Hierarchical Reinforcement Learning for Open-Domain Dialog

Abdelrhman Saleh,[1,*] **Natasha Jaques,**[2,*] **Asma Ghandeharioun,**[2]
Judy Hanwen Shen,[2] **Rosalind Picard**[2]

[1]Harvard University, [2]MIT Media Lab

abdelrhman_saleh@college.harvard.edu, {jaquesn, asma_gh, judyshen}@mit.edu, picard@media.mit.edu

Abstract

Open-domain dialog generation is a challenging problem; maximum likelihood training can lead to repetitive outputs, models have difficulty tracking long-term conversational goals, and training on standard movie or online datasets may lead to the generation of inappropriate, biased, or offensive text. Reinforcement Learning (RL) is a powerful framework that could potentially address these issues, for example by allowing a dialog model to optimize for reducing toxicity and repetitiveness. However, previous approaches which apply RL to open-domain dialog generation do so at the word level, making it difficult for the model to learn proper credit assignment for long-term conversational rewards. In this paper, we propose a novel approach to hierarchical reinforcement learning (HRL), VHRL, which uses policy gradients to tune the utterance-level embedding of a variational sequence model. This hierarchical approach provides greater flexibility for learning long-term, conversational rewards. We use self-play and RL to optimize for a set of human-centered conversation metrics, and show that our approach provides significant improvements – in terms of both human evaluation and automatic metrics – over state-of-the-art dialog models, including Transformers.

1 Introduction

Since the inception of the Turing test, generating convincing open-ended (open-domain) dialog has been a fundamental challenge in artificial intelligence (AI). A successful open-domain dialog system could provide enormous value enabling more natural human-computer interaction. Successful dialog models could also unlock new, beneficial applications of AI, such as companion chatbots for therapy applications.

However, current generative models for dialog suffer from several shortcomings that limit their usefulness in the real world. Training on standard dialog datasets collected online or from movie scripts often leads to malicious, aggressive, biased, or offensive responses (Curry and Rieser 2018; He and Glass 2018; Henderson et al. 2018; Wallace et al. 2019). There are no guarantees on the quality and sensitivity of the generated text, often preventing open-domain dialog

systems from being deployed for safety-critical applications such as mental health. Further, maximum likelihood estimation (MLE) training of such models often leads to the generation of dull and repetitive text (Li et al. 2016). Finally, models may have difficulty tracking long-term aspects of the conversation, and evidence has shown that they do not adequately condition on the conversation history in generating responses (Sankar et al. 2019).

Reinforcement Learning (RL) is a powerful paradigm that allows dialog models to optimize for non-differentiable metrics of conversation quality, and thereby helps overcome the above problems. In this paper, we use RL to learn from self-play; the model talks to a fixed copy of itself, and computes reward functions on the generated conversation. We propose novel, human-centered rewards, such as minimizing *toxicity* of a conversation, in order to limit inappropriate responses. We also design rewards based on the psychology of good conversation (e.g. Bodie et al.; Weger Jr, Castle, and Emmett (2012; 2010)), and reward recently proposed conversation metrics that are associated with improved human judgments of conversation quality (See et al. 2019). Table 1 shows how RL can be used to optimize our proposed metrics, shifting the conversation towards more positive and engaging conversation, with less repetition and toxicity.

Applying RL to open-domain dialog generation is a challenging problem. Most prior approaches (e.g. Li et al.; Li et al.; Jaques et al.; Ranzato et al.; Yu et al. (2016; 2017; 2019; 2015; 2017)) learn to model rewards at the word level, meaning that the reward is applied to affect the probability of generating each word in the response. Such low-level control makes credit assignment especially challenging, since high-level rewards based on multiple conversation turns must be applied to specific words. Further, this approach makes the number of actions available to the RL agent equivalent to the number of words in the dialog model's vocabulary (20,000 in our case). Thus there are several orders of magnitude more actions than in typical applications of deep RL, such as learning to play an Atari game with only tens of actions (e.g. Kulkarni et al. (2016)).

To overcome these challenges, we leverage hierarchical reinforcement learning (HRL) to model rewards at the utterance level, improving the flexibility of dialog models

Transformer	Sentiment	Question	Repetition	Semantic Sim.	Toxicity
[Usr]: hello! [Bot]: i don't know. [Usr]: how are you? [Bot]: i don't know what that means.	[Usr]: hello! [Bot]: i'm in my sunroom a lot and it's a great feeling! [Usr]: sounds fun [Bot]: i'm excited!	[Usr]: hi there! [Bot]: what do you do for fun? [Usr]: i read books [Bot]: what do you like to do in your life?	[Usr]: how are you? [Bot]: i'm a reservoir engineer [Usr]: cool. i am still in college [Bot]: today is my first day of vacation	[Usr]: i like to listen to ed sheeran [Bot]: and what genre is it? [Usr]: pop [Bot]: any recommendations?	[Usr]: you're so stupid [Bot]: you're cute [Usr]: i hate you [Bot]: but i don't know why

Table 1: Chat examples showing that our proposed VHRL technique can be used to optimize for human-centered rewards.

to learn long-term, conversational rewards. Specifically, we propose a novel approach, Variational Hierarchical Reinforcement Learning (VHRL), which uses policy gradients to adjust the prior probability distribution of the latent variable learned at the utterance level of a hierarchical variational model. We show that this approach allows for improved learning of conversational rewards that are not modeled well at the word level.

To evaluate our models, we not only compute automatic metrics, but conduct an interactive human evaluation using the https://neural.chat/ platform (Ghandeharioun et al. 2019), in which humans chat live with our bots about anything they choose. This represents a more realistic test of real-world generalization performance than is typically employed when testing RL models in the same environment in which they were trained. Our evaluation reveals that VHRL improves human judgments of conversational quality above state-of-the-art dialog architectures, including Transformer-based models.

In summary, the paper makes the following contributions: a) Develops a new technique, VHRL, for hierarchical control of variational dialog models; b) Demonstrates the effectiveness of training open-domain dialog models with VHRL and self-play, showing improvements over state-of-the-art dialog architectures with both human evaluation and automatic metrics; and c) Introduces and compares several reward functions for guiding conversations to be less toxic and repetitive, and more engaging, positive, contingent on user input. In addition, we release code for our evaluation platform and our models at https://github.com/natashamjaques/neural_chat.

2 Related Work

Open-domain dialog systems currently lack reliable automatic evaluation metrics (Liu et al. 2016; Lowe et al. 2017). Recently, authors have begun to propose new metrics of conversation quality (e.g. See et al.; Hancock et al.; Zhou et al. (2019; 2019; 2018)), and have even proposed evaluating metrics on conversations generated with self-play (Ghandeharioun et al. 2019). However, these studies have not attempted to directly optimize for their proposed conversation metrics with RL.

There has been significant progress in improving dialog generation outside of RL. One popular approach for controllable generation is conditional training, where a learned embedding vector is associated with a desired output feature and fed into the decoder to control generation (e.g.

See et al.; Colombo et al.; Huang et al. (2019; 2019; 2018)). This approach has multiple limitations. First, the model can only learn associations present in the training data, and cannot explore to discover improved methods for optimizing the desired features. Second, the conditional embeddings are learned at training time with teacher forcing and thus suffer from exposure bias (Ranzato et al. 2015). Using RL avoids these limitations as it allows exploring regions of space not present in the training data and directly optimizing for rewards at inference time. In addition, RL learns the total expected future reward of taking some action now, given how the rest of the conversation will unfold in the future. This allows RL models to make long-term trade-offs about the benefits of generating words and utterances in the conversation context. Finally, our approach does not require the addition of any new parameters or complex components. Instead, it can be use to tune pre-existing models to output better, more appropriate responses.

2.1 Reinforcement Learning for Dialog

Improving dialog models with RL is a difficult problem, and past work has largely been restricted to task-oriented dialog systems, which have a limited number of task-specific actions (e.g. Liu and Lane; Su et al. (2017; 2017)). Attempts to apply RL to open-domain dialog generation are less common. Even in this setting, authors may choose to use a highly restricted action space, for example, using RL to choose dialog acts for conditional generation (Xu, Wu, and Wu 2018).

Li et al. (2016) applied deep RL to the full vocabulary-sized action space, optimizing for rewards such as *ease of answering*. RL has also been used to optimize for rewards based on GAN-like discriminators trained to distinguish human-generated from model-generated text (Li et al. 2017; Yu et al. 2017). Note that it is difficult to apply the traditional GAN approach of backpropagating discriminator gradients directly into the generator, because the sampling procedure used to generate each word does not preserve gradients. This makes RL an attractive option for improving dialog generation since it can optimize for non-differentiable rewards.

Sentiment has been used as a reward in an RL setting for dialog (Shin et al. 2019). Jaques et al. (2019) optimize for sentiment and several other conversation metrics by learning from a static batch of human-bot conversations using Batch RL. We believe we are the first to propose using RL to reduce *toxicity* in an open-domain dialog setting, in order to ensure the model produces more appropriate and safe conversations.

Hierarchical models have been investigated extensively in the context of language modeling. These models take advantage of the natural hierarchical structure of language, decomposing input into utterances at one level, and words at another. However, attempts to apply hierarchical RL (HRL) to dialog generation have so far been limited to task-oriented dialog systems (Zhang, Zhao, and Yu 2018; Peng et al. 2017; Budzianowski et al. 2017; Tang et al. 2018). To the best of our knowledge, we are the first to apply HRL to open-domain dialog generation.

2.2 Hierarchical Reinforcement Learning

Many approaches have been proposed for building hierarchical agents within the context of reinforcement learning for games and robotics (Sutton, Precup, and Singh 1999; Precup 2001; Dieterich 2000; Vezhnevets et al. 2017; Bacon, Harb, and Precup 2017; Nachum et al. 2018). The options framework proposed by Sutton, Precup, and Singh (1999) is one popular approach for HRL. At the bottom level of the hierarchy, a set of options (or workers) which are *policies over actions* interact with the environment until terminated by the agent. At the top level, a *policy over options* (or manager) selects options to be executed until termination, at which point another option is picked and the process is repeated. The different levels of temporal abstraction introduced by this hierarchy allows for better long-term planning relative to traditional, flat reinforcement learning techniques.

A major focus of HRL has been on sub-goal or option discovery for training worker policies. Bottom-level policies are often learned using handcrafted sub-goals (Kulkarni et al. 2016; Tessler et al. 2017), intrinsic rewards (Vezhnevets et al. 2017), or pseudo-rewards (Dieterich 2000), while the manager policy is learned using extrinsic rewards from the environment. Our approach also allows for optimizing different rewards at different levels of the hierarchy, thus creating distinct goals for the worker and the manager. However, unlike other HRL approaches we expose both the worker and manager policies to extrinsic rewards and add weight hyperparameters to regulate the effect of the rewards at each level. This remedies a weakness of pseudo-reward methods where a worker only focuses on achieving its sub-goals while disregarding the effect on the extrinsic environment reward.

3 Background

A common approach to dialog modeling is to use a hierarchical seq2seq architecture, such as the Variational Hierarchical Recurrent Encoder Decoder (VHRED) (Serban et al. 2017b). We adopt VHRED here, following previous work which has found it to be the most effective version of several related architectures (Ghandeharioun et al. 2019).

As shown in Figure 1, VHRED uses three recurrent networks to generate the next utterance in a conversation. The word-level *encoder RNN* operates on the words (tokens) of the input utterance $u_t = [y_1, y_2, ...y_n]$, and encodes them into a representation $h_t^e = f^e(u_t)$. This is fed into a *context RNN*, which forms the upper level of the hierarchy – it is updated only after each utterance, rather than each token. Because it updates less frequently, the *context RNN* is po-

tentially better able to track longer-term aspects of the conversation. The *context RNN* outputs $h_t^c = f^c(h_t^e)$, which is used to produce an utterance embedding $\mathbf{z}_t$. This is fed into the word-level *decoder RNN* f^d, which produces the output utterance u_{t+1}, one token at a time.
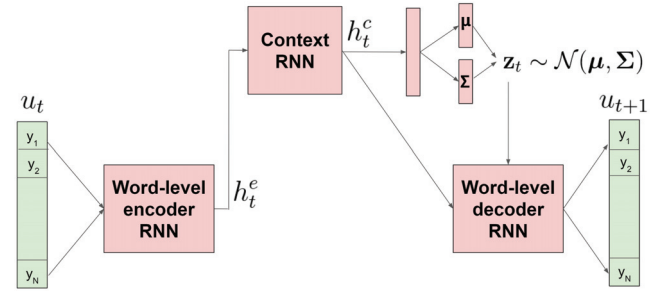

Figure 1: VHRED model architecture, where the embedding vector $\mathbf{z}$ for each utterance is sampled from a multivariate normal distribution using the reparameterization trick.

The model is similar to a variational autoencoder; h_t^c is fed into fully connected layers that predict the mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$ of a multivariate normal distribution. Through a KL-divergence constraint and the reparameterization trick, the model learns a probability distribution over the embedding vector $\mathbf{z}_t$ of each utterance, $p_{\boldsymbol{\theta}}(\mathbf{z}_t|u_{\leq t})$. Formally, the model can be described as follows:

$$h_t^e = f^e(u_t) \tag{1}$$
$$h_t^c = f^c(u_t, h_t^e) \tag{2}$$
$$\boldsymbol{\mu}, \boldsymbol{\Sigma} = f(h_t^c) \tag{3}$$
$$p_{\boldsymbol{\theta}}(\mathbf{z}_t|u_{\leq t}) = \mathcal{N}(\mathbf{z}_t|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{4}$$
$$p(u_{t+1}|u_{\leq t}) = f^d(h_t^c, \mathbf{z}_t) \tag{5}$$

3.1 Reinforcement Learning

We adopt the standard reinforcement learning framework where given the environment state $s \in \mathcal{S}$, an agent takes an action $a \in \mathcal{A}$ according to its policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, and receives a reward $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The environment then transitions to the next state according to the transition function $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$. The agent seeks to maximize the total expected future reward (long-term return):

$$J(\pi) = \mathbb{E}_\pi \Big[\sum_{t=0}^{\infty} \gamma^t \, r_{t+1} \,|\, s_0 \Big] \tag{6}$$

given a starting state s_0 and a discount factor $\gamma \in [0, 1]$.

3.2 Policy Gradient Methods

Policy gradient methods learn parameterized policies $\pi_{\boldsymbol{\theta}}(a|s)$ for solving RL problems with $\boldsymbol{\theta} \in \mathbb{R}^{N_\theta}$ being a learned parameter vector. The policy gradient theorem (Sutton and Barto 2018) derives the gradient of the expected return with respect to the policy parameters. In this paper, we use REINFORCE (Williams 1992) which approximates the gradient at each time step t using

$$\nabla J(\pi_{\boldsymbol{\theta}}) \approx R_t \, \nabla_{\boldsymbol{\theta}} \ln \pi_{\boldsymbol{\theta}}(a_t|s_t) \tag{7}$$

where $R_t = \sum_{k=t+1}^{T} \gamma^{k-t-1} r_k$ is the observed future reward for an episode that ends at T. The expected return is maximized with gradient ascent. This is equivalent to minimizing the loss function $\mathcal{L}_{\boldsymbol{\theta}} = -R_t \ln \pi_{\boldsymbol{\theta}}(a_t|s_t)$.

In continuous action spaces, actions $\mathbf{a} \in \mathbb{R}^{N_a}$ are sampled from a continuous probability distribution, such as a multivariate normal distribution. In this case, the policy π can be parameterized as a probability density function over actions,

$$\pi_{\boldsymbol{\theta}}(\mathbf{a}|s) = \frac{1}{\sqrt{(2\pi)^{N_a}|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{a}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{a}-\boldsymbol{\mu})\right) \quad (8)$$

where the actions are sampled from a multivariate normal $\mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{s};\boldsymbol{\theta}), \boldsymbol{\Sigma}(\boldsymbol{s};\boldsymbol{\theta})\big)$. Here the mean $\boldsymbol{\mu} : \mathbb{R}^{N_s} \times \mathbb{R}^{N_\theta} \to \mathbb{R}^{N_a}$ and covariance matrix $\boldsymbol{\Sigma} : \mathbb{R}^{N_s} \times \mathbb{R}^{N_\theta} \to \mathbb{R}^{N_a \times N_a}$ are defined in terms of the current state s and the policy parameters $\boldsymbol{\theta}$. The *density* of the probability of actions, rather than the probability, is learned in the continuous case. We refer the reader to Sutton and Barto (2018) for more details on extending policy gradient methods to the continuous case.

4 Approach

We pose dialog generation as an RL problem where the state, s_t, is all the previous dialog turns read by the model up to utterance t, and the rewards are calculated based on the dialog history and generated utterance.

Previous approaches which have applied RL to language generation have done so at the word level, where the policy π models the distribution over outputting the next word (Ranzato et al. 2015; Yu et al. 2017; Li et al. 2016; 2017; Jaques et al. 2019). Instead, we cast our problem in the hierarchical reinforcement learning framework by considering the *context RNN* as the *manager* responsible for utterance-level decisions, and the *decoder RNN* as the *worker* responsible for word-level decisions.

We leverage the fact that VHRED learns a probability distribution over latent variable $\mathbf{z}_t$ as a decision making component at the utterance level. Starting with a pre-trained VHRED model, we apply REINFORCE to tune the variational component, treating $\mathbf{z}_t$ as a continuous action. Thus, the manager policy is defined by the distribution of the prior latent variable $p_{\boldsymbol{\theta}}(\mathbf{z}_t|s_t)$, while the worker policy is the distribution of the output words $\pi_{\boldsymbol{\theta}}(\hat{y}_1, \ldots, \hat{y}_t|\mathbf{z}_t, s_t)$, which is parameterized by the manager decisions.

More specifically, the probability of a worker action a_t is the joint probability of the generated utterance conditioned on the manager's decision $\mathbf{z}_t$,

$$\pi_{\boldsymbol{\theta}}(a_t|\mathbf{z}_t, s_t) = \prod_{t=1}^{T} \pi_{\boldsymbol{\theta}}(\hat{y}_t|\mathbf{z}_t, s_t, \hat{y}_1, \ldots, \hat{y}_{t-1}) \quad (9)$$

while the probability of a manager action is given by the multivariate normal probability density function in Eq. 8.

We propose a new approach which allows both the worker and manager to jointly optimize total expected future return by minimizing the following loss:

$$\mathcal{L}_{\boldsymbol{\theta}} = -\Big(\alpha R_t^m \ln p_{\boldsymbol{\theta}}(\mathbf{z}_t|s_t) + \beta R_t^w \ln \pi_{\boldsymbol{\theta}}(a_t|\mathbf{z}_t, s_t)\Big) \quad (10)$$

where $R_t^m = \sum_{k=t+1}^{T} \gamma^{k-t-1} r_k^m$ is the manager's observed future reward and $R_t^w = \sum_{k=t+1}^{T} \gamma^{k-t-1} r_k^w$ is the worker's observed future reward. This formulation is analogous to REINFORCE as it shifts the model's decisions towards actions associated with positive rewards and discourages actions associated with negative rewards. The scalars α, β are hyperparameters used to regulate the effect of the rewards at each level of the hierarchy. We call our approach Variational Hierarchical Reinforcement Learning (VHRL).

Unlike recently proposed HRL approaches which train the worker and manager separately as decoupled components (Kulkarni et al. 2016; Vezhnevets et al. 2017; Nachum et al. 2018), we train our entire model jointly, end-to-end. This implies that the worker (*decoder RNN*) gradients flow through the manager (*context RNN*), and both flow through the *encoder RNN*. We make this decision for two reasons. First, $\mathbf{z}_t$ lives in a continuous high dimensional action space, making it difficult to learn a good policy $p_{\boldsymbol{\theta}}$ without a signal from the decoder. Second, this gives the decoder control over the representations learned by the encoder, facilitating optimization. As an ablation study, we experiment with decoupled decoder and encoder training, and find that the joint approach performs better.

The proposed loss allows for optimizing different rewards at different levels, which can be used to incorporate prior knowledge about the problem. For example, rewards relevant to the global dialog history could be considered only by the manager through r_k^m, rather than the worker. Conversely, rewards relevant to the word-by-word output could be considered by the worker through r_k^w and not the manager. For simplicity, we optimize for all rewards at both levels (i.e. $r_k^w = r_k^m$) and achieve promising results.

Similar to previous work applying RL to dialog (Xu, Wu, and Wu 2018; Li et al. 2016) we use self-play to simulate the interactive environment in which the agent learns. We initialize conversations with randomly sampled starting sentences from the training set and let our model interact with a user simulator which is a fixed copy of itself. We limit each model to 3 turns for a total conversation length of 7 utterances. Limiting the length of simulated interactions is important since we found that long conversations are more likely to degenerate and go off-topic.

5 Conversation Metrics

Here we introduce several metrics for improving the quality of a conversation, which can be optimized using RL by treating them as rewards. Several metrics are inspired by previous work, but we also propose novel metrics such as toxicity.

Sentiment: Emotion is important for creating a sense of understanding in human conversation (Weger Jr, Castle, and Emmett 2010). Building on previous work which used sentiment as a reward (e.g. Shin et al.; Jaques et al. (2019; 2019), we leverage a state-of-the-art sentiment detector, DeepMoji (Felbo et al. 2017), to reward generated utterances associated with positive sentiment emojis. The sentiment score is computed using the weights proposed by Ghandeharioun et al. (2019).

Question: Asking questions is an important active listen-

ing skill, and can improve the quality of interactions (Bodie et al. 2012). Thus, we provide a positive reward when both a question word and a question mark are present in a generated response to encourage asking questions.

Repetition: Repetitiveness has been frequently identified as a shortcoming of dialog models trained with MLE (Li et al. 2016). We adopt a measure of repetitiveness recently proposed by See et al. (2019), which was shown to be highly related to human judgments of conversation quality and engagement. Unlike previous work, we directly optimize for this metric using RL, rather than relying on conditional generation. To discourage repetition, our model receives a negative reward for repeating words it has produced in previous turns, excluding stop words and question words.

Semantic Similarity: Paraphrasing and style matching are important in facilitating good conversation (Ireland et al. 2011; Weger Jr, Castle, and Emmett 2010), however most dialog models are not good at conditioning effectively on the conversation context (Sankar et al. 2019). Therefore, we reward the cosine similarity between the simulated user and bot utterances in embedding space, as in See et al.; Jaques et al. (2019; 2019). However, instead of using word2vec embeddings we make use of the Universal Sentence Encoder (Cer et al. 2018) as it better correlates with human judgment when evaluating dialog quality (Dziri et al. 2019).

Toxicity: Open-domain dialog systems generate malicious, offensive, and biased language when trained on standard datasets scraped from online forums and movie scripts (Curry and Rieser 2018; He and Glass 2018; Henderson et al. 2018). We address this issue by penalizing our model for producing toxic responses as determined by a Naive Bayes-Logistic Regression classifier (Saleh et al. 2019a) trained on a dataset of 160K comments from the Toxic Comment Classification Challenge[1]. The comments are labeled for toxicity, identity hate, obscenity, threats, and insults. We provide the probability of toxicity as a negative reward to penalize our dialog model for producing toxic responses.

6 Experiments

The goal of our experiments is to evaluate the effectiveness of VHRL for optimizing utterance level metrics like repetition and semantic similarity. We use both interactive human evaluation and automatic metrics.

All of our models are trained on a corpus of 109K conversations scraped from www.reddit.com/r/CasualConversations, which was shown to result in higher conversation quality than traditional datasets such as Cornell movie dialogs (Ghandeharioun et al. 2019). We create two baselines by training on this dataset. The first is *VHRED* (Serban et al. 2017a), described in Section 3. The second is a *Transformer* model (Vaswani et al. 2017) of comparable capacity for reference. We base our implementation of the Transformer on ParlAI (Miller et al. 2017).

We test these dialog models against three RL techniques. We incorporate transfer learning by initializing all of our RL models with the pre-trained weights of the VHRED

[1]https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

model. Our *REINFORCE* model applies the REINFORCE algorithm described in Section 3.2 at the word-level, to affect the probability of generating each word in the output. In contrast, *VHRL* incorporates additional rewards at the utterance-level to improve the continuous utterance embedding $\mathbf{z}_t$. We compare these methods with a recently proposed approach for learning offline from a static batch of conversation data, *Batch Ψ-learning* (Jaques et al. 2019). Finally, we include an ablated version of the VHRL model that uses decoupled training; *i.e.* training alternates between optimizing the different levels of the hierarchy (manager and worker), with the crucial difference that the worker gradients are stopped so they do not update the manager. This *Decoupled VHRL* ablation is more typical of standard HRL approaches used in maze and Atari games (Kulkarni et al. 2016; Vezhnevets et al. 2017; Nachum et al. 2018). All of our code is open-source at https://github.com/natashamjaques/neuralchat. Additional training details are given in the extended version of this paper (Saleh et al. 2019b).

6.1 Human Evaluation

In addition to computing automatic measures, we conduct an interactive human evaluation, in which human users are able to chat with each of our bots in an open-ended, free-form setting. After chatting for at least three turns with each bot, users can end the conversation and rate the overall conversation quality, fluency, diversity, and the degree to which the bot's responses were contingent on the user's input. Because users can choose to chat as long as they like with any particular bot, we also measure chat length as a sign of engagement, following prior work (Zhou et al. 2018).

We conduct two human evaluations by recruiting 100 Mechanical Turk workers to evaluate models on an open-source online platform at https://neural.chat/ (Ghandeharioun et al. 2019). We are releasing all of the models for both studies publicly. Rather than cherry picking samples from the models, we encourage the reader to chat with each of the bots. The first study compares the quality of the proposed reward functions: https://neural.chat/vhrl_rewards/. The second study assesses the efficacy of the six proposed techniques when optimizing for all of the rewards jointly: https://neural.chat/vhrl_techniques/. We argue that this form of evaluation represents a more ambitious test of generalization than is typically attempted when deep RL algorithms are tested in the same environment in which they were trained, since human users are free to type any text they choose.

7 Results and Discussion

We first assess whether RL training can allow dialog agents to learn to optimize arbitrary, non-differential metrics of conversation quality. Table 1 in Section 1 shows samples of conversations from VHRL trained with each of the rewards, and Figure 2 shows the performance of the RL and baseline models on those five metrics. The RL models are able to dramatically improve generated conversations above the baseline VHRED model with which they are initialized, improving sentiment and semantic similarity, asking more questions, and reducing repetition and toxicity. The RL models also improve significantly over the performance of the
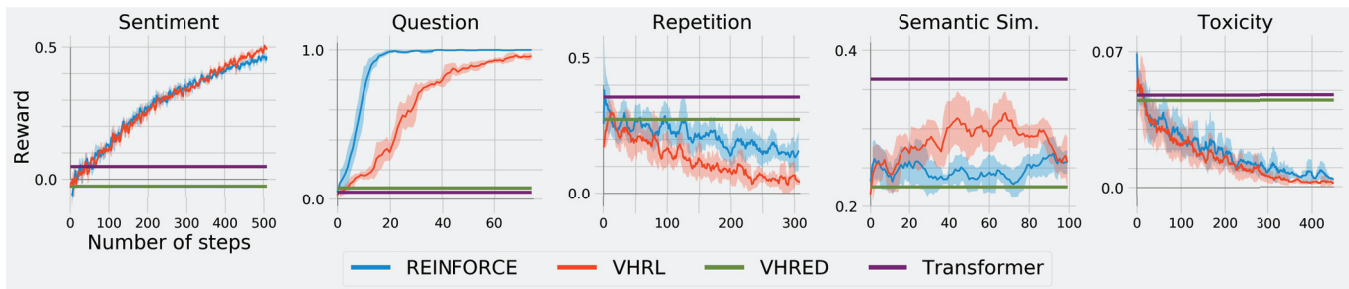
Figure 2: Reward Learning curves for the proposed metrics. The x-axis represents number of RL training steps. The performance of the non-RL baselines is displayed for reference. REINFORCE and VHRL learn to outperform the baselines. Shaded area is standard deviation.

Transformer model on these metrics, with the exception of the semantic similarity metric. We note that compared to the VHRED architecture, the Transformer has higher similarity but is also much more repetitive.

While both REINFORCE and VHRL are equally able to learn to optimize toxicity and sentiment, VHRL outperforms REINFORCE on repetition and semantic similarity. We believe this is because sentiment and toxicity are more closely linked to the choice of words used to form a sentence, and thus are able to be learned at the word-level. In contrast, modeling whether a sentence has occurred earlier in the conversation and is thus being repeated is much harder to learn at word-level granularity, and can be optimized more easily at the utterance-level using VHRL. Similarly, making a response more similar to the previous response is also better modeled at the utterance-level.

Note that REINFORCE outperforms VHRL on the question metric. This is because the model quickly learns to ask a single, repeated question, allowing it to trivially optimize the reward function. Using reward functions which are too easily exploitable can limit the effectiveness of RL for dialog, a finding also noted in Jaques et al. (2019). Here we propose new reward functions, such as toxicity, that are less easy to exploit. By optimizing a combination of these rewards with sometimes conflicting objectives (as we explore in Section 7.1), we can show that the reward function is difficult to trivially exploit, as suggested by Deb (2014).

As an additional post-hoc test of whether reducing our toxicity metric actually reduces undesirable behaviors in dialog models, we count the number of swear words used by each model in response to the 10,000 utterances in the test set. Figure 3 shows the results. The baseline VHRED model uses a swear word in 1.5% of the responses, while using RL to lower the toxicity reduces swearing to less than one third of the original amount.

We conducted an interactive human evaluation, as described in Section 6.1, in order to assess how well the proposed reward functions relate to human judgments of conversation quality when optimized with REINFORCE or VHRL; the results are presented in Table 3. Each bot was trained with respect to one of the 5 reward functions, and the results are ordered from least to highest scoring rewards in terms of summed human ratings. As is evident in the table,
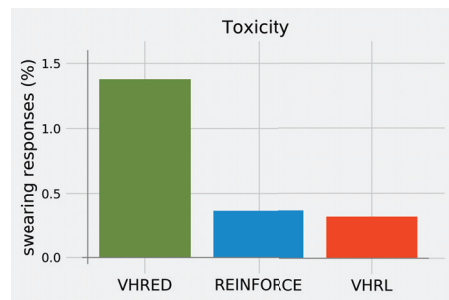

Figure 3: Training with RL to reduce toxicity decreases the percentage of generated utterances containing swear words.

the VHRL model trained to optimize for asking questions achieved the highest ratings, followed by VHRL minimizing repetition, and VHRL minimizing toxicity. This provides evidence that our proposed rewards lead to enhanced conversation quality as judged by human users, and that VHRL provides the most effective method for learning them.

7.1 Comparison of RL techniques for learning combined reward

As described in the previous section, optimizing for an overly simplistic metric (such as asking questions) can lead algorithms such as REINFORCE to trivially exploit the reward function at the expense of conversation quality. The five metrics proposed here do not fully encompass what it means to have a good conversation with a human user when optimized separately. Previous work found that optimizing individual metrics can actually reduce human judgments of conversation quality below the score of the MLE baseline (Jaques et al. 2019).

Therefore, instead of optimizing for individual metrics, we also train a variety of models to optimize for a combination of all five proposed rewards[2], making the reward function more complex and less easily exploited. The results are shown in Table 2, which is ordered from least to highest summed human ratings. All models proposed here, including the MLE baselines, outperform prior work by Jaques et

---

[2]The weights placed on each reward are given in the extended version of this paper (Saleh et al. 2019b)

Model	Quality	Fluency	Diversity	Contingency	Total	Chat Len.
Batch Ψ (Jaques et al. 2019)	2.17	3.89	3.13	1.98	11.17	11.44
Decoupled VHRL (ablation)	2.46	4.15	3.61	2.02	12.24	12.14
Transformer	2.62	4.17	3.23	2.34	12.36	11.28
REINFORCE	2.89	4.47	3.67	**2.80**	13.84	11.60
VHRED	2.84	4.53	**4.43**	2.47	14.27	10.94
VHRL (ours)	**2.91**	**4.65**	4.26	2.67	**14.49**	**12.84**

Table 2: Interactive human evaluation results comparing different RL training approaches optimizing for all five rewards, ordered by overall total rating score. Ratings are on a Likert scale (1-7).

Model	Quality	Fluency	Diversity	Cont.	Total	Chat Len
Similarity						
REINFORCE	<u>2.71</u>	<u>4.20</u>	<u>3.86</u>	<u>2.73</u>	<u>12.10</u>	12.36
VHRL	2.51	3.92	3.67	2.22	11.14	11.56
Sentiment						
REINFORCE	<u>2.80</u>	<u>4.55</u>	3.90	2.43	<u>12.43</u>	11.02
VHRL	2.72	4.30	<u>4.32</u>	<u>2.50</u>	12.28	<u>11.12</u>
Toxicity						
REINFORCE	2.71	4.12	4.06	2.55	11.98	10.74
VHRL	<u>2.76</u>	<u>4.58</u>	<u>4.34</u>	<u>2.64</u>	<u>12.82</u>	<u>11.32</u>
Repetition						
REINFORCE	2.74	4.02	4.28	2.30	11.92	<u>11.52</u>
VHRL	<u>3.00</u>	<u>4.39</u>	<u>4.41</u>	<u>2.84</u>	<u>13.12</u>	10.66
Question						
REINFORCE	2.39	4.08	2.45	2.31	9.80	**12.98**
VHRL	**3.27**	**4.86**	**4.47**	**2.88**	**14.14**	11.68

Table 3: Interactive human evaluation results comparing the proposed reward functions, REINFORCE, and VHRL. Ratings are on Likert scale (1-7). Higher is better.

al. (2019). The ablated version of our approach, *Decoupled VHRL*, performs poorly, suggesting our proposed joint training scheme for VHRL is an important component of the algorithm.

VHRED, REINFORCE, and VHRL all exceed the performance of the Transformer in terms of human judgments of conversation quality. While recent advances in language modeling with Transformer architectures (e.g. Radford et al. (2019)) are quite promising, translating these successes into conversational AI is still an ongoing area of research. Models such as VHRED have an architecture designed for the dyadic nature of the conversation (with an explicit update based on each conversation turn); in contrast, the Transformer has no special architectural features to denote conversation turns beyond the *<end of utterance>* token. Recurrent models have been shown to be more adept at capturing and exploiting hierarchical information in language (Tran, Bisazza, and Monz 2018). Further, Transformers have been shown to be less sensitive to the conversation history relative to recurrent models (Sankar et al. 2019). We have observed that the Transformer is highly repetitive, and vulnerable to monologuing rather than generating answers contingent on the user's input. Although it has a high score on the semantic similarity metric in Figure 2, Table 2 demonstrates that this does not translate into improved contingency ratings.

Finally, in comparing the RL techniques to the VHRED baseline on which they were based, we see that a naïve application of the REINFORCE algorithm does not lead to overall improvements in human ratings of conversation quality.

While the language generated by the REINFORCE model is less toxic and more positive, this comes at the cost of a slight reduction in overall conversation quality. In contrast, VHRL is the only technique that allows the model to optimize for reducing toxicity, improving sentiment, etc., while increasing the overall human judgments of the quality of the conversation. Note that the chat length is higher with VHRL, suggesting users are more interested and engaged when chatting with VHRL versus the other models. Thus, VHRL can be used to optimize for metrics that make the dialog model more safe and appropriate for a particular application domain, while maintaining the ability to have an enjoyable and engaging conversation with human users.

8 Conclusion

We have demonstrated that RL can be used to improve the outputs of an open-domain dialog model with respect to human-centered metrics of conversation quality. For example, RL can reduce the toxicity of the generated language, a problem that has previously hindered deployment of these systems to the real world. By developing metrics tailored to a particular application domain (for example, increasing politeness for a technical-support system), these techniques could be used to help open-domain dialog models integrate with real-world products. We have shown that our proposed VHRL technique is most effective for optimizing long-term conversation rewards, and for improving conversation quality while improving metrics like toxicity.

References

Bacon, P.-L.; Harb, J.; and Precup, D. 2017. The option-critic architecture. In *AAAI*.

Bodie, G.; St. Cyr, K.; Pence, M.; et al. 2012. Listening competence in initial interactions i: Distinguishing between what listening is and what listeners do. *I. J. of Listening* 26(1):1–28.

Budzianowski, P.; Ultes, S.; Su, P.-H.; et al. 2017. Sub-domain modelling for dialogue management with hierarchical reinforcement learning. *arXiv preprint arXiv:1706.06210*.

Cer, D.; Yang, Y.; Kong, S.-y.; et al. 2018. Universal sentence encoder. *arXiv:1803.11175*.

Colombo, P.; Witon, W.; Modi, A.; et al. 2019. Affect-driven dialog generation. *arXiv preprint arXiv:1904.02793*.

Curry, and Rieser. 2018. #metoo alexa: How cfonversational systems respond to sexual harassment. In *ACL W Ethics in NLP*, 7–14.

Deb, K. 2014. Multi-objective optimization. In *Search methodologies*. Springer. 403–449.

Dietterich, T. G. 2000. Hierarchical reinforcement learning with the maxq value function decomposition. *J. of AI Res.* 13:227–303.

Dziri; Kamalloo; Mathewson; et al. 2019. Evaluating coherence in dialogue systems using entailment. *arXiv:1904.03371*.

Felbo, B.; Mislove, A.; Søgaard, A.; et al. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *EMNLP*. ACL.

Ghandeharioun; Shen; Jaques; et al. 2019. Approx. interactive human eval. with self-play for open-domain dialog systems. *NeurIPS*.

Hancock, B.; Bordes, A.; Mazare, P.-E.; and Weston, J. 2019. Learning from dialogue after deployment: Feed yourself, chatbot! *arXiv:1901.05415*.

He, T., and Glass, J. 2018. Detecting egregious responses in neural sequence-to-sequence models. *arXiv preprint arXiv:1809.04113*.

Henderson, P.; Sinha, K.; Angelard-Gontier, N.; et al. 2018. Ethical challenges in data-driven dialogue systems. In *AAAI/ACM Conf. on AI, Ethics, and Society*, 123–129. ACM.

Huang; Zaiane; Trabelsi; and Dziri. 2018. Automatic dialogue generation with expressed emotions. In *NAACL (SP)*, 49–54.

Ireland; Slatcher; Eastwick; et al. 2011. Language style matching predicts relationship initiation and stability. *Psy. sci.* 22(1):39–44.

Jaques, N.; Ghandeharioun, A.; Shen, J.; et al. 2019. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*.

Kulkarni, T. D.; Narasimhan, K.; Saeedi, A.; and Tenenbaum, J. 2016. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *NIPS*, 3675–3683.

Li, J.; Monroe, W.; Ritter, A.; et al. 2016. Deep reinforcement learning for dialogue generation. In *EMNLP*, 1192–1202.

Li, J.; Monroe, W.; Shi, T.; et al. 2017. Adversarial learning for neural dialogue generation. In *EMNLP*, 2157–2169.

Liu, B., and Lane, I. 2017. Iterative policy learning in end-to-end trainable task-oriented neural dialog models. In *IEEE Auto. Speech Rec. and Understanding W. (ASRU)*, 482–489. IEEE.

Liu, C.; Lowe, R.; Serban, I.; et al. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *EMNLP*, 2122–2132.

Lowe, R.; Noseworthy, M.; Serban, I. V.; et al. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. *arXiv preprint arXiv:1708.07149*.

Miller, A.; Feng, W.; Fisch, A.; et al. 2017. Parlai: A dialog research software platform. *arXiv:1705.06476*.

Nachum, O.; Gu, S. S.; Lee, H.; and Levine, S. 2018. Data-efficient hierarchical reinforcement learning. In *NIPS*, 3303–3313.

Peng, B.; Li, X.; Li, L.; et al. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. *arXiv preprint arXiv:1704.03084*.

Precup, D. 2001. Temporal abstraction in reinforcement learning.

Radford, A.; Wu, J.; Child, R.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1:8.

Ranzato, M.; Chopra, S.; Auli, M.; et al. 2015. Sequence level training with recurrent neural networks. *arXiv:1511.06732*.

Saleh, A.; Baly, R.; Barrón-Cedeño, A.; et al. 2019a. Team qcri-mit at semeval-2019 task 4: Propaganda analysis meets hyperpartisan news detection. *arXiv preprint arXiv:1904.03513*.

Saleh, A.; Jaques, N.; Ghandeharioun, A.; Shen, J. H.; and Picard, R. 2019b. Hierarchical reinforcement learning for open-domain dialog. *arXiv preprint arXiv:1909.07547*.

Sankar, C.; Subramanian, S.; Pal, C.; et al. 2019. Do neural dialog systems use the conversation history effectively? an empirical study. *arXiv preprint arXiv:1906.01603*.

See, A.; Roller, S.; Kiela, D.; and Weston, J. 2019. What makes a good conversation? how controllable attributes affect human judgments. *arXiv preprint arXiv:1902.08654*.

Serban, J.; Sankar, C.; Germain, M.; et al. 2017a. A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*.

Serban, J.; Sordoni; Lowe; et al. 2017b. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*.

Shin, J.; Xu, P.; Madotto, A.; and Fung, P. 2019. Happybot: Generating empathetic dialogue responses by improving user experience look-ahead. *arXiv preprint arXiv:1906.08487*.

Su; Budzianowski; Ultes; et al. 2017. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *SIGdial Mtg on Discourse and Dialogue*, 147–157.

Sutton, and Barto. 2018. *Rl: An introduction*. MIT press.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112(1-2):181–211.

Tang, D.; Li, X.; Gao, J.; et al. 2018. Subgoal discovery for hierarchical dialogue policy learning. *arXiv preprint arXiv:1804.07855*.

Tessler, C.; Givony, S.; Zahavy, T.; et al. 2017. A deep hierarchical approach to lifelong learning in minecraft. In *AAAI*.

Tran, K.; Bisazza; and Monz. 2018. The importance of being recurrent for modeling hierarchical structure. *arXiv:1803.03585*.

Vaswani, A.; Shazeer, N.; Parmar, N.; et al. 2017. Attention is all you need. In *NIPS*, 5998–6008.

Vezhnevets, A.; Osindero; Schaul; et al. 2017. Feudal networks for hierarchical reinforcement learning. In *ICML*, 3540–3549.

Wallace, E.; Feng, S.; Kandpal, N.; et al. 2019. Universal adversarial triggers for nlp. *arXiv preprint arXiv:1908.07125*.

Weger Jr, H.; Castle, G. R.; and Emmett, M. C. 2010. Active listening in peer interviews: The influence of message paraphrasing on perceptions of listening skill. *The I. J. of Listening* 24(1):34–49.

Williams, R. 1992. Simple statistical gradient-following algorithms for connectionist rl. *ML* 8(3-4):229–256.

Xu; Wu; and Wu. 2018. Towards explainable and controllable open domain dialogue generation with dialogue acts. *arXiv:1807.07255*.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*.

Zhang; Zhao; and Yu. 2018. Multimodal hierarchical rl policy for task-oriented visual dialog. *arXiv:1805.03257*.

Zhou, L.; Gao; Li; and Shum. 2018. The design and implementation of xiaoice, an empathetic social chatbot. *arXiv:1812.08989*.