WeaSuL$^{\pi}$: Weakly Supervised Dialogue Policy Learning: Reward Estimation for Multi-turn Dialogue

Anant Khandelwal
India Machine Learning
Amazon
anantkha@amazon.com

Abstract

An intelligent dialogue system in a multi-turn setting should not only generate the responses which are of good quality, but it should also generate the responses which can lead to long-term success of the dialogue. Although, the current approaches improved the response quality, but they over-look the training signals present in the dialogue data. We can leverage these signals to generate the weakly supervised training data for learning dialog policy and reward estimator, and make the policy take actions (generates responses) which can foresee the future direction for a successful (rewarding) conversation. We simulate the dialogue between an agent and a user (modelled similar to an agent with supervised learning objective) to interact with each other. The agent uses dynamic blocking to generate ranked diverse responses and exploration-exploitation to select among the Top-K responses. Each simulated state-action pair is evaluated (works as a weak annotation) with three quality modules: Semantic Relevant, Semantic Coherence and Consistent Flow. Empirical studies with two benchmarks indicate that our model can significantly out-perform the response quality and lead to a successful conversation on both automatic evaluation and human judgement.

1 Introduction

Dialog policy for multi-turn dialogue decides the next best action to take on the environment so as to complete the conversation based on various success criteria. Reinforcement learning can help to learn such a policy where the environment can be users (human or model) and the policy takes action on the environment from which it gets a reward signal (Fatemi et al., 2016; Peng et al., 2017; Chen et al., 2017; Yarats and Lewis, 2018; Lei et al., 2018; He et al., 2018; Su et al., 2018).

Learning a dialogue policy using reinforcement learning can be challenging with humans users, since it requires a large set of samples with a reward to train. Since there are a lot of previous works on neural response generation (Gu et al., 2020; Zhao et al., 2020; Zhang et al., 2019; Xing et al., 2018; Serban et al., 2016) we can model the users also, using any of these encoder-decoder architectures. This helps to simulate the conversations between the simulated user and the agent (policy model) replying to each other (Zhao and Eskenazi, 2016; Dhingra et al., 2016; Shah et al., 2018). Reward signal for policy learning can be as simple as the small constant negative reward at each turn and a large reward at the end (if the goal completes) to encourage shorter conversations (Takanobu et al., 2019).

However, reward estimation for dialogue is challenging, the small constant negative reward at each turn may lead to ending the conversation prematurely. Instead of handcrafting the reward at the end based on success or failure, it is more useful if we can evaluate reward at every turn to guide the policy to dynamically change actions as per the need for the user and end the conversation naturally. With the growing complexity of the system across different topics, it is required to build a more sophisticated reward function to avoid manual intervention for accounting different factors towards conversation success.

In this work, we proposed a novel model for contextual response generation in multi-turn dialogue. The model includes the turn-level reward estimator, which combines the weak supervision signals obtained from three basic modules 1) Semantic Coherence, 2) Consistent Flow, 3) Semantic Relevance. These modules are learned jointly with the response generation model with the counterfactual examples obtained from negative sampling. Leveraging the weak supervision signals obtained from these models, we further update the reward estimator and dialog policy jointly in an alternative way, thus improving each other.

Proceedings of the 14th International Conference on Natural Language Generation (INLG), pages 64–75,
Aberdeen, Scotland, UK, 20-24 September 2021.

Our proposed approach integrates semantic understanding of utterances using encoder-decoder systems with the power of Reinforcement Learning (RL) to optimize long-term success. We test the proposed approach with two benchmarks: DailyDialog (Li et al., 2017b) and PersonaChat (Zhang et al., 2018). Experimental results demonstrate on both datasets indicate that our model can significantly outperform state-of-the-art generation models in terms of both automatic evaluation and human judgment.

2 Related Work

Open-domain dialogue in a multi-turn setting has been widely explored with different encoder-decoder architectures (Gu et al., 2020; Feng et al., 2021; Kottur et al., 2017; Li et al., 2016; Shah et al., 2018; Shang et al., 2015; Vinyals and Le, 2015; Wu et al., 2019; Zhao et al., 2020; Zhong et al., 2019). The basic encoder-decoder architectures like Seq-to-Seq models have been widely extended and modified to generate the generic responses, context modelling and grounding by persona/emotion/knowledge (Li et al., 2015; Xing et al., 2017; Serban et al., 2016; Xing et al., 2018; Zhang et al., 2019, 2018; Zhou et al., 2018; Dinan et al., 2018).

The dialogue literature widely applies reinforcement learning, including the recent ones based on deep architectures (Takanobu et al., 2019, 2020; Li et al., 2020; Takanobu et al., 2020; Li et al., 2020; Gordon-Hall et al., 2020a,b). But these task-oriented RL dialogue systems often model the dialogue with limited parameters and assumptions specific to the dataset, targeted for that task. The dataset includes hand-built templates with state, action and reward signals designed by humans for each new domain making this setting difficult for extending these to open domain dialogue systems.

Our goal in this work is to integrate the state-of-the-art encoder-decoder architectures like in Gu et al. (2020); Zhao et al. (2020); Csaky and Recski (2020) and reinforcement learning paradigms to efficiently learn the dialogue policy optimized for long-term success in the multi-turn dialogue scenarios. We are recently inspired by the works in Takanobu et al. (2019); Li et al. (2020, 2016) to jointly learn the reward function and dialogue policy, and reduce the effort and cost for manual labelling the conversations for building the reward model. Specifically, we leverage the weak supervision inspired from Chang et al. (2021a,b) to generate the labelled dataset to facilitate this joint learning and building reward estimation model.

3 Approach

We represent dialog sessions $\mathcal{D} = \{\tau_1, \tau_2, \tau_3, ...\tau_n\}$ where each dialog session τ represents the trajectory of state-action pairs as $\{s_0^u, a_0^u, s_0, a_0, s_1^u, a_1^u, s_1, a_1, ...\}$. The user in our case is a simulator which utters a response a^u given the state s^u denoted as $\mu(a^u, e^u|s^u)$ where e^u denotes the binary signal indicating the end of a dialog session, in that case the response a^u is empty. The dialog policy $\pi_\theta(a|s)$ decides the action a according to the current state s after the agent interacts with the user simulator μ. At each time, the state given to the either dialog party is updated after recording the action uttered by the other party. The reward estimator f evaluates the quality of response/action uttered by the dialog policy π. The dialog policy π is based on the BERT (Devlin et al., 2019) encoder-decoder model and the reward function f is the MLP model parameterized by θ and ω respectively. We have modeled the user simulator exactly in the same way as the agent but trained only using supervised learning objective.

In the subsequent section, we will introduce the components action, state, policy, quality modules and reward estimator. Further, sections explain the setup we have used for weakly supervised learning and, finally, the experimental results.

3.1 Action

An action a is the dialogue utterance generated by the encoder-decoder model as shown in Figure 1. The model takes as input the context history (state), and outputs the probability distribution over a set of possible actions denoted as $\pi_\theta(a|s)$ parameterized by θ. The user simulator generates the action a^u, policy generates the action a, and the input state for the agent and the user is s and s^u respectively.

3.2 State

The state is the past conversation history between an agent and a user denoted as, $s_t = \{q_1, a_1, q_2, a_2, q_3, a_3, ..., q_t\}$. The state for an agent and a user are differently denoted as s and s^u respectively. Let's say the agent utterances are denoted by a's, then state, $s = s_t$ and the agent utters a_t. Similarly, the user state

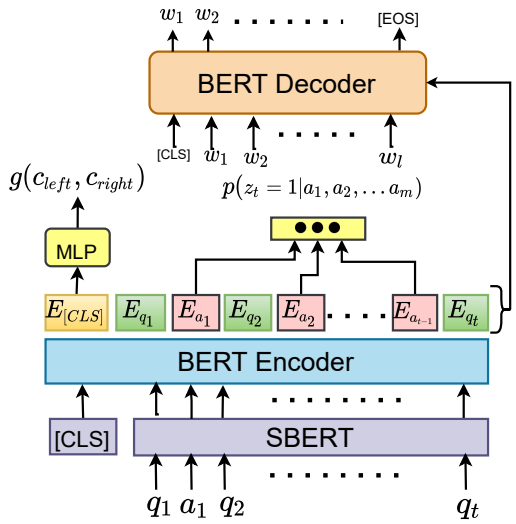

Figure 1: BERT based Encoder-Decoder with Semantic Coherence and Relevance. Similarly, Consistent Flow loss is also calculated using encoder.

$s_t^u = \{q_1, a_1, q_2, a_2, q_3, a_3, ..., q_t, a_t\}$ and the user utters q_{t+1}. Each of the utterances is mapped to a fixed-length sentence vector using SBERT (Reimers and Gurevych, 2019).

3.3 Dialogue Policy

The dialogue policy takes the form of a BERT based encoder-decoder (i.e. $\pi_\theta(a|s)$) (Gu et al., 2020) as shown in Figure 1. Similar to Xu et al. (2020), we have used the BERT based encoder and transformer decoder, but instead of feeding the utterance at word level, we instead fed the utterance representation (obtained from SBERT) into the encoder. The encoder takes as input the previous context history as s_t and output the response a_t at the output of the decoder.

3.4 User Simulator

We have modelled the user simulator in exactly the same way as the BERT based encoder-decoder shown in Figure 1. However, the user simulator is trained only (with supervised learning objective) for utterances in dialog corpus and predicting user response (Gu et al., 2020).

3.5 Conversation Quality Modules

We calculate the reward for each state-action pair (see Section. 3.8) and use this signal to train the dialogue policy so that it can avoid reaching bad states so as to reach the successful end of the conversation between a user and an agent. We have leveraged the signals from three basic modules, namely, Semantic Coherence, Consistent Flow and Semantic Relevance (which are jointly learned with the dialogue policy). For each of the three modules, the data for the positive class is obtained from the source corpus while for the negative class it has been generated dynamically during training. We describe each of the three modules in the following sections.

3.5.1 Semantic Relevance

We need to filter out the utterances generated with high confidence by the dialog policy but are semantically irrelevant to the previous context. To quantify such a characteristic, we modeled the general response relevance prediction task which utilizes the sequential relationship of the dialog data fed to the encoder side of BERT encoder-decoder framework. Since, the task of semantic relevance is to match the two sequences of conversation, so instead of matching the context and response, we have measured the relevance of two fragments of dialogue session.

Specifically, given a context $c = \{q_1, a_1, q_2, a_2, ...q_m\}$, we randomly split c into two consecutive pieces $c_{left} = \{q_1, a_1, q_2, a_2, ...q_t, a_t\}$ and $c_{right} = \{q_{t+1}, a_{t+1}, ...q_m\}$. Similar to Xu et al. (2020), we replaced the left or right part with the sampled piece from the corpus. Also, we additionally generate the negative samples by internal shuffling in the left or right part. The whole model is trained like a classifier with corresponding labels $y_{sr} \in \{0, 1\}$. Since the individual utterances are fed after obtaining their vector representation, the aggregated representation of two pieces is represented by E^{sr}_{CLS} over which the non-linear transformation is applied, the score for semantic relevance is given by $g(c_{\text{left}}, c_{\text{right}})$, and similar to Xu et al. (2020), it has been trained using the binary cross-entropy loss as:

$$L_{sr} = -y_{sr} \log(g(c_{\text{left}}, c_{\text{right}})) - (1 - y_{sr}) \log(1 - g(c_{\text{left}}, c_{\text{right}})) \quad (1)$$

3.5.2 Semantic Coherence

The response generated should be rewarded only if it is coherent despite having adequate content. This makes the model to generate the coherent responses while avoiding the incoherent ones. Specifically, given a context $c = \{q_1, a_1, q_2, a_2, ...q_m\}$, we randomly select any of the agent response at time t, denoted as a_t, and replace it with any random

utterance from the corpus. We also generate the incoherent samples by internal shuffling of bi-grams. The incoherent utterance is labelled as $y_t^{coh} = 0$ and coherent samples as $y_t^{coh} = 1$. The semantic coherence model is also trained like a classifier for each of the utterance representations obtained at the output of BERT encoder as shown in Figure 1. The probability of the t-th utterance being incoherent is given as:

$$p(z_t = 1|a_1, .., a_t) = softmax(W_{coh}E_{a_t}+b_{coh}) = \frac{exp(W_{coh}E_{a_t} + b_{coh})}{\sum_{l=1}^{m} exp(W_{coh}E_{a_l} + b_{coh})} \quad (2)$$

and the loss function is given as:

$$L_{coh} = -\sum_{t=1}^{m} z_t \log p(z_t = 1|a_1, a_2.....a_m) \quad (3)$$

3.5.3 Consistent Flow

We want the agent to continuously add the information to keep the conversation going in the forward direction. To determine the flowing conversation, we take the cosine similarity between the last two agent utterances denoted as $E_{a_{i-1}}$ and E_{a_i} denoted as $g(a_{i-1}, a_i)$, and we measure the similarity with randomly sampled utterance v in place of a_{i-1} given as $g(a_{i-1}, v)$. We would like $g(a_{i-1}, a_i)$ to be larger than $g(a_{i-1}, v)$ by at least a margin Δ and define the learning objective as a hing loss function:

$$L_{cf} = max\{0, \Delta - g(a_{i-1}, a_i) + g(a_{i-1}, v)\} \quad (4)$$

3.6 Joint Training of Agent and Reward Modules

To initialize the parameters of agent and reward modules $\mathcal{M}$ ={Semantic Relevance, Semantic Coherence, Consistent Flow}, we used the supervised learning objective since all the state-action pairs obtained from the pre-training corpus are the ground-truth and can be used as close approximation for further fine-tuning on other dialog corpus. We used the pre-training corpus $\mathcal{P}$ as Gutenberg dialog corpus (Csaky and Recski, 2020). Since the agent model in our case is based on BERT encoder-decoder parameterized by θ similar to Gu et al. (2020), the probability of generating agent's response **a** is given as:

$$p_\theta(\mathbf{a}|\mathbf{s}) = \prod_{j=1}^{N} p_\theta(a_j|a_{<j}, \mathbf{s}), \quad (5)$$

where a_j is the j-th word generated at the output of decoder and **s** is the whole context history utterances fed to the encoder and N is the maximum sequence length of decoder. The loss function for generating agent response **a** is given as:

$$L_a = J(\theta) = -\sum_{i=1}^{N} \log p_\theta(a_j|a_{<j}, \mathbf{s}) \quad (6)$$

The joint loss function is defined as:

$$L_{full} = L_a + \alpha * (L_{sr} + L_{coh} + L_{cf}) \quad (7)$$

The policy π_θ is also parameterized by θ, and the probability of action a is given by $\pi_\theta(a|s)$ similar to $p_\theta(a|s)$, since the probability distribution is learned only from (s, a) pairs obtained from the corpus with human demonstrations. It is a good approximation to initialize the parameters of policy $\pi_\theta(a|s)$ with parameters of $p_\theta(a|s)$. Furthermore, we update the policy π_θ (Step 13 in the Algorithm. 1) to avoid actions a which do not lead to rewarding conversations.

3.7 Dialogue Simulation between Agent and User

We setup simulation between virtual agent and user, and let them take turns talking to each other. The simulation is started with a starter utterance obtained from the dialog samples D_H (Step 5 of Algorithm 1) and fed to the agent, it then encodes the utterance and generates the response a, the state s^u is then updated with previous history and fed to the user model to obtain the next response a^u. The response a^u is appended to s^u to obtain the updated state s. Similarly, the process is repeated until one of the following conditions occurs after a few number of turns[1]: a) When agent starts to produce dull responses like "I don't know" [2]. b) When agent starts to generate repetitive response consecutively [3] c) Or, the conversation achieved the maximum number of turns handled by agent and user models.[4]

[1]The number of turns after these rules applied is average number of turns in the corpus

[2]Used simple rule matching method with 9 phrases collected from the corpus, instead of having false positives and negatives this works well in practice.

[3]If by rule two consecutive utterances matched more than 80% it is said to be repetitive.

[4]The maximum number of turn is set as 20.

3.8 Weakly Supervised Learning Algorithm

Learning with weak supervision is widely used with the rise of data-driven neural approaches (Ratner et al., 2020; Mrkšić et al., 2017; Chang et al., 2020; Bach et al., 2017; Wu et al., 2018; Chang et al., 2021a). Our approach incorporates a similar line of work by providing noisy text to a pre-trained model which incorporates prior knowledge from general-domain text and small in-domain text (Peng et al., 2020; Chen et al., 2019; Harkous et al., 2020) and use it as a weak annotator similar to Ratner et al. (2020). The primary challenge with the synthetic data is the noise introduced during the generation process, and the noisy labels tend to bring little to no improvement (Frénay and Verleysen, 2013). To train on such noisy data, we employ three step training process: a) pre-training b) generate data with weighted categories c) fine-tuning similar to Chang et al. (2021a); Dehghani et al. (2017).

Step 1: Pre-train Generation and Quality Modules Jointly. This step involves pre-training the agent with quality modules jointly as explained in Section 3.6. Quality modules trained on clean data as well as automatically generated negative samples by random sampling. These modules are further fine-tuned on the sampled dialogues from target dialogue corpus at each training iteration. Similarly, we initialized the user also by supervised training on the pre-training dialogue corpus with fine-tuning on target dialogue corpus. (see steps 2-7 of Algorithm 1). The fine-tuning steps make use of continual learning to avoid catastrophic forgetting (Madotto et al., 2020; Lee, 2017).

Step 2: Generates the Weakly Labelled data with Reward categories. After the models are initialized with trained parameters, the dialogue simulation has been started between the agent and the user (see Section. 3.7) to interact with each other and generates the synthetic data with annotated scores with each quality module for every state-action pair in sampled dialogues. During dialogue simulation, we employ Dynamic Blocking mechanism(Niu et al., 2020) to generate novel words and paraphrased responses. Specifically, we generate Top-7 response at each turn and set the agent to exploration for 60 percent of the times and for the rest of the times it exploits by selecting the response from top two ranked responses. We specifically filter the state-action pairs into three reward categories namely, *VeryHigh*, *High* and *Low*. For the state-action pairs whose scores by each module are greater than or equal to 0.8 are put into the *VeryHigh* category. Other, state-action pairs whose scores by each module are between 0.6 and 0.8 are put into the *High* reward category. The rest of all state-action pairs are put into the *Low* reward category. Additionally, we include state-action pairs sampled from target dialog corpus in Step 1. into the *VeryHigh* category.

Step 3: Update the reward estimator and policy. The reward estimator maximizes the log likelihood state-action pairs of higher rewards than the lower ones. The reward estimator f_ω, parameterized by ω, and let's say H, V and L represents the collection of all state action pairs of *High*, *VeryHigh* and *Low* reward category respectively.

$$\begin{aligned}
\omega^* &= \arg\max \mathbb{E}_{(s_k,a_k)\sim\{H,V,L\}}[f_\omega(s_k,a_k)] \\
f_\omega(s_k,a_k) &= \log p_\omega(s_k,a_k) = \log \frac{e^{R_\omega(s_k,a_k)}}{Z_\omega} \\
R_\omega(s_k,a_k) &= \sum_{t=k}^{T} \gamma^{t-k} r_\omega(s_t,a_t) \\
Z_\omega &= \sum_{\forall(s_k,a_k)} e^{R_\omega(s_k,a_k)}
\end{aligned} \tag{8}$$

where f models state-action pairs of H, V and L category as a Boltzmann distribution (Takanobu et al., 2019). The cost function for reward estimator in terms of trajectories obtained from respective reward categories is given as:

$$\begin{aligned}
J_f(\omega) = &-0.5 * KL(p_H(s,a) \parallel p_\omega(s,a)) \\
&- KL(p_V(s,a) \parallel p_\omega(s,a)) \\
&+ KL(p_L(s,a) \parallel p_\omega(s,a))
\end{aligned} \tag{9}$$

It minimize the KL-divergence between reward distribution and the state-action pairs of high and very high reward but maximize the distribution from the ones with low category. The gradient yields:

$$\begin{aligned}
\nabla_\omega J_f = 0.5 * \mathbb{E}_{(s,a)\sim H}[\nabla_\omega f_\omega(s,a)] \\
+\mathbb{E}_{(s,a)\sim V}[\nabla_\omega f_\omega(s,a)] - \mathbb{E}_{(s,a)\sim L}[\nabla_\omega f_\omega(s,a)]
\end{aligned} \tag{10}$$

Since, the dialog policy is required to put the actions atleast to that of high category, i.e. maximize the entropy regularized expected reward ($\mathbb{E}_\pi[R] + H(\pi)$) which is effectively minimizes

the KL divergence between the policy distribution and Boltzmann distribution.

$$J_\pi(\theta) = -KL(\pi_\theta(a|s) \parallel p_\omega(s,a)) \\ = \mathbb{E}_{(s,a)\sim\pi}[f_\omega(s,a) - \log \pi_\theta(a|s)] \\ = \mathbb{E}_{(s,a)\sim\pi}[R_\omega(s,a)] - \log Z_\omega + H(\pi_\theta) \quad (11)$$

where the term $\log Z_\omega$ is independent to θ, and $H(\cdot)$ denotes the entropy of a model. Using likelihood ratio trick the gradient for policy is given as:

$$\nabla_\theta J_\pi = \mathbb{E}_{(s,a)\sim\pi}[(f_\omega(s,a) \\ - \log \pi_\theta(a|s))\nabla_\theta \log \pi_\theta(a|s)]. \quad (12)$$

Hence, the reward is $r_\omega(s,a) = f_\omega(s,a) - \log \pi_\theta(a|s)$ for each state-action pair and the loss function re-written as:

$$J_\pi(\theta) = \mathbb{E}_{(s,a)\sim\pi}[\sum_{k=t}^{T} \gamma^{k-t}(f_\omega(s_k,a_k) \\ - \log \pi_\theta(a_k|s_k))] \quad (13)$$

Like in Takanobu et al. (2019) the reward estimator f_ω includes the shaping term. Formally, we include next state s_{t+1} also instead of just (s_t, a_t)

$$f_\omega(s_t, a_t, s_{t+1}) = g_\omega(s_t, a_t) + \gamma h(s_{t+1}) - h(s_t) \quad (14)$$

where h is the MLP network with input as pre-sigmoid scores from each quality modules, and g_ω is also the MLP network with input as the concatenation of E_{CLS} as state vector and SBERT sentence embedding of action a.

Algorithm 1 Dialogue Policy Learning

Require: Pre-Training corpus P, Dialogue Corpus $\mathcal{D}$.

1: Modules $\mathcal{M}$ = {Semantic Relevance, Semantic Coherence, Consistent Flow}
2: Do Agent training on $\mathcal{P}$ as in Section 3.6 jointly with modules $\mathcal{M}$
3: User μ supervised training on $\mathcal{P}$.
4: **for each** *training iteration* **do**
5: Sample dialogues $\mathcal{D}_H$ from $\mathcal{D}$ randomly.
6: Fine-tune user simulator μ on $\mathcal{D}_H$.
7: Fine-tune Agent and $\mathcal{M}$ on $\mathcal{D}_H$ jointly.
8: Collect dialog samples $\mathcal{D}_\pi$ by executing the dialog policy π and interacting with μ, $a^u \sim \mu(\cdot|s^u)$, $a \sim \pi(\cdot|s)$ where s and s^u is updated each time after getting response from user and agent respectively.
9: Get weak annotation scores for all $(s,a) \in \mathcal{D}_\pi$ from each of the modules $\mathcal{M}$.
10: Filtering the (s,a) pairs into {*VeryHigh*, *High* and *Low*} reward categories.
11: Update the reward estimator f by minimizing J_f w.r.t ω (Eq.10)
12: Compute reward for each $(s,a) \in \mathcal{D}_\pi$ as,

$$\hat{r} = f_\omega(s_t, a_t, s_{t+1}) - \log \pi(a_t|s_t)$$

13: Update the policy π_θ by minimizing J_π w.r.t θ (Eq. 13).
14: **end for**

4 Experiments

We conduct experiments on DailyDialog (Li et al., 2017b), PersonaChat (Zhang et al., 2018) and used Gutenberg Dialogue Dataset (Csaky and Recski, 2020) as a pre-training corpus. We compare our model performance with baselines on various aspects of response quality.

4.1 Datasets

We considered DailyDialog (Li et al., 2017b) and PersonaChat (Zhang et al., 2018) which are open domain dialog corpus to evaluate our system. DailyDialog contains conversation revolving around various topics pertaining to daily life, and PersonaChat contains conversations between people with their respective persona profiles. These dialogues can be of varying length, we limit the maximum length to 20, that can be fed to the BERT Encoder-Decoder model. Since average length of DailyDialog is 7.9 and that of PersonaChat is 9.4, so most of the dialogues fit easily without truncation from the history. For rest of the dialogues, it can be slided across to include the more recent utterances and remove it from the starting. Since we are mapping the utterances to their corresponding vectors using SBERT, the length of individual utterances truncated automatically and retain only first 512 word pieces in case of longer utterances. For pre-training corpus the vocabulary is limited to 100,000 while the vocabularies for DailyDialog and PersonaChat are 25,000 and 32,768 respectively.

4.2 Baselines

We select various multi-turn response generation baselines. The baselines which are not included pre-training are (1) **HRED** : Hierarchical encoder-decoder framework (Serban et al., 2016)

(2) **VHRED** : an extension of HRED that generates response with latent variables (Serban et al., 2017) (3) **HRAN** : Hierarchical attention mechanism based encoder-decoder framework (Xing et al., 2018) (4) **ReCoSa** : Hierarchical transformer based model (Zhang et al., 2019) (5) **SSN**: dialogue generation learning with self-supervision signals extracted from utterance order (Wu et al., 2019) (6) **Transformer-Auxiliary Tasks**: A recent state-of-the are model leaning language generation with joint learning of transformer with auxiliary tasks (Zhao et al., 2020). The another two baselines from Csaky and Recski (2020) which involve pre-training on the Gutenberg corpus are: (1)**Transformer** : 50M parameters version and (2) **GPT-2** : Pre-trained model with version of 117M parameters. The repository[5] contains these two trained models.

4.3 Evaluation Metrics

We evaluate the performance of our model on various aspects of response quality using both automatic and human evaluation. Although, most of the automatic metrics poorly correlate with human evaluation (Liu et al., 2016), and the recently proposed metrics (Li et al., 2017a; Lowe et al., 2017; Tao et al., 2018) are harder to evaluate than perplexity and BLEU (Papineni et al., 2002). Additionally, human evaluation has its inherent limitation of bias, cost and replication difficulty (Tao et al., 2018). Due to this consensus, some used only automatic metrics (Xing and Fernández, 2018; Xu et al., 2018b) and some used only human evaluation (Krause et al., 2017; Fang et al., 2018) while some used both (Shen et al., 2018; Xu et al., 2018a; Baheti et al., 2018; Ram et al., 2018).

We mainly used the automatic metrics using the DIALOG-EVAL repository[6], it contains 17 different metrics, but we measure only a few metrics to facilitate the comparison with the published baselines results. We specifically follow (Zhao et al., 2020) to measure automatic evaluation and human evaluation. For response content quality we measured BLEU-4 (Papineni et al., 2002) and Perplexity(PPL) (Sutskever et al., 2014). Like in Zhao et al. (2020) used embedding metrics average (AVG), extrema (EXT), and greedy (GRE) measuring similarity between response and target embedding. Similar to Zhao et al. (2020) we also measured the informativeness of responses with distinct-1 and distinct-2 that are calculated as the ratios of distinct unigrams and bigrams.

Since our main objective is not to judge the response quality but to predict the response for long-term success of dialogue. We follow the guidelines as in Li et al. (2016) to explore both single-turn and multi-turn settings. We picked 500 dialogues from the test set and asked 3 native speakers for their judgement. In the first setting, we asked judges to pick the better response among the one generated by our model and a baseline model (**Pre-Trained GPT2**) based on various criteria like answerability and semantics. In the second setting, in case of multi-turn we used 200 simulated conversations between RL agent and a user model to judge the whole conversation for responses uttered by agent. In a complete end-to-end conversation we asked the judges to decide which of the simulated conversations are of higher quality. To compare against the RL model we employ baseline model to simulate the 200 conversations with the same starter utterance used by RL model. Automatic and Human evaluation are shown in Table. 1 and 2 respectively.

4.4 Results and Discussions

Table. 1 reports automatic evaluation metrics on the baseline and the proposed model. Our model outperforms for most of the metrics on both datasets. Since our main idea is to generate the responses for successful conversation in the long run than just evaluating the response quality at each of the turn. This is the main reason of why our model outperforms on both distinct-1 and distinct-2 metrics, in comparison to Transformer-auxiliary task model which also trained jointly with the similar tasks but lacks fine-tuning with the weak supervision signals indicate that an additional training with weakly labelled data improves the generalization performance. Although, we see the perplexity also improves since our model is generating the responses more like humans to optimize the conversation in long run. Similarly, embedding metrics also shown the improvement but little on average since it capturing the sense but due to length mismatch which occurs owing to the fact that our model is generating more novel words with futuristic sense. However, Distinct-{1,2} scores shows improvement because of the large pre-trained vocabulary, it gives the model more flexibility to generate novel words without disturbing the sense of the sentence.

[5]https://github.com/ricsinaruto/gutenberg-dialog
[6]https://github.com/ricsinaruto/ dialog-eval

Dataset	Model	PPL	BLEU	Distinct-1	Distinct-2	Average	Greedy	Extrema
DailyDialog	HRED	56.22	0.535	1.553	3.569	81.393	65.546	48.109
	HRAN	47.23	0.447	1.953	7.400	83.460	67.239	49.599
	VHRED	44.79	0.997	1.299	6.113	83.866	67.186	48.570
	SSN	44.28	1.250	2.309	7.266	72.796	73.069	44.260
	ReCoSa	42.34	1.121	1.987	10.180	84.763	67.557	48.957
	Transformer-Auxiliary Tasks	38.60	1.658	3.457	14.954	85.224	69.518	49.069
	Pre-Trained Transformer	-	11.5	2.92	14.7	55.1	53.5	59.8
	Pre-Trained GPT2	-	12.8	4.07	25.9	56.8	54.0	59.6
	Our Model	**20.13**	**15.171**	**6.316**	**28.422**	85.417	**73.118**	**61.539**
	Our Model w/o weak supervision	20.51	14.718	4.611	26.752	**86.481**	73.003	59.911
PersonaChat	HRED	46.04	1.279	0.164	0.450	83.329	65.546	48.109
	HRAN	41.94	1.997	0.235	0.771	82.850	67.239	49.599
	VHRED	42.07	2.181	0.312	1.915	82.995	67.186	48.570
	SSN	47.90	2.288	0.637	2.623	85.002	73.069	44.260
	ReCoSa	34.19	2.258	0.915	4.217	83.963	67.557	48.957
	Transformer-Auxiliary Tasks	33.23	2.434	1.279	5.816	83.632	69.518	49.069
	Pre-Trained Transformer	-	15.5	1.04	4.8	51.3	57.5	57.1
	Pre-Trained GPT2	-	15.3	1.82	12.9	53.6	55.9	55.8
	Our Model	**19.78**	**16.651**	**2.434**	**13.912**	84.941	**73.081**	**59.241**
	Our Model w/o weak supervision	21.49	16.017	2.318	13.274	**85.018**	72.438	58.816

Table 1: Automatic metrics comparison with baselines. Results in bold indicate the best performing model on the corresponding metrics.

DailyDialog			
Setting	**RL-Win**	**RL-Lose**	**Tie**
Single-Turn general quality	0.41	0.28	0.31
Single-Turn ease to answer	0.55	0.12	0.33
Multi-turn general quality	0.76	0.13	0.11
PersonaChat			
Setting	**RL-Win**	**RL-Lose**	**Tie**
Single turn general quality	0.36	0.22	0.42
Single-Turn ease to answer	0.51	0.14	0.35
Multi-turn general quality	0.71	0.17	0.12

Table 2: Human Evaluation Results. Ratios are calculated after taking majority vote among the decisions made by three judges.

We also note the results for our model without weak supervision training, namely, **Our Model w/o Weak Supervision**, this model just fine-tunes on the DailyDialog (Li et al., 2017b) and PersonaChat (Zhang et al., 2018) without generating the weak labelled data. Clearly, the distinct-1 and distinct-2 metrics are lower than the proposed model, because the model tends to generate the repetitive words more frequently. Similarly, the embedding metrics and PPL does not show any improvement over the proposed model except on embedding metric based on Average. However, it performs well on BLEU scores since it learns well to reproduce the responses as in the ground truth but not optimized for a successful conversation in the long run.

Table 1 also reports the results of another two baselines which are pre-trained models on Gutenberg Dialogue Corpus (Csaky and Recski, 2020). These models are fine-tuned on DailyDialog and PersonaChat dataset respectively. These models although improved much on BLEU scores and distinct-1 and distinct-2 scores since it gets the larger vocab and more enhanced training for learning the language structure. But lags in the embedding metrics indicating the response quality is low.

Table 2 reports the human evaluation results, the objective for which our model training is to generate the response for a successful conversation in the long run for the multi-turn scenario. Clearly, the evaluation results are up to our expectation, since the RL system does not bring a significant boost in single-turn response quality than the case of multi-turn setting.

5 Conclusions

We proposed a weak supervision framework for policy and reward estimation for long-term success of the dialogue by simulating the conversation between a virtual agent and user. Empirical studies

on two benchmarks proves the effectiveness of our approach.

References

Stephen H Bach, Bryan He, Alexander Ratner, and Christopher Ré. 2017. Learning the structure of generative models without labeled data. In *International Conference on Machine Learning*, pages 273–282. PMLR.

Ashutosh Baheti, Alan Ritter, Jiwei Li, and Bill Dolan. 2018. Generating more interesting responses in neural conversation models with distributional constraints. *arXiv preprint arXiv:1809.01215*.

Ernie Chang, David Ifeoluwa Adelani, Xiaoyu Shen, and Vera Demberg. 2020. Unsupervised pidgin text generation by pivoting english data and self-training. *arXiv preprint arXiv:2003.08272*.

Ernie Chang, Vera Demberg, and Alex Marin. 2021a. Jointly improving language understanding and generation with quality-weighted weak supervision of automatic labeling. *arXiv preprint arXiv:2102.03551*.

Ernie Chang, Xiaoyu Shen, Dawei Zhu, Vera Demberg, and Hui Su. 2021b. Neural data-to-text generation with lm-based text augmentation. *arXiv preprint arXiv:2102.03556*.

Lu Chen, Xiang Zhou, Cheng Chang, Runzhe Yang, and Kai Yu. 2017. Agent-aware dropout dqn for safe and efficient on-line dialogue policy learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2454–2464.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2019. Few-shot nlg with pre-trained language model. *arXiv preprint arXiv:1904.09521*.

Richard Csaky and Gabor Recski. 2020. The gutenberg dialogue dataset. *arXiv preprint arXiv:2004.12752*.

Mostafa Dehghani, Arash Mehrjou, Stephan Gouws, Jaap Kamps, and Bernhard Schölkopf. 2017. Fidelity-weighted learning. *arXiv preprint arXiv:1711.02799*.

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

Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2016. Towards end-to-end reinforcement learning of dialogue agents for information access. *arXiv preprint arXiv:1609.00777*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*.

Hao Fang, Hao Cheng, Maarten Sap, Elizabeth Clark, Ari Holtzman, Yejin Choi, Noah A. Smith, and Mari Ostendorf. 2018. Sounding board: A user-centric and content-driven social chatbot. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 96–100, New Orleans, Louisiana. Association for Computational Linguistics.

Mehdi Fatemi, Layla El Asri, Hannes Schulz, Jing He, and Kaheer Suleman. 2016. Policy networks with two-stage training for dialogue systems. *arXiv preprint arXiv:1606.03152*.

Shaoxiong Feng, Xuancheng Ren, Kan Li, and Xu Sun. 2021. Multi-view feature representation for dialogue generation with bidirectional distillation. *arXiv preprint arXiv:2102.10780*.

Benoît Frénay and Michel Verleysen. 2013. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869.

Gabriel Gordon-Hall, Philip John Gorinski, and Shay B Cohen. 2020a. Learning dialog policies from weak demonstrations. *arXiv preprint arXiv:2004.11054*.

Gabriel Gordon-Hall, Philip John Gorinski, Gerasimos Lampouras, and Ignacio Iacobacci. 2020b. Show us the way: Learning to manage dialog from demonstrations. *arXiv preprint arXiv:2004.08114*.

Xiaodong Gu, Kang Min Yoo, and Jung-Woo Ha. 2020. Dialogbert: Discourse-aware response generation via learning to recover and rank utterances. *arXiv preprint arXiv:2012.01775*.

Hamza Harkous, Isabel Groves, and Amir Saffari. 2020. Have your text and use it too! end-to-end neural data-to-text generation with semantic fidelity. *arXiv preprint arXiv:2004.06577*.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling strategy and generation in negotiation dialogues. *arXiv preprint arXiv:1808.09637*.

Satwik Kottur, Xiaoyu Wang, and Vítor Carvalho. 2017. Exploring personalized neural conversational models. In *IJCAI*, pages 3728–3734.

Ben Krause, Marco Damonte, Mihai Dobre, Daniel Duma, Joachim Fainberg, Federico Fancellu, Emmanuel Kahembwe, Jianpeng Cheng, and Bonnie

Webber. 2017. Edina: Building an open domain socialbot with self-dialogues. *arXiv preprint arXiv:1709.09816*.

Sungjin Lee. 2017. Toward continual learning for conversational agents. *arXiv preprint arXiv:1712.09943*.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017a. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017b. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of The 8th International Joint Conference on Natural Language Processing (IJCNLP 2017)*.

Ziming Li, Sungjin Lee, Baolin Peng, Jinchao Li, Shahin Shayandeh, and Jianfeng Gao. 2020. Guided dialog policy learning without adversarial learning in the loop. *arXiv preprint arXiv:2004.03267*.

Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*.

Ryan Lowe, Michael Noseworthy, Iulian V Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. *arXiv preprint arXiv:1708.07149*.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul Crook, Bing Liu, Zhou Yu, Eunjoon Cho, and Zhiguang Wang. 2020. Continual learning in task-oriented dialogue systems. *arXiv preprint arXiv:2012.15504*.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Vancouver, Canada. Association for Computational Linguistics.

Tong Niu, Semih Yavuz, Yingbo Zhou, Huan Wang, Nitish Shirish Keskar, and Caiming Xiong. 2020. Unsupervised paraphrase generation via dynamic blocking. *arXiv preprint arXiv:2010.12885*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. *arXiv preprint arXiv:1704.03084*.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot natural language generation for task-oriented dialog. *arXiv preprint arXiv:2002.12328*.

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. 2018. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604*.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2020. Snorkel: Rapid training data creation with weak supervision. *The VLDB Journal*, 29(2):709–730.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3295–3301. AAAI Press.

Pararth Shah, Dilek Hakkani-Tur, Bing Liu, and Gokhan Tur. 2018. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 41–51.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*.

Xiaoyu Shen, Hui Su, Wenjie Li, and Dietrich Klakow. 2018. Nexus network: Connecting the preceding and the following in dialogue generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4316–4327.

Shang-Yu Su, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Yun-Nung Chen. 2018. Discriminative deep dyna-q: Robust planning for dialogue policy learning. *arXiv preprint arXiv:1808.09442*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.

Ryuichi Takanobu, Runze Liang, and Minlie Huang. 2020. Multi-agent task-oriented dialog policy learning with role-aware reward decomposition. *arXiv preprint arXiv:2004.03809*.

Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 2019. Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog. *arXiv preprint arXiv:1908.10719*.

Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2018. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Jiawei Wu, Xin Wang, and William Yang Wang. 2019. Self-supervised dialogue learning. *arXiv preprint arXiv:1907.00448*.

Yu Wu, Wei Wu, Zhoujun Li, and Ming Zhou. 2018. Learning matching models with weak supervision for response selection in retrieval-based chatbots. *arXiv preprint arXiv:1805.02333*.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Chen Xing, Yu Wu, Wei Wu, Yalou Huang, and Ming Zhou. 2018. Hierarchical recurrent attention network for response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Yujie Xing and Raquel Fernández. 2018. Automatic evaluation of neural personality-based chatbots. *arXiv preprint arXiv:1810.00472*.

Can Xu, Wei Wu, and Yu Wu. 2018a. Towards explainable and controllable open domain dialogue generation with dialogue acts. *arXiv preprint arXiv:1807.07255*.

Ruijian Xu, Chongyang Tao, Daxin Jiang, Xueliang Zhao, Dongyan Zhao, and Rui Yan. 2020. Learning an effective context-response matching model with self-supervised tasks for retrieval-based dialogues.

Xinnuo Xu, Ondřej Dušek, Ioannis Konstas, and Verena Rieser. 2018b. Better conversations by modeling, filtering, and optimizing for coherence and diversity. *arXiv preprint arXiv:1809.06873*.

Denis Yarats and Mike Lewis. 2018. Hierarchical text generation and planning for strategic dialogue. In *International Conference on Machine Learning*, pages 5591–5599. PMLR.

Hainan Zhang, Yanyan Lan, Liang Pang, Jiafeng Guo, and Xueqi Cheng. 2019. Recosa: Detecting the relevant contexts with self-attention for multi-turn dialogue generation. *arXiv preprint arXiv:1907.05339*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.

Tiancheng Zhao and Maxine Eskenazi. 2016. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. *arXiv preprint arXiv:1606.02560*.

Yufan Zhao, Can Xu, and Wei Wu. 2020. Learning a simple and effective model for multi-turn response generation with auxiliary tasks. *arXiv preprint arXiv:2004.01972*.

Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. An affect-rich neural conversational model with biased attention and weighted cross-entropy loss. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7492–7500.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

A Implementation Details

Our implementation uses the open source Huggingface Transformer repository (Wolf et al., 2020). Specifically, we have used the base version from sentence transformers pre-trained on millions of paraphrase examples, named as '*paraphrase-distilroberta-base-v1*'. The encoder-decoder framework is initialized with the base version '*bert-base-uncased*'but with configuration of smaller size. The smaller sized model reduces the '*bert-base-uncased*'configuration to 6 transformer layers, has a hidden size of 768, and contains 2 attention heads, {L=6, H=768, A=2}. Similar to Gu et al. (2020) we sum the position embeddings to the output sentence embeddings of size 768 to indicate the user or agent utterances. Odd ones indicate the user utterances and even ones are that of an agent. The MLP network for semantic relevance and semantic coherence used a hidden dimension of 128. The Δ has been set to best value of 0.54 after performing a grid search in the range of $\{0.4, 0.7\}$ with step size of 0.02. The reward estimator models g_ω using two hidden layers of size 512 and 256 respectively. And, h is modelled using a single hidden layer of size one. In each training iteration the policy and reward estimator are updated with continual learning to avoid catastrophic forgetting mechanism using EWC modified loss, the λ value used as a parameter is set to 0.4. Also, at each training iteration the policy and reward parameters are saved if it reduces the perplexity on the validation set (calculated after running for all the batches of the training dataset) and patience is set to 3 as a stopping criterion before we terminate the training.