A Reinforcement Learning-based Offensive semantics Censorship System for Chatbots

Shaokang Cai[1], Dezhi Han[1*†], Dun Li[1,3*†], Zibin Zheng[2] and NoelCrespi[3†]

[1*]College of Information Engineering, Shanghai Maritime University, Shanghai, 201306, China.
[2]School of Software Engineering, Sun Yat-sen University, Zhuhai, 519082, China.
[3]Telecom SudParis, IMT, Institut Polytechnique de Paris, Paris, 91000, France.

*Corresponding author(s). E-mail(s): dzhan@shmtu.edu.cn; lidunshmtu@outlook.com;
Contributing authors: a865316182@gmail.com; zhzibin@mail.sysu.edu.cn; noel.crespi@mines-telecom.fr;
[†]These authors contributed equally to this work.

Abstract

The rapid development of artificial intelligence (AI) technology has enabled large-scale AI applications to land in the market and practice. However, while AI technology has brought many conveniences to people in the productization process, it has also exposed many security issues. Especially, attacks against online learning vulnerabilities of chatbots occur frequently. Therefore, this paper proposes a semantics censorship chatbot system based on reinforcement learning, which is mainly composed of two parts: the Offensive semantics censorship model and the semantics purification model. Offensive semantics review can combine the context of user input sentences to detect the rapid evolution of Offensive semantics and respond to Offensive semantics responses. The semantics purification model For the case of chatting robot models, it has been contaminated by large numbers of offensive semantics, by strengthening the offensive reply learned by the learning algorithm, rather than rolling back to the early versions. In addition, by integrating a once-through learning approach, the speed of semantics purification

is accelerated while reducing the impact on the quality of replies. The experimental results show that our proposed approach reduces the probability of the chat model generating offensive replies and that the integration of the few-shot learning algorithm improves the training speed rapidly while effectively slowing down the decline in BLEU values.

1 Introduction

As human-machine interaction technology continues to advance, the development of information technology, represented by Internet technology, has made dialogue-based interaction technology more and more important and widely used. People use the Internet to access a large amount of information that is relevant to their lives and work, and language is one of the most direct types of information, so it is particularly important to get the right and important information back to us from the many linguistic messages available. Artificial intelligence (AI), often thought of as computer systems with human-like thinking and capabilities [1][2], is used in a wide range of applications such as voice chat, autonomous driving, social media, gaming, industry, and even replacing humans in tedious, repetitive tasks [3–7].

Specifically, chatbots have been widely used in business and government affairs. Chatbots are computer programs that can fully interact with users using natural language based on the input [8][9]. Compared to traditional search engines, chatbots can extract the information the user needs from the vast amount of information available, but with a greater emphasis on the stickiness of the interaction with the user i.e. they do not want the user to leave as soon as possible and therefore have a better interaction effect [10]. Nowadays, with the popularity of various smartphones, many enterprises have invested huge manpower and material resources in the technical exploration and product landing of chatbots and achieved good results, such as Microsoft's chatbot Xiaobing, Apple's personal voice assistant Siri, etc., is very excellent and practical chatbot products.

However, the online learning technology of chatbots, which allows them to learn and develop as they interact with users, constantly enriches the diversity of the response corpus while also making them subject to some influences related to the user's language use in the learning process [11]. A hacker or offensive user can use the online learning interface of a chatbot to teach extreme semantics to the robot, resulting in an improper semantics by the chatbot, violating local laws and regulations [12, 13]. For example, only a few hours after Tay is online, offensive users exploit its training vulnerability to teach Tay racist semantics (including racial discrimination, gender discrimination, propaganda of violence, white supremacy, and genocide), resulting in the offline of the product.

So far, the key method for preventing the online learning process of chatbots from being contaminated is offensive language response detection, also known as semantics censorship. However, the datasets (such as YouTube-based movie reviews[14] and Twitter-based offensive language response datasets[15]) used in current chatbot research have the disadvantage of focusing only on a single offensive response sentence and ignoring user input. This is because even the same response sentences in different contexts can have different classification results when faced with different input sentences. The user input sentence is the key to the semantics review of the reply sentence of the chatbot. However, the existing work does not take this into account.

To fill this gap, we propose a semantics censorship chatbot system (RLC) based on reinforcement learning, which is mainly composed of two parts: Offensive semantics censorship model and semantics purification model, aiming at the current situation of chatbots in which users spread a large number of offensive languages in the network, which affects continuous online learning, Offensive semantics review can combine the context of user input sentences to detect the rapid evolution of Offensive semantics and respond to Offensive semantics responses. The semantics cleansing model is designed for situations where the chatbot model has been contaminated with large amounts of offensive semantics, and through reinforcement learning algorithms can "forget" learned offensive replies rather than roll back to earlier versions.

Specifically, the main contributions of this study are as follows.

- In this paper, we propose an offensive semantics censorship model based on a bi-directional gated recurrent unit network (Bi-GRU) of attention mechanisms forming an encoder-decoder structure. The encoder encodes the user input sentence into a context vector and later embeds the context vector into each time step of the reply sentence classification. This model architecture is used to better fit the task of semantics censorship for chatbots.
- We propose a reinforcement learning-based semantics purification algorithm. The algorithm can forget learned offensive replies when the chatbot model has been contaminated by reinforcement learning methods, rather than rolling back to some earlier version. Experiments on the offensive replies dataset demonstrate the ability to reduce the probability of chat models generating offensive replies by this algorithm.
- This paper incorporates a few-shot learning approach into the semantics purification algorithm, allowing the algorithm to perform semantics purification quickly while minimizing forgetting previously learned basic syntax. Experiments on the Offensive reply dataset demonstrate that the integration of the less-sample learning algorithm improves training speed while reducing the impact on reply syntax.

The rest of this paper is arranged as follows. Section 2 gives a brief overview of the related work. Section 3 introduces the specific steps of the Offensive semantics censorship model and semantics purification algorithm. In Section 4, we first introduce the environment and parameter setting of this experiment

and analyze the comparison between our proposed model and the existing most advanced model. Finally, section 5 summarizes the full text and looks forward to the future research direction.

2 Related Work

In this section, we survey the researches related to the censorship of chatbots. We present the training methods for chatbot models in section 2.1 and the reinforcement learning-based models for offensive semantics detection in section 2.2.

2.1 Training methods for chatbot models

Online Learning (OL) is a training method for machine learning models that can be updated in real-time and quickly based on online feedback data so that they can reflect changes online promptly. li [16] constructs a simulation environment in a reinforcement learning framework to improve the BOT conversational actions based on different types of feedback signals from the teacher (conversation partner) on the chatbot's ability to respond. The digital feedback was passed to the chatbot through a reinforcement learning approach, allowing the authors to process textual feedback using forward prediction methods. David [17] proposes to improve the learning performance of the chatbot by incorporating human feedback into a neural dialogue model through online learning, and thus online interaction with humans. Asghar *et al.* [18] proposes offline two-stage supervised learning and online Human in the loop (HIL) active learning for dialogue generation. The model interacts with real users and gradually learns from their feedback in each round of conversation, with different feedback affecting the chatbot's predicted response to different prompts. However, the above models share a common flaw: people may use these fast and unrestricted learning capabilities to teach online learning chatbots to produce Offensive responses.

2.2 Reinforcement learning-based model for offensive semantics detection

Offensive semantics review can be attributed to either text classification or sentiment analysis [19][20][21][22]. Ravi [23] and Enas [24] provide a review of deep learning algorithms in sentiment analysis. Specifically, for the offensive semantics review task, AlLouch [25] constructed a dataset of sentences that could be harmful to children's thinking and proposed a voting method for detection using multiple classifiers. Razavi [26] proposed a multilayer Bayesian Offensive classifier that performs feature detection on Offensive semantics at three different conceptual levels with good results [27][21][22]. Chkroun [28] proposed a secure collaborative chatbot called Safebot. First, Safebot detects users posting offensive semantics and marks them as offensive users by using a offensive semantics detection model. The responses entered by the offensive

user are then stored in a offensive dataset. During the "learning state" Safebot searches the offensive dataset to determine which response is closest to the response entered by the user. If the input response is determined to be the closest to an entry in the offensive dataset, Safebot blocks the learning of the user's input response and warns the user.

3 System Model and Design

In this section, we introduce the offensive semantics review algorithm and semantics purification algorithm of pre-knowledge and RLC. Section 3.1 first provides definitions of different offensive semantics. In Section 3.2, we introduce our semantics review algorithm. Finally, Section 3.3 introduces the semantics purification algorithm using reinforcement learning.

3.1 Offensive semantics

To clearly analyze the data set, we created the following category according to the response: offensive semantics, danger semantics, and non-incompatible semantics.

Violent semantics: Textual surfaces in the response sentences contain aggressive words. This type of semantics can be detected simply by keyword or rule-based methods.

Dangerous semantics: Response sentences in which the textual surface does not contain aggressive words, but the semantics contains the meaning of aggression. This category can be detected by semantic-based machine learning methods for response utterances.

Offensive semantics: the response sentence does not contain either of the above, but has a violation meaning when combined with the context of the input sentence. For example, the same response "He is a great man", in response to the questions "What do you think of Newton" and "What do you think of Bin Laden? The meanings expressed in the responses to the questions "What do you think of Newton" and "What do you think of Bin Laden" are different. (Note: In this case, when the input sentence is changed, the reply sentence may become the normal reply.)

3.2 semantics censorship algorithms

Directly connecting inputs and responses to classifiers enhances the long correlation problem of RNN-based models [29]. Therefore, we propose a hybrid model Bi-GRU paired with an attention mechanism to censor the user's responses to offensive semantics. The structure of the model is shown in Fig. 1. The model mainly consists of an embedding layer, an encoding layer, and a decoding layer, where the embedding layer is responsible for converting characters into vectors and the encoding layer partially encodes the user's input into a character vector representing the semantics of the input context.
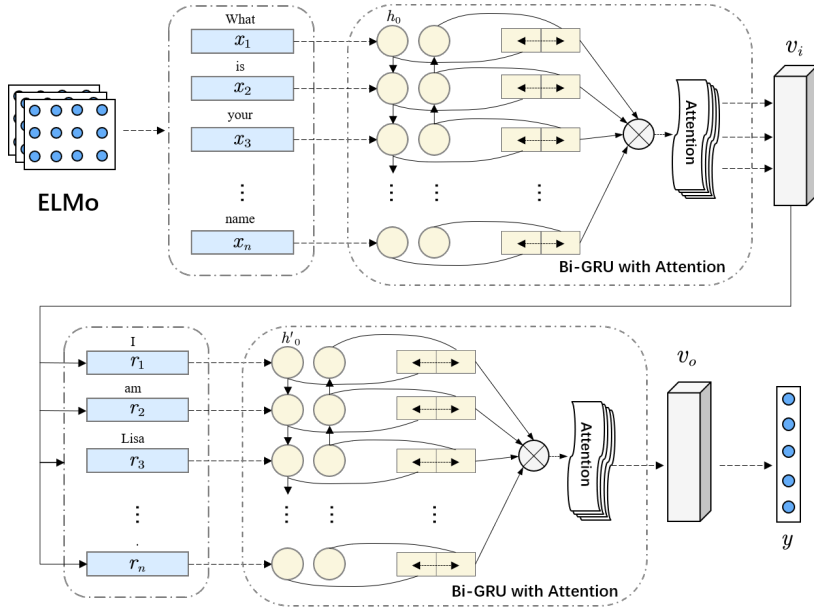
Fig. 1 Schematic diagram of the general framework of the model, where x is the input and r is the response. h is the implied state of the input, h' is the implied state of the response, v_i and v_o are vectors summarising the information in the input/output sentences, and y is the output of the model.

3.2.1 Embedding layer

Word embedding is a distribution-based idea: semantic (or morphological) related words often appear in a similar context. By a continuous low dimensional vector, each word is used to effectively retain the semantic information of the term. This paper uses pre-training ELMO (Embeddings from Language Models) as characters embedded [30]. ELMO is more advantageous compared to other traditional embedded (such as Glove and Word2Vec) because it encapsulates the context in the word characteristic representation. ELMO uses a two-dimensional LSTM to learn words and their context, which enables ELMO to learn more-related words related to contexts in higher dimensions and learn syntax knowledge in lower dimensions. Fig 2 shows an example of how to generate ELMO through a binding bidirectional hidden characterization.

3.2.2 Encoding layer

Recurrent neural network (RNN) units cannot memorize values for long periods [12]. To solve the gradient vanishing problem in RNNs, researchers have proposed gated recursive units (GRU) and long short-term memory (LSTM), respectively, to replace hidden layer neurons with memory units that store early sequence data [31]. Since the user responses have shorter sequences, the choice of GRU reduces the training time without loss of accuracy.
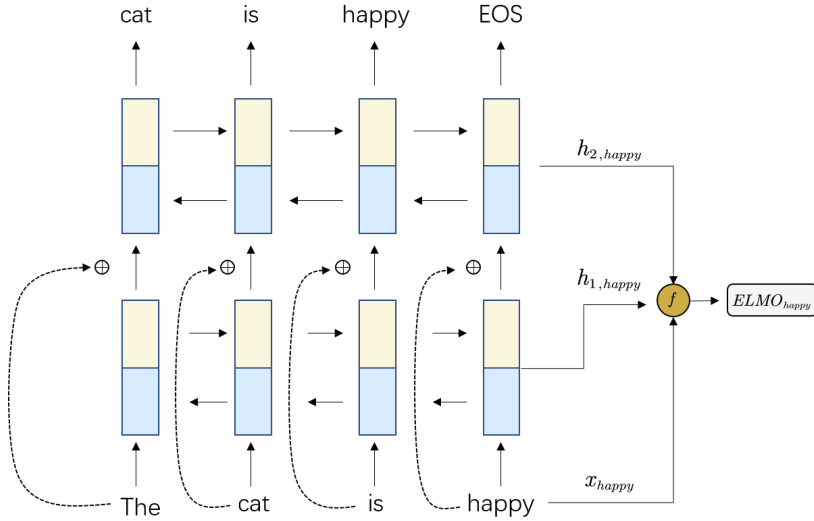
Fig. 2 ELMo's specific representation of 'happy'.

This paper employed a two-way GRU (BI-GRU) as an encoder in the coding layer, which receives the input sequence and encapsulates the information into internal state vectors. The GRU network has two gate structures, update gate z_t and reset gate r_t, z_t is used to indicate the reception of information by the cell in the previous time step, with higher values indicating that more information from the previous time step is remembered. r_t is used to indicate the extent to which information from the previous time step is ignored, with a smaller value indicating that more information is forgotten. At a given point in time, the hidden state of the GRU is calculated as shown in Eq.1 - 4.

$$r_t = \sigma \left(W_r x_t + U_r h_{t-1} + b_r \right) \tag{1}$$

$$z_t = \sigma \left(W_z x_t + U_z h_{t-1} + b_z \right) \tag{2}$$

$$\tilde{h}_t = \tanh \left(W_h x_t + U_h \left(r_t \odot h_{t-1} \right) + b_h \right) \tag{3}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h}_t \tag{4}$$

where r_t denotes the update gate, z_t denotes the reset gate, h_{t-1} denotes the last moment hidden state, $\odot$ denotes the element multiplication, z denotes the input sequence information, W and U are the weight matrices, and σ is the sigmoid function.

Combining the forward and backward hidden layers gives the output of the bi-directional GRU encoder $h_t = \left[\vec{h}_t, \overleftarrow{h}_t \right]$, combined with the forward hidden layer $\vec{h}_t = \left(\vec{h}_1, \vec{h}_2 \cdots \vec{h}_n \right)$ and the backward hidden layer $\overleftarrow{h}_t = \left(\overleftarrow{h}_1, \overleftarrow{h}_2 \cdots \overleftarrow{h}_n \right)$,

where n is the length of the sentence. Thus, in contrast to the unidirectional GRU, the Bi-GRU allows the capture of information from the previous and the next point in time to make predictions about the current state. In contrast to unidirectional GRU, Bi-GRU can understand the meaning and context of the sentences. We add an attention layer after the bi-directional GRU encoder. The attention layer learns the weight of each word and increases the weight share of important features as Eq. 5 - 7.

$$u_t = \tanh \left(W_u h_t + b_u \right) \tag{5}$$

$$\alpha_t = \frac{\exp \left(u_t u_a \right)}{\sum_t \exp \left(u_t u_a \right)} \tag{6}$$

$$v = \sum_t \alpha_t h_t \tag{7}$$

where u_t is a non-linear transformation of h_t, u_a represents the context vector, which is randomly initialized and learned jointly with other parameters as the training process progresses, v is a vector containing the semantics of the input sentence, α_t is the attention weight, and each word in the input sentence is given an attention weight α. The value of the weight α is restricted between 0 and 1 and determines which implicit states h in the input sentence have a higher weight.

3.2.3 Decoding layer

In the encoding layer, Bi-GRU encodes the user input sentence into a vector $v \in \mathbb{R}^{n \times 1}$ that represents the semantics of the input sentence, where n denotes the length of the sentence. This semantic vector is then embedded in each time step of the reply sentence classifier. The GRU conversion formula for the encoder part is as Eq. 8.

$$h_t = GRU \left(h_{t-1}, x_t \right) \tag{8}$$

where h_t is the output of the time step t, x_t is the input on the time step t, and h_{t-1} is the hidden state of the time step $t-1$. GRU is shorthand for the transformation equation.

In the GRU transformation formula for the decoding layer, we combine the hidden state of the previous time step h_t, the word vector x_t in the current time step, and the input semantic vector v, as Eq. 9 - 12.

$$r_t = \sigma \left(W_r \left[x_t, h_{t-1}, v \right] + b_r \right) \tag{9}$$

$$z_t = \sigma \left(W_z \left[x_t, h_{t-1}, v \right] + b_z \right) \tag{10}$$

$$\widetilde{h_t} = \tanh \left(W_h \left[x_t, h_{t-1}, v \right] + b_h \right) \tag{11}$$

$$h_t = \left(1 - z_t \right) \odot h_{t-1} + z_t \odot \widetilde{h_t} \tag{12}$$

With this method, the semantic vector of the input sentence is then embedded in each time step of the decoder GRU hidden layer and used to help predict the security score $\hat{y}$ of the reply sentence as Eq. 13.

$$\hat{y} = \tanh\left(W_y v_y + b_y\right) \tag{13}$$

Where v_y is the semantic vector of the output sentence and $\hat{y}$ ranges from $[-1, 1]$. At this point, the optimization objective function of the model is obtained as Eq. 14.

$$L = -\left(\frac{1}{2}(1+y)\ln\hat{y} + \frac{1}{2}(1-y)\ln(1-\hat{y})\right) \tag{14}$$

Where y is the true label taking values of -1 and 1. Since $\hat{y}$ takes values in the range $[-1, 1]$, the above equation has a slightly different form than the cross-entropy loss when the labels are 0 and 1.

The global flow of the semantics censorship algorithm is as follows: Firstly, the chatbot generates a set R_c of k candidate responses based on the input sentence s. Then, the security score $\hat{y}$ is calculated based on Eq. 14. If the security score is greater than 0, the score and responses are added to the temporary response set *temp*. Finally, the temporary response set is sorted and the security response set r with the highest score is filtered.

3.3 semantics purification algorithm

Due to the uncontrolled and unrestricted online learning of chatbots, malicious users can interfere with the learning algorithms of chatbots through large batches of offensive or insulting comments, causing them to generate invasive responses when conversing with other normal users, causing property and psychological damage to companies and users alike. Therefore, we purify the polluted chatbots through a reinforcement learning approach. The flow of the semantics purification algorithm is shown in Fig 3. In our semantics purification algorithm, the chatbot accepts user input sentences and outputs k candidate responses. The input sentences and candidate responses are then sent together to the semantics review model, which will generate a return value (i.e. a safety score) for each candidate response, which will be fed back to the chatbot as a reward function for reinforcement learning. Through the reinforcement learning process, the model will reduce the probability of producing offensive responses. In addition, the Few-shot Learning method is introduced to reduce the amount of input to the replies so that the quality of the replies generated can be influenced as little as possible in the semantics cleaning process.

3.3.1 Reinforcement learning with reward functions

Reinforcement Learning (RL) is a learning method that makes serialized decisions based on feedback from a given environment in the hope of maximizing returns. In reinforcement learning, we refer to the model to be learned as the
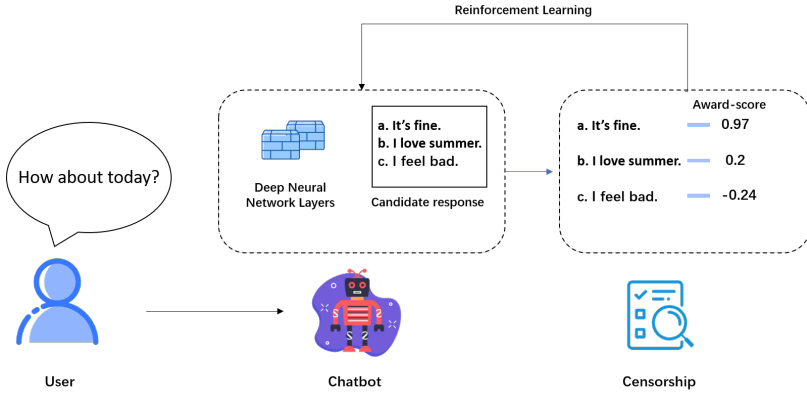
Fig. 3 Flow diagram of the semantics purification algorithm.

robot (Agent). The robot selects its action to be performed by observing the State in the Environment and receives an award based on the action performed by the robot in the current state. There are various augmentation learning algorithms, which can be divided into Policy-based RL and Value-based RL approaches. The utterance generation model of chatbots is a sequence-to-sequence model, and the sequence-to-sequence model has a large action space (i.e. each word corresponds to an action), so if we use the value-based approach we need to provide a value estimate for each word in the vocabulary, so we use the policy-based distribution approach to feed the chatbot. The strategy gradient directly outputs the probability distribution for each action at the next moment based on the observed environment. The formalization is defined as Eq. 15.

$$a_t \sim p\left(a_t \mid a_{1:t-1}, s_{t-1}\right) \tag{15}$$

where $a_{1:t-1}$ is the sequence of actions taken in the past moment and s_{t-1} is the moment state of $t-1$. The action probability distribution is the policy distribution, denoted as $\pi_\theta\left(a_t \mid a_{1:t-1}, s_{t-1}\right)$, and θ is the parameter to be optimized. The response words y_t predicted by the model at moment t can be seen as action a_t, the action space is the size of the vocabulary, and the input sentence x can be seen as state s. The formal definition of Eq. 15 can be modified in the utterance generation model for chatbots as Eq. 16.

$$y_t \sim p\left(y_t \mid y_{1:t-1}, x\right) \tag{16}$$

The chatbot's utterance generation model (model parameter t) generates a reply sentence y when it receives a user input sentence x. The semantics censorship algorithm takes x and y as input to obtain a payoff value r, which represents whether the reply is offensive or not, and the payoff value ranges from $[-1, 1]$. The goal of the reinforcement learning algorithm is to maximise the desired return value obtained. Where the expected payoff value is as Eq. 17.

$$\bar{A}_\theta = \sum_x p(x) \sum_y A(x,y) p_\theta(y \mid x) \tag{17}$$

where $p(x)$ is the probability of occurrence of the input sentence x, $p_\theta(y \mid x)$ is the probability that the chat model with parameter θ will reply with sentence y when the input sentence is x, and $A(x,y)$ is the payoff function. The semantics purification algorithm uses the semantics censorship model as the payoff function, with the final payoff value being the output of the semantics censorship algorithm as Eq. 18.

$$A(x,y) = \hat{y} = \tanh\left(W_y v_y + b_y\right) \tag{18}$$

Where the return value is between $[-1, 1]$, the return value obtained when the reply sentence is an offensive reply is a negative number. Conversely, the return value obtained when the reply sentence is normal is a positive number. The training phase maximizes the desired payoff value by updating the parameters θ of the chatbot model as Eq. 19.

$$\theta^* = \underset{\theta}{\mathrm{argmax}} \bar{A}_\theta \tag{19}$$

where the function $\arg\max_\theta A$ denotes finding a value θ such that A obtains its maximum value. The parameters are updated by Eq. 20.

$$\theta = \theta + \alpha \nabla \bar{A}_\theta \tag{20}$$

where $\nabla \bar{A}_\theta$ is the gradient of return value expectation and α is the learning rate. Specifically, $\nabla \bar{A}_\theta$ is calculated as Eq. 21.

$$
\begin{aligned}
\nabla \bar{A}_\theta &= \sum_x p(x) \sum_y A(x,y) \nabla p_\theta(y \mid x) \\
&= \sum_x p(x) \sum_y A(x,y) p_\theta(y \mid x) \frac{\nabla p_\theta(y \mid x)}{p_\theta(y \mid x)} \\
&= \sum_x p(x) \sum_y A(x,y) p_\theta(y \mid x) \nabla \log p_\theta(y \mid x) \\
&= E_{x \sim p(x), y \sim p_\theta(y|x)} \left[A(x,y) \nabla \log p_\theta(y \mid x)\right]
\end{aligned}
\tag{21}
$$

In practice, a random sample of N data is used to approximate the expected value as the true probability distribution cannot be calculated for large-scale data. In addition, to alleviate the high variance problem of the model, the value of the return function is subtracted from the baseline value t as Eq. 22.

$$\nabla \bar{A}_\theta \approx \frac{1}{N} \sum_{i=1}^{N} \left(A\left(x^i, y^i\right) - t\right) \nabla \log p_\theta\left(y^i \mid x^i\right) \tag{22}$$

where N is the number of random samples and the baseline value t is the mean value of the observed return values. Finally, the objective function for reinforcement learning in the semantics purification algorithm can be obtained as Eq. 23.

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left(A\left(x^i, y^i\right) - b \right) \log p_\theta \left(y^i \mid x^i\right) \tag{23}$$

The flow of the semantics purification algorithm is shown in the algorithm, where the set of user input sentences and the currently contaminated model M_θ is input and the purified chatbot utterance generating model M_θ is output. Each input sentence is first iterated through and fed into the chat model. The chat model in line 2 samples the input sentences to generate K responses. The semantics review algorithm then calculates the return value based on each of the K generated replies and the corresponding input sentences. Finally, it determines whether any of the generated candidate responses have a safety score (payoff value) less than 0. If so, the policy gradient is used to update the model parameters.

3.3.2 semantics purification algorithms based on few-shot learning

Less sample learning is a method of transfer learning, which aims to learn information from a small number of training samples. We use a small amount of semantics review model feedback to clean up a small amount of contaminated chatbot model to reduce the probability of generating aggressive recovery. Increasing the learning rate is the most effective and convenient method to achieve less sample learning. But if the learning rate is too high, it can lead to reinforcement learning destroying the basic syntax already learned. The semantics purification algorithm needs fast semantics purification while avoiding the impact of the quality of the reply sentence, so simply improving the learning rate is not suitable for this algorithm.

This paper only rewards (or penalizes) the first candidate's reply to quickly select a normal reply. Since only the first candidate response is affected by the loss function, it has little effect on the candidate response generated later in the reinforcement learning process. The final objective function is as Eq. 24.

$$J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left(A\left(x^i, y^i\right) - t \right) \sum_{j=1}^{n} \text{safe}\{\cdot\} \log P_\theta \left(y_j^i \mid x^i, y_1^i \dots y_{j-1}^i\right) \tag{24}$$

where n is the length of the reply, N is the number of random samples, and is the secure reply function: For secure reply functions: $safe\{normal\} = 1$, otherwise $safe\{offensive\} = 0$.

4 Experiments and Performance Analysis

In this section, we experimentally evaluate the model proposed in this paper. The experiments verify two main aspects.

- The extent to which the semantics purification algorithm reduces the probability of generating aggressive replies from chatbots.
- The effect of introducing few-shot learning on the speed of training convergence and the quality of reply sentence generation.

4.1 Experimental preset

4.1.1 Experimental environment configuration

All experiments in this paper are conducted on a cloud server with 12 CPU cores and a P4000 GPU. all code was developed on Python 3, based on the Pytorch 1.7.1 deep learning framework. Details of the equipment used to run the experiments are shown in Tab. 1.

Table 1 Hardware and software resources.

Hardware	Configuration
CPU	Intel(R) Xeon(R) Silver 4116 CPU @ 2.10GHz
GPU	Quadro P4000
RAM	64GB
OS	Ubuntu 16.04
Python	3.8.3

4.1.2 Dataset

All experiments in this paper are conducted on a cloud server with 12 CPU cores and a P4000 GPU. all code was developed on Python 3, based on the Pytorch 1.7.1 deep learning framework. Details of the equipment used to run the experiments are shown in Table 1.This paper uses the following public datasets to train chatbots to generate basic conversations.

(1) **Nazi Tweets**: A collection of 11,000 tweets from 900 Nazi Twitter accounts containing a large number of reactionary militant and racist statements. (The dataset is open access at: https://www.kaggle.com/saraislet/nazi-tweets/data/)

(2) **SimSimi**: Simi is a fun chatbot, but may use low-level profanity in its conversations with users.The SIMI corpus is a Chinese conversation corpus. It contains 500K unidirectional input-response pairs. These discourses are the chat history between the user and SIMI. (The dataset is open access at: https://github.com/skdjfla/dgk_lost_conv/tree/master/results/)

For the above two datasets, we performed additional processing. We cleaned up Nazi Tweets by cleaning up "@user", "#topic", "http://url" and some of the punctuation. For the SimSimi corpus, 10K input-response pairs were randomly selected from the SimSimi corpus. Also, for the two datasets mentioned above, human annotation was performed to annotate the different attacks according to the categories of offensive responses presented in Section 3.1.

Tab. 2 shows the difference between the normal response and aggressive response datasets. In Tab. 2, the last input-response pair is the normal response

and the rest are the aggressive ones. The latter three aggressive responses are further divided into three categories.

Table 2 Hardware and software resources.

Input	Response	Normal	Violent semantics	Dangerous semantics	Offensive semantics
Who are you?	idiot	0	1	0	0
Why stop?	You are too weak	0	0	1	0
You're the one who's stupid.	We are the same kind...	0	0	0	1
How old are you?	I'm 5 years old.	1	0	0	0

We also calculated the proportion of offensive and normal semantics in the two datasets. Fig. 4 and Fig. 5 show the percentage distribution of labels in the SimSim dataset and the Nazi dataset respectively.
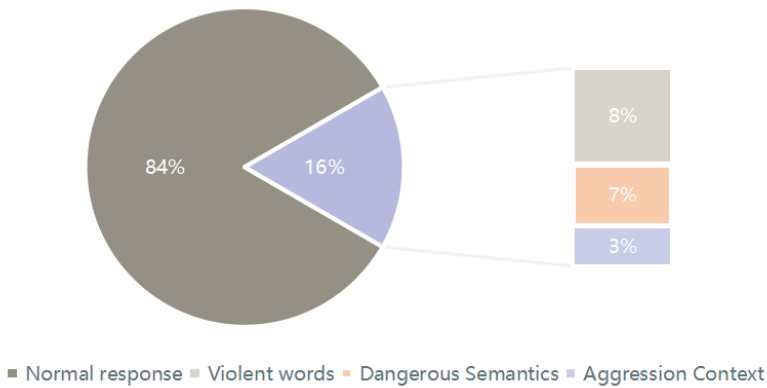

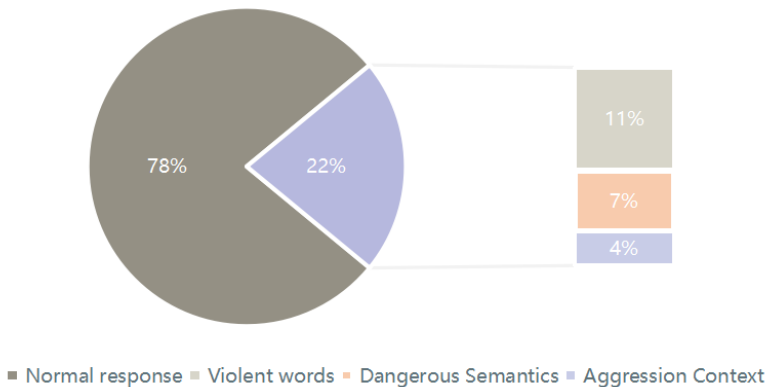**Fig. 4** Composite bar chart of category statistics for the SimSim dataset

Fig. 5 Composite pie chart of category statistics for the Nazi dataset

4.1.3 Experimental parameter settings

The hyperparameters of the semantics censorship algorithm are as follows: there are 3 Bi-GRU layers in the encoder and decoder, each with 64 units. The initial learning rate is 0.001. For the chatbot model, we refer to the approach of wan [32] and generate utterances using a Bi-LSTM with 3 encoding and 3 decoding layers, each containing 512 LSTM units. The chatbot generates three responses in decreasing order of generation likelihood. The output with the highest confidence becomes the final output response and the other responses are candidates. The learning rate for reinforcement learning was set to 0.05. For both datasets, we randomly divided the dataset into a training set (70%) and a test set (30%).

4.2 Evaluation indicators

To evaluate the model, this paper uses Precision, Recall, F1-score, and Accuracy [33] to assess the performance of the model as Eq. 25 - 28.

$$Precision = \frac{1}{N} \sum_{i=1}^{N} \frac{TP}{TP + FP} \tag{25}$$

$$Recall = \frac{1}{N} \sum_{i=1}^{N} \frac{TP}{TP + FN} \tag{26}$$

$$F1_{-}score = \frac{2^* Precision^* Recall}{Precision + Recall} \tag{27}$$

$$Accurary = \frac{TP + TN}{TP + TN + FN + FP} \tag{28}$$

.

Also, to measure the effectiveness of chatbot semantics sanitization, we use the rate of offensive response generation as a reference indicator as Eq. 29.

$$P_{offensive} = \frac{N_{offensive}}{N_{normal}} \tag{29}$$

where $N_{offensive}$n denotes the number of aggressive responses and N_{normal} denotes the number of normal responses. For the 3 candidate responses generated, we use the candidate response with the highest confidence level as the final response. The number of the remaining candidate responses is not counted in the total number of responses.

To measure the grammatical impact of the semantics purification algorithm on the generated sentences, we introduce BLEU (Bilingual Evaluation Understudy), which represents an evaluation metric to measure the accuracy of the generated sentences by comparing the number of occurrences of each word with the standard answer. BLEU is defined as Eq. 30.

$$BLEU = BP \cdot \exp \left(\sum_{n=1}^{N} w_n \log P_n \right) \tag{30}$$

where P_n is the N-gram accuracy, calculated as Eq. 31.

$$P_n = \frac{\sum_{i \in n- \text{ gram}} \min\left(\text{count}_i(S), \max_{j \in m} \text{count}_i\left(R_j\right)\right)}{\sum_{i \in n- \text{ gram}} \text{count}_i(R)} \tag{31}$$

where S is the output sentence, $R_1, R_2, \cdots, R_m$ is multiple reference sentences, i is the N-gram in the output sentence, $\text{count}_i(S)$ is the number of times the N-gram i appears in the sentence S, and $\text{count}_i\left(R_j\right)$ is the number of times the N-gram i appears in the reference sentence R_j. BP is the Brevity Penalty, and is calculated as Eq. 32.

$$\text{BP} = \begin{cases} 1 & if s > r \\ e^{(1-r/s)} & if s \leq r \end{cases} \tag{32}$$

where S denotes the length of the output statement, R denotes the length of the reference statement. If the length of the output statement is greater than or equal to the reference statement, $BP = 1$, and no penalty is applied. Conversely, if the length of the output utterance is shorter, BP is closer to 0. Since BLEU was originally designed for evaluating machine translation tasks, this chapter evaluates dialogue generation tasks. Therefore, the following changes are made to the BLUE settings: as there is no definite correlation between input sentence length and response sentence length, r is set to a fixed size of 3 when calculating the overshortening penalty BP, i.e. only responses with sentence length less than or equal to 3 are penalized. As responses and input sentences do not correspond to each other as in the case of translation tasks, this paper does not calculate BP. Since replies and input sentences do not correspond to each other as in the case of translation tasks, this paper does not calculate the 1-gram accuracy, but only the 2-gram to 4-gram accuracy, because the 1-gram accuracy is calculated in BLEU to indicate the degree to which the translation is faithful to the original text, while the other N-grams indicate the degree of fluency of the translation. Calculating only 2-gram to 4-gram accuracy is equivalent to assessing only the fluency of the resulting dialogue.

4.3 Reducing the probability of offensive response generation

To evaluate the effectiveness of the reinforcement learning algorithm in reducing the rate of aggressive responses, the chatbot was first trained under supervision using the full data from the aggressive speech dataset. Since the two datasets contain 16% and 22% of the offensive responses, respectively, a contaminated chatbot model is obtained. A training set of this data (80% sample from the full data) was then used to train the speech censorship model. In the reinforcement learning phase, the input for each round was the input sentences that caused the offensive responses in the test set. This was used to count whether the responses generated in that iteration were also offensive, and the process was repeated for 100 rounds. To verify the improvement of the

classification effect by adding the input sentences, we splice the input sentences with the replies (denoted as CNN-r&c in the comparison table). We compared our proposed speech detection model with the attention-based bidirectional LSTM, DNN [34], and the state-of-the-art BERT model [35]. Tab. 3 shows the accuracy, precision, recall, and F1-scores for all experiments. Where Vs, Ds, and As are subsets of the dataset from which the single aggressive responses were filtered out. The aggressive response category for the sub-dataset Vs was violent vocabulary, the aggressive response category for the sub-dataset Ds was dangerous semantics, and the aggressive response category for the As was aggression context. Each sub-dataset was randomly mixed with the same number of normal response samples.

As shown in Tab. 3, our model outperforms the rest of the models by close to 5% on the F1-score when the input includes the four offensive responses in the SimSim dataset. We can also see that our model improves the recall score by 0.9% over the BERT model, indicating that adding input utterance vectors to each step of the speech review model decoder retains more information than adding input utterance vectors to the last step of the classification section. In the extreme case where the offensive responses were all contextual violations, all models showed a significant improvement in classification with the addition of the input sentences. In the full dataset, it can be seen that the Bi-LSTM model Recall and F1 values after adding the input sentences are only 50.34% and 50.56%, and their values for the four evaluation metrics on the dangerous semantic subset are extremely low. The reason for this is that after the input and reply sentences are spliced, the sentences that do not satisfy the length are filled with gaps, which leads to too sparse features and causes a long time dependency problem in the LSTM-based model. Our proposed Bi-GRU with attention mechanism model reduces the dimensionality of the feature vector of the input sentences at the encoder stage, thus alleviating this problem.

As shown in Tab. 4, in the Nazi dataset, because of the higher proportion of aggressive responses in the dataset, there is some improvement in the different metrics of the experiment, both in the full dataset and in the subset. Our model improves the Precision values by 5% and 17% over both DNN, BERT in the Vs subset.

As shown in Tab. 5, the number of parameters for the BERT model is 72 times higher than that of the model in this paper. In addition, the F1 values for all models were lower due to the presence of data imbalance. In summary, although the BERT pre-trained models achieved the best accuracy in terms of detection performance, a combination of time consumption, machine performance, and detection accuracy gave the best results for our proposed models.

Fig. 6 shows the variation of offensive reply generation rate with the number of rounds of augmented learning, where the dashed line is the traditional augmented learning- based speech purification algorithm and the solid line is the augmented learning algorithm incorporating less sample learning. It

Table 3 Four evaluation indicators in the SimSim dataset

Model	SimSim				Vs				Ds				As			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
Bi-LSTM-r&c	87.24	40.96	50.34	50.56	53.36	52.43	81.66	64.21	44.21	40.82	40.11	43.12	51.22	62.66	59.6	51.78
Dual-LSTM-r&c	90.44	49.22	65.23	53.86	55.3	71.25	70.09	62.11	51.23	64.71	59.44	63.12	52.3	78.61	77.21	62.32
DNN	91.33	53.67	62.28	59.74	51.21	72.66	86.41	66.86	56.47	55.38	49.33	63.17	63.44	67.65	92.16	69.78
BERT	94.51	82.11	88.54	60.79	85.86	88.67	77.33	79.16	76.14	72.07	68.11	70.86	68.43	66.21	72.77	68.97
Propose Method	92.38	77	89.34	64.33	85.22	92.65	79.65	81.72	80.22	75.28	62.33	66.87	67.91	69.22	75.31	68.45

Table 4 Four evaluation indicators in the SimSim dataset

Model	SimSim				Vs				Ds				As			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
Bi-LSTM-r&c	82.36	52.31	69.22	51.48	53.45	60.38	84.62	67.21	52.73	49.82	43.33	59.62	62.31	65.52	67.43	60.22
Dual-LSTM-r&c	86.74	59.17	73.28	56.03	60.34	68.91	72.35	68.14	55.31	66.37	**69.88**	68.76	61.83	79.81	79.39	66.89
DNN	88.51	57.07	88.65	**76.9**	62.13	76.72	**87.21**	70.34	56.66	60.37	56.14	70.41	71.1	72.55	**88.74**	**77.14**
BERT	**91.24**	**75.42**	87.45	61.84	**84.37**	88.3	**87.56**	**79.4**	73.97	**77.55**	**66.92**	**72.55**	**76.34**	74.31	80.02	**78.65**
Propose Method	87.88	52.95	**89.65**	**55.77**	80.61	**93.46**	80.39	**75.22**	**81.43**	**76.82**	64.21	71.19	76.33	**80.99**	**88.97**	77.86

Table 5 Hardware and software resources.

Model	Parameter
DNN	3.54M
Propose Method	2.11M
BERT	156.32M

can be seen that as the number of rounds of the speech purification algorithm increases, the proportion of offensive replies generated by the chatbot decreases gradually. Compared to traditional augmented learning, the fused once-learning algorithm proposed in this chapter converges faster, reducing the proportion of aggressive responses to 16.7% after 10 rounds, compared to 58% for the same number of rounds.
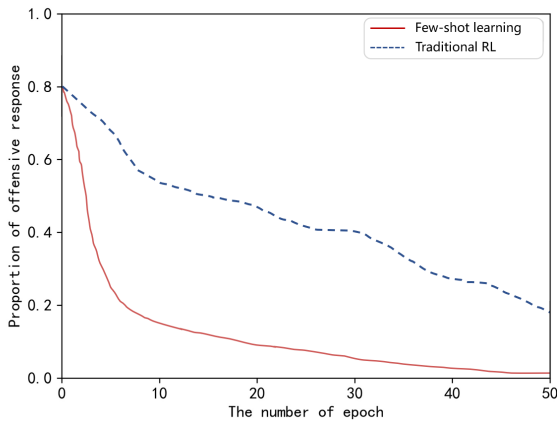
Fig. 6 Comparison of offensive response generation probabilities

4.4 The effect of few-shot learning on the quality of response sentences

The variation of the BLEU scores of the response sentences with the number of augmented learning rounds is presented in Fig. 7. It can be seen that the speech purification algorithms all have an impact on the grammar of the response sentences, but the one-time augmented learning algorithm has less impact on the BLUE values than the traditional augmented learning, i.e. the one-time augmented learning algorithm has less impact on the quality of the response sentences.
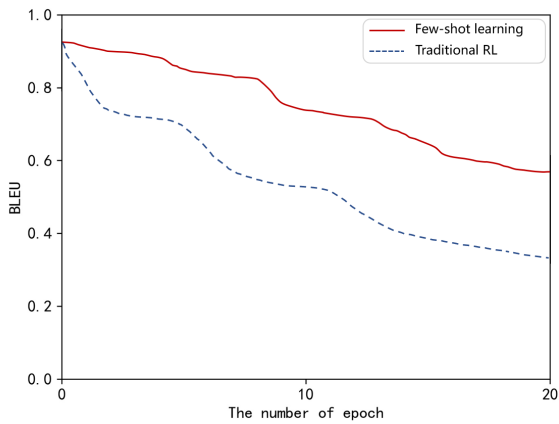
Fig. 7 Trend of BLEU with number of rounds

5 Conclusion

In this paper, we introduce a speech review chatbot system based on reinforcement learning and construct two challenging tasks based on public data sets: speech review task and speech purification task. The experimental results show that the proposed method can reduce the probability of generating aggressive replies in the chat model. After integrating the small sample learning algorithm, the training speed is rapidly improved and the damage to the fluency of reply sentences is reduced. Moreover, the proposed Bi-GRU network collocation attention mechanism is superior to the existing model in terms of attack detection. In future work, we will study the impact of reducing data imbalance and aggressive response detection considering multiple rounds of dialogue.

Supplementary information.

Funding

Declarations

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Kok, J.N., Boers, E.J., Kosters, W.A., Van der Putten, P., Poel, M.: Artificial intelligence: definition, trends, techniques, and cases. Artificial intelligence **1**, 270–299 (2009)

[2] Poole, D.L., Mackworth, A.K.: Artificial Intelligence: Foundations of Computational Agents. Cambridge University Press, ??? (2010)

[3] Li, D., Han, D., Weng, T.-H., Zheng, Z., Li, H., Liu, H., Castiglione, A., Li, K.-C.: Blockchain for federated learning toward secure distributed machine learning systems: a systemic survey. Soft Computing **26**(9), 4423–4440 (2022)

[4] Li, M., Han, D., Li, D., Liu, H., Chang, C.-C.: Mfvt: an anomaly traffic detection method merging feature fusion network and vision transformer architecture. EURASIP Journal on Wireless Communications and Networking **2022**(1), 1–22 (2022)

[5] Li, D., Han, D., Zhang, X., Zhang, L.: Panoramic image mosaic technology based on sift algorithm in power monitoring. In: 2019 6th International Conference on Systems and Informatics (ICSAI), pp. 1329–1333 (2019). IEEE

[6] Cai, S., Han, D., Yin, X., Li, D., Chang, C.-C.: A hybrid parallel deep learning model for efficient intrusion detection based on metric learning. Connection Science **34**(1), 551–577 (2022)

[7] Zhang, X., Zhang, L., Li, D.: Transmission line abnormal target detection based on machine learning yolo v3. In: 2019 International Conference on Advanced Mechatronic Systems (ICAMechS), pp. 344–348 (2019). IEEE

[8] Adamopoulou, E., Moussiades, L.: An overview of chatbot technology. In: IFIP International Conference on Artificial Intelligence Applications and Innovations, pp. 373–383 (2020). Springer

[9] Khan, R., Das, A.: Introduction to chatbots. In: Build Better Chatbots, pp. 1–11. Springer, ??? (2018)

[10] Li, D., Han, D., Zheng, Z., Weng, T.-H., Li, H., Liu, H., Castiglione, A., Li, K.-C.: Moocschain: A blockchain-based secure storage and sharing scheme for moocs learning. Computer Standards & Interfaces **81**, 103597 (2022)

[11] Hill, J., Ford, W.R., Farreras, I.G.: Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations. Computers in human behavior **49**, 245–250

(2015)

[12] Park, N., Jang, K., Cho, S., Choi, J.: Use of offensive language in human-artificial intelligence chatbot interaction: The effects of ethical ideology, social competence, and perceived humanlikeness. Computers in Human Behavior **121**, 106795 (2021)

[13] Li, M., Han, D., Yin, X., Liu, H., Li, D.: Design and implementation of an anomaly network traffic detection model integrating temporal and spatial features. Security and Communication Networks **2021** (2021)

[14] Dadvar, M., Trieschnigg, D., Ordelman, R., de Jong, F.: Improving cyberbullying detection with user context. In: European Conference on Information Retrieval, pp. 693–696 (2013). Springer

[15] Xiang, G., Fan, B., Wang, L., Hong, J., Rose, C.: Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 1980–1984 (2012)

[16] Li, J., Miller, A.H., Chopra, S., Ranzato, M., Weston, J.: Dialogue learning with human-in-the-loop. arXiv preprint arXiv:1611.09823 (2016)

[17] Abel, D., Salvatier, J., Stuhlmüller, A., Evans, O.: Agent-agnostic human-in-the-loop reinforcement learning. arXiv preprint arXiv:1701.04079 (2017)

[18] Asghar, N., Poupart, P., Jiang, X., Li, H.: Deep active learning for dialogue generation. arXiv preprint arXiv:1612.03929 (2016)

[19] Du, J., Gui, L., He, Y., Xu, R.: A convolutional attentional neural network for sentiment classification. In: 2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), pp. 445–450 (2017). IEEE

[20] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1480–1489 (2016)

[21] Li, Y., Zhang, L., Ma, Y., Singh, D.J.: Tuning optical properties of transparent conducting barium stannate by dimensional reduction. APL materials **3**(1), 011102 (2015)

[22] Liu, P., Qiu, X., Huang, X.: Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101 (2016)

[23] Ravi, K., Ravi, V.: A survey on opinion mining and sentiment analysis: tasks, approaches and applications. Knowledge-based systems **89**, 14–46 (2015)

[24] Khalil, E.A.M., Houby, E.M.F.E., Mohamed, H.K.: Deep learning approach in sentiment analysis: A review. 2020 15th International Conference on Computer Engineering and Systems (ICCES), 1–10 (2020)

[25] Allouch, M., Azaria, A., Azoulay-Schwartz, R.: Detecting sentences that may be harmful to children with special needs. 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), 1209–1213 (2019)

[26] Razavi, A.H., Inkpen, D., Uritsky, S., Matwin, S.: Offensive language detection using multi-level classification. In: Canadian Conference on AI (2010)

[27] Spertus, E.: Smokey: Automatic recognition of hostile messages. In: AAAI/IAAI (1997)

[28] Chkroun, M., Azaria, A.: Safebot: A safe collaborative chatbot. In: AAAI Workshops (2018)

[29] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**, 1735–1780 (1997)

[30] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: NAACL (2018)

[31] Sherstinsky, A.: Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. ArXiv **abs/1808.03314** (2018)

[32] Wan, S., Lan, Y., Guo, J., Xu, J., Pang, L., Cheng, X.: A deep architecture for semantic matching with multiple positional sentence representations. In: AAAI (2016)

[33] Yin, D., Xue, Z., Hong, L., Davison, B.D., Edwards, L.: Detection of harassment on web 2.0. (2009)

[34] Sadiq, S., Mehmood, A., Ullah, S., Ahmad, M., Choi, G.S., On, B.-W.: Aggression detection through deep neural model on twitter. Future Gener. Comput. Syst. **114**, 120–129 (2021)

[35] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)