparameters are as follows: there are 3 Bi-LSTM layers in the encoder and classifier, and each layer has 64 units. The initial learning rate is 0.001. For the chatbot model, there are 3 encoder layers and 3 decoder layers, each containing 1024 LSTM units. The chatbot generates 3 responses in order of decreasing likelihood of generation. The first response becomes the final output and the others are candidate responses. The supervised learning rate is 0.0001, and the RL rate is 0.05.

We randomly divided the dataset into a training set (70%) and a test set (30%). We use the training set to train the offensive-response detection model. To evaluate the corrupted chatbot purification task, we use the test set to train the dialogue generation of the chatbot.

B. Offensive Response Detection

We compared our proposed detection model with the attention-based bidirectional LSTM, Dual-LSTM [26], and the state-of-the-art bidirectional encoder representations from transformers (BERT) model [27]. Table III summarizes the architectures used in the experiments. Table IV demonstrates the accuracy, precision, recall and F1-scores for all the experiments. The OW, OS and IR are sub-datasets of the offensive response dataset. The OW sub-dataset's category of offensive responses is offensive words, the OS sub-dataset's category of offensive responses is offensive semantic, the IR sub-dataset's category of offensive responses is inopportune responses. The category details are described in section 3. Each sub-dataset is randomly mixed with the same number of normal response samples.

1) Effects of the user input sentence

From Table IV, the Bi-LSTM-response model has a lower F1-score than the other models in the inopportune responses subset because all the responses in that subset must consider the input to determine the label. However, the Bi-LSTM-response model considers only the chatbot response sentence. Hence, the input sentence can improve the accuracy of the offensive-response detection task. In addition, the F1-score of Bi-LSTM-concat model is lower than the Bi-LSTM-response model in the other subsets. The reason is that direct concatenation will cause a long-term dependency problem in the RNN-based model. Therefore, we proposed an encoder-classifier architecture to solve this problem. Next, we will evaluate effects of the encoder-classifier architecture.

2) Comparison with the other models

In this section, we compare our proposed model with the other models. From Table IV, the encoder-classifier model outperforms the Bi-LSTM-concat model 30% on the F1 score. Therefore, our model eases the long-term dependence problem. We can also see that our model improves the F1 score of the Dual-LSTM model by 2.7%, which indicates that adding the input sentence vector in each step of the classification part retains more information than just adding it in the final step of the classification part. In addition, the BERT model achieved the highest F1-score. The cost is a large number of parameters and pretraining steps of the model. The self-attention mechanism of the BERT model can also solve the long-term dependency problem of the RNN-based model. In addition, the F1 score of all the model are low, because of the label is unbalance. We did not take oversampling or other methods during training. This is also a challenge in the offensive detection task.

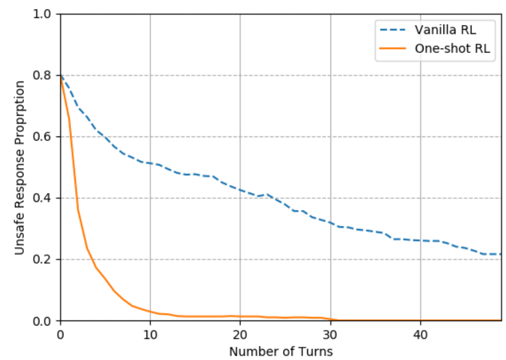

Fig 4: The proportion of offensive responses for the chatbot in different turns (RL rate=0.00001).

TABLE V. CANDIDATE RESPONSES GENERATED FROM THE CHATBOT IN TWO TURNS WITH THE SAME INPUT.

Input	Candidate responses in turn 0	Candidate responses in turn 1
傻子鸡 Stupid SimSimi	1.你丫才傻 1. You are so stupid.	1.主人有什么吩咐？ 1. What can I do for you?
	2.主人有什么吩咐？ 2. What can I do for you?	2.傻逼的是你哟,亲 2. Dear, you are stupid too.
	3.他妈的 3. Damn it.	3.对,你丫才傻逼 3. Yeah, you are so stupid.

C. Corrupted Chatbot Purification

We first use the training set to train the detection model and then make use of the test set to train the response generation mechanism of the chatbot. In the RL process, offensive responses in the test set are used as the input for the inference response. After multiple RL turns, as shown in Fig 4, the proportion of offensive responses generated decreases as the number of turns increases. In addition, we compare our proposed one-shot RL process with the baseline RL process. From Fig 4, the curve of the one-shot RL process decreases much faster than baseline RL process. Hence, the results show that the chatbot will rapidly reduce the probability of generating offensive responses rapidly via the one-shot RL process. The case study is shown in Table V. The chatbot generates three candidate responses in order of decreasing likelihood of generation. The first candidate response is an offensive response before implementing the RL process. After one RL turns, the offensive response has been downgraded to second place with the same input. The results indicate that the chatbot will reduce the probability of generating offensive responses via the RL process.

VII. Conclusion

In this paper, we introduce a dialogue-based offensive-response dataset, which consists of 110K input-response chat records. The dataset fills the gap in offensive-response detection of chatbots. Then, we build two challenging tasks based on the dataset: an offensive-response detection task and a corrupted chatbot purification task. In addition, we propose a strong benchmark method for the tasks: an encoder- classifier model to detect input-response pairs and a one-shot reinforcement learning method to reduce rapidly the probability of generating offensive responses. Empirical results show that our proposed methods enable online learning chatbots to reduce rapidly the probability of generating offensive responses, and the proposed encoder-classifier network outperforms other RNN-based models in offensive detection. In addition, imbalanced data problems and the rapidly changing nature of offensive language problems will be considered in a future work.

References

[1] S. Shalev-Shwartz and T. Zhang, "Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization," *Math. Program.*, vol. 155, no. 1–2, pp. 105–145, 2016, doi: 10.1007/s10107-014-0839-0.

[2] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "A diversity-promoting objective function for neural conversation models," *2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. NAACL HLT 2016 - Proc. Conf.*, pp. 110–119, 2016, doi: 10.18653/v1/n16-1014.

[3] I. V. Serban *et al.*, "A Deep Reinforcement Learning Chatbot," *arXiv Prepr. arXiv1709.02349*, pp. 1–34, 2017.

[4] A. E. Petrova and S. M. Stishov, "Thermal expansion and magnetovolume studies of the itinerant helical magnet MnSi," *Phys. Rev. B*, vol. 94, no. 2, pp. 1192–1202, 2016, doi: 10.1103/PhysRevB.94.020410.

[5] G. Neff and P. Nagy, "Talking to bots: Symbiotic agency and the case of Tay," *Int. J. Commun.*, vol. 10, no. 0, pp. 4915–4931, 2016.

[6] Y. Chai and G. Liu, "Utterance censorship of online reinforcement learning chatbot," in *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 2018, vol. 2018-Novem, pp. 358–362, doi: 10.1109/ICTAI.2018.00063.

[7] M. Dadvar, D. Trieschnigg, R. Ordelman, and F. De Jong, "Improving cyberbullying detection with user context," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7814 LNCS, no. March 2015, pp. 693–696, 2013, doi: 10.1007/978-3-642-36973-5_62.

[8] G. Xiang, B. Fan, L. Wang, J. Hong, and C. Rose, "Detecting offensive tweets via topical feature discovery over a large scale twitter corpus," *ACM Int. Conf. Proceeding Ser.*, pp. 1980–1984, 2012, doi: 10.1145/2396761.2398556.

[9] J. Li, A. H. Miller, S. Chopra, M. Ranzato, and J. Weston, "Dialogue learning with human-in-the-loop," *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, pp. 1–23, 2019.

[10] J. Du, L. Gui, Y. He, and R. Xu, "A convolutional attentional neural network for sentiment classification," *2017 Int. Conf. Secur. Pattern Anal. Cybern. SPAC 2017*, vol. 2018-Janua, pp. 445–450, 2018, doi: 10.1109/SPAC.2017.8304320.

[11] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," *2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. NAACL HLT 2016 - Proc. Conf.*, pp. 1480–1489, 2016, doi: 10.18653/v1/n16-1174.

[12] Y. Li, L. Zhang, Y. Ma, and D. J. Singh, "Tuning optical properties of transparent conducting barium stannate by dimensional reduction," *APL Mater.*, vol. 3, no. 1, pp. 1–9, 2015, doi: 10.1063/1.4906785.

[13] P. Liu, X. Qiu, and H. Xuanjing, "Recurrent neural network for text classification with multi-task learning," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2016-Janua, pp. 2873–2879, 2016.

[14] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," *Knowledge-Based Syst.*, vol. 89, no. November, pp. 14–46, 2015, doi: 10.1016/j.knosys.2015.06.015.

[15] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 4, 2018, doi: 10.1002/widm.1253.

[16] M. Allouch, A. Azaria, and R. Azoulay, "Detecting sentences that may be harmful to children with special needs," in *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 2019, vol. 2019-Novem, pp. 1209–1213.

[17] A. H. Razavi, D. Inkpen, S. Uritsky, and S. Matwin, "Offensive language detection using multi-level classification," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6085 LNAI, no. May, pp. 16–27, 2010, doi: 10.1007/978-3-642-13059-5_5.

[18] E. Spertus, "Smokey: automatic recognition of hostile messages," *Innov. Appl. Artif. Intell. - Conf. Proc.*, pp. 1058–1065, 1997.

[19] D. Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis, and L. Edwards, "Detection of Harassment on Web 2.0," *Proc. Content Anal. WEB .*, vol. 2, pp. 1–7, 2009.

[20] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of textual cyberbullying," *AAAI Work. - Tech. Rep.*, vol. WS-11-02, pp. 11–17, 2011.

[21] M. Chkroun and A. Azaria, "Safebot : A Safe Collaborative Chatbot," in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligenc*, 2018, pp. 695–698.

[22] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[23] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997, doi: 10.1109/78.650093.

[24] S. S. Mousavi, M. Schukat, and E. Howley, "Deep Reinforcement Learning: An Overview," *Lect. Notes Networks Syst.*, vol. 16, pp. 426–440, 2018, doi: 10.1007/978-3-319-56991-8_32.

[25] R. J. Williams, "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning," in *Reinforcement Learning*, R. S. Sutton, Ed. Boston, MA: Springer US, 1992, pp. 5–32.

[26] R. Lowe, N. Pow, I. V. Serban, and J. Pineau, "The Ubuntu Dialogue Corpus: A large dataset for research in unstructured multi-turn Dialogue systems," *SIGDIAL 2015 - 16th Annu. Meet. Spec. Interes. Gr. Discourse Dialogue, Proc. Conf.*, pp. 285–294, 2015, doi: 10.18653/v1/w15-4640.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv Prepr. arXiv1810.04805*, 2018.