Towards Automatic Evaluation of Dialog Systems: A Model-Free Off-Policy Evaluation Approach

Haoming Jiang*[1,2], Bo Dai[3], Mengjiao Yang[3], Tuo Zhao[2], Wei Wei[4]
[1]Amazon.com Inc, CA, USA [2]Georgia Institute of Technology, GA, USA
[3]Google Brain, CA, USA [4]Google Cloud AI, CA, USA
{jianghm,tourzhao}@gatech.edu
{bodai,sherryy,wewei}@google.com

Abstract

Reliable automatic evaluation of dialogue systems under an *interactive* environment has long been overdue. An ideal environment for evaluating dialog systems, also known as the Turing test, needs to involve human interaction, which is usually not affordable for large scale experiments. Though researchers have attempted to use metrics for language generation tasks (e.g., perplexity, BLEU) or some *model-based* reinforcement learning methods (e.g., self-play evaluation) for automatic evaluation, these methods only show very weak correlation with the actual human evaluation in practice. To bridge such a gap, we propose a new framework named ENIGMA for estimating human evaluation scores based on recent advances of off-policy evaluation in reinforcement learning. ENIGMA only requires a handful of pre-collected experience data, and therefore does not involve human interaction with the target policy during the evaluation, making automatic evaluations feasible. More importantly, ENIGMA is *model-free* and *agnostic to the behavior policies* for collecting the experience data (see details in Section 2), which significantly alleviates the technical difficulties of modeling complex dialogue environments and human behaviors. Our experiments show that ENIGMA significantly outperforms existing methods in terms of correlation with human evaluation scores.

1 Introduction

One of the fundamental research bottlenecks for developing dialog systems falls in evaluation, namely how to measure the performance of these systems in an automatic and scalable manner. Different from supervised natural language understanding tasks (e.g., text classification and machine translation), an ideal environment for evaluating dialog systems, also known as the Turing test, involves multi-turn human interaction (Turing, 1950; Liu et al., 2016; Ghandeharioun et al., 2019; See et al., 2019). While online platforms such as Amazon Mechanical Turk can provide human-based evaluation, they are often expensive and not scalable.

Researchers have adopted language quality metrics for single-turn response generation given a fixed context (e.g., BLEU score and perplexity) to implement automatic dialog systems evaluation (DeVault et al., 2011; Xiang et al., 2014; Higashinaka et al., 2014; Gandhe and Traum, 2016; Lowe et al., 2017). However, these metrics only weakly correlate to human evaluation in practice (Liu et al., 2016; Ghandeharioun et al., 2019). One cause of such weak correlation is that language quality metrics rely on the exact match between generated text and ground-truth, which generally do not fully overlap. While certain embedding-based metrics have been developed to combat this lack of coverage (Mitchell and Lapata, 2008; Dziri et al., 2019), they are only post-hoc judgments based on static experience data, and does not necessarily reflect the dynamic quality of multi-turn interactive dialog well (Ghandeharioun et al., 2019). Moreover, evaluation of goal-oriented dialog systems should be based on how well dialog systems collect information from users and whether the goal is completed; language quality metrics are thus unable to meet these requirements.

To overcome the limitations of the aforementioned static evaluation methods, another line of work has proposed to model the interactive process of a conversation as a Markov decision process (MDP) (Möller et al., 2006; Li et al., 2016; Yu et al., 2016; Shah et al., 2018; Jaques et al., 2019). Accordingly, automatic evaluation of dialog systems can be formulated as an off-policy evaluation (OPE) problem, where a human subject is the so-called “environment" in the reinforcement learning (RL) literature. For instance, Wei et al. (2018) propose a model-based approach for goal-oriented dialog systems. They first learn an envi-

[*] Work was done during internship at Google Cloud AI.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7419–7451
November 7–11, 2021. ©2021 Association for Computational Linguistics

ronment/human model from the experience data consisting of human response, and then evaluate a dialog agent/policy by executing the policy within the learned environment. This procedure is known as "self-play evaluation". Such a model-based approach requires an accurate estimation of an environment/human. However, both the input and output of the environment are in a *combinatorially* large space, i.e., the trained model needs to be able to mimic complex human behavior of generating meaningful sentences from huge vocabulary. Unfortunately, such a requirement is far beyond the current capability of model-based RL algorithms. As a result, evaluations that require accurate modeling of the environment are often unreliable. A similar model-based approach is proposed (Ghandeharioun et al., 2019) to evaluate open-domain chit-chat dialog systems. In addition to modeling human behavior, they also model the reward function (for mimicking the complex mechanism behind human ratings) based on handcrafted features, which makes evaluation even more unreliable.

In this paper, we propose a general OPE framework named ENIGMA (EvaluatiNg dIaloG systeMs Automatically) for estimating human evaluation score (i.e., how a human would rate a dialog system). Different from the existing model-based approaches, which rely on complex modeling of human behavior given combinatorially large vocabulary, ENIGMA takes advantage of recent advances in model-free OPE and avoids direct modeling of dynamic transitions and reward functions in a complex environment. Moreover, ENIGMA overcomes several limitations of existing OPE methods in order to evaluate dialog systems: **(I)** Existing OPE methods only apply to infinite or fixed horizon settings (where horizon length corresponds to number of turns in a conversation), while conversations, on the other hand, often have varying horizon lengths; **(II)** Existing OPE methods require experience data to sufficiently cover states and actions a target policy might visit. Due to limited experience data and the combinatorial nature of languages, such a requirement can hardly be satisfied in dialog evaluation; **(III)** Certain OPE methods rely on accurate estimation of the behavior policies used to collect the experience data. Unfortunately, such behavior policies are humans or complex dialog systems, and estimating their probabilistic model is a challenging imitation learning problem. [1]

To address **(I)**, we propose a pseudo state padding method, which augments each conversation into infinitely many turns while preserving the original policy value; to address **(II)**, we leverage pre-trained language models (Devlin et al., 2018), which essentially transfer knowledge from open-domain data to learn a representation for alleviating the coverage requirement in original combinatorial space; to address **(III)**, we adopt a stationary distribution correction estimation approach (Nachum et al., 2019a), which directly models the state-action density ratio between the experience data and the target policy (Liu et al., 2018), and is therefore agnostic to the behavior policy.

We conduct thorough experiments on evaluating goal-oriented (AirDialog, Wei et al. (2018)) and chit-chat (ConvAI2, Dinan et al. (2020)) dialog systems to demonstrate the superiority of ENIGMA. Specifically, we follow the experimental settings similar to Ghandeharioun et al. (2019); See et al. (2019) (See details in Section 4), and show ENIGMA significantly outperforms the existing static and self-play evaluation methods.

2 Background

• Dialog Generation as Markov Decision Process. A conversation is generated through interactions alternating between an agent π (i.e., a dialog system) and an environment $\mathcal{E}$ (i.e., a human). We denote the conversation as $h = \{e_0, a_1, e_1, ..., a_T\}$, where a_i and e_i are sentences generated by π and $\mathcal{E}$ respectively, and T is the number of turns in the conversation. Dialog can be naturally described as a MDP (Puterman, 1995), $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, \mu_0 \rangle$. Specifically, at the t-th turn, state $s_t \in \mathcal{S}$ captures the previous conversation history $s_t = \{e_0, a_1, e_1, ..., a_{t-1}, e_{t-1}\}$. An action $a_t \in \mathcal{A}$ is an agent's response given this context. Conversation can then be represented by the last state and action, i.e., $h = \{s_T, a_T\}$. An agent π is essentially a policy that maps $\mathcal{S}$ to $\mathcal{P}(\mathcal{A})$, where $\mathcal{P}(\cdot)$ denotes the set of probability measures over the action space. A transition kernel $P(\cdot|s_t, a_t)$ returns s_{t+1} as the state at turn $t+1$, and an environment $\mathcal{E}$ generates a reward $r_t = R(s_t, a_t) \in [0, 1]$. Note that s_{t+1} essentially concatenates s_t and a_t with e_t. The initial state $s_1 = \{e_0\}$ is randomly sampled from some distribution μ_0. We follow the *sparse reward* setting,

[1] Note that even though some of the model-free OPE estimators still require modeling behavior policies, they are still significantly easier than model-based OPE, which has to model the underlying dialog environment.

where each conversation is only evaluated at the ending state, i.e., $r_t = 0$ for $t < T$.

- **Automatic Dialog Evaluation as Off-Policy Evaluation**. Dialog evaluation can be naturally viewed as computing the expected reward of the above MDP defined as

$$\rho(\pi) = \mathbb{E}_{h \sim \mu_0, \pi, \mathcal{E}}[R(s_T, a_T)], \qquad (1)$$

where $h = \{s_T, a_T\}$ is sampled from the initial distribution μ_0 and the interaction between π and $\mathcal{E}$. When the environment (i.e., human) is accessible, $\rho(\pi)$ can be directly estimated by interaction with the environment, which is known as *on-policy evaluation* (Sutton and Barto, 2018).

When interaction with human is prohibited, human-free automatic evaluation is required and *Off-policy evaluation* (OPE) (Precup, 2000) is an appealing choice. In particular, OPE can estimate $\rho(\pi)$ based solely on pre-collected tuples $\{(s, a, r, s')_i\}_{i=1}^N$ from (multiple) behavior policies that are different from π.

OPE has been considered as one of the most fundamental problems in RL. A straightforward approach is to first directly learn an environment model (R and P) from experience data and then estimate $\rho(\pi)$ by executing the policy within the learned environment. Such *model-based* OPE exactly corresponds to the so-called "self-play evaluation" in the dialog system literature (Wei et al., 2018; Ghandeharioun et al., 2019). Unfortunately, it is notoriously difficult to specify a proper model for highly complicated environments such as a dialog environment (i.e., a human), where the state and action spaces are combinatorially large due to huge vocabulary size and complex transitions. As a result, the estimation error of the environment accumulates as interaction proceeds, and model-based self-play evaluation of dialog systems often becomes unreliable (Voloshin et al., 2019).

To address the challenge above, many *model-free* OPE methods that avoid direct modeling of the environment have been proposed. Model-free OPE can be categorized into *behavior-aware* and *behavior-agnostic* methods. Specifically, behavior-aware methods rely on either knowing or accurately estimating the probabilistic model of the behavior policies used for collecting the experience data (e.g., inverse propensity scoring, Horvitz and Thompson (1952)). Unfortunately, behavior policies are often unknown in practice. Estimating their probabilistic models is also quite challenging, as it requires modeling human behaviors or complex dialog systems. Behavior-agnostic methods, on the other hand, do not require explicit knowledge or direct modeling of behavior policies, and are therefore more favorable when experience data is collected by multiple (potentially unknown) behavior policies.

Unfortunately, most of the existing model-free behavior-agnostic OPE methods focus on either infinite-horizon (Nachum et al., 2019a; Zhang et al., 2020b; Yang et al., 2020) or fixed-horizon settings (Yin and Wang, 2020; Duan and Wang, 2020), and cannot be applied to evaluating dialog systems whose horizon (number of turns) vary between conversations. While LSTDQ (Lagoudakis and Parr, 2003) can be adopted to handle varying horizons, it has been shown to not work well under the sparse reward setting (Lagoudakis and Parr, 2003; Mataric, 1994).

3 ENIGMA

We present the ENIGMA framework for automatically evaluating dialog systems. In particular, ENIGMA is model-free and agnostic to behavior policies for generating the experience data. ENIGMA has three components: **(1)** pseudo-state padding for converting a dialog into an infinite-horizon MDP, **(2)** distribution-correction estimation (DICE, Nachum et al. (2019a)) with post-normalization for estimating the value of the target policy based on experience data, and **(3)** function approximation with pre-trained language models.

3.1 Pseudo-State Padding

As mentioned in Section 2, existing model-free behavior-agnostic OPE methods cannot handle varying horizon lengths in conversations under the sparse reward setting. To address this issue, we design a special padding scheme, so that the policy value can be estimated by OPE methods from the resulting padded MDP. We first pad conversation sequences with pseudo states, which leads to a padded MDP with a fixed horizon length $T_{\max}$. We then convert such a fixed horizon MDP into infinite horizon by augmentation, i.e., we repeatedly concatenate the ending state of the fixed horizon MDP to its initial state. More specifically, the policy takes a deterministic action at all pseudo states, i.e., $\pi(a = \mathrm{NextPad} | s = \mathrm{Pad}_k) = 1$. The transition kernel of the new process can be defined as

Conversation Transition :
$P(s' = s \cup a \cup e | s, a, \text{incomplete conv.}) = \mathcal{E}(e|s, a),$
Padding with Pseudo States :
$P(s'{=}\text{Pad}_{T+1}|s, a, \text{complete conv. with } T \text{ turns}){=}1,$
$P(s'{=}\text{Pad}_{k+1}|s = \text{Pad}_k, a = \text{NextPad}, k < T_{\max}){=}1,$
Concatenate Conversations :
$P(s'|s = \text{Pad}_{T_{\max}}, a = \text{NextPad}) = \mu_0\left(s'\right).$

This new process is still a valid MDP, as its transition kernel satisfies the Markov property. For notational simplicity, we refer to this new process as "the augmented MDP".

Accordingly, the policy value of π for the augmented MDP can be defined as

$$\rho_A(\pi){=}\lim_{N\to\infty} \mathbb{E}_{\{h_i\}_{i=1}^N \sim \mu_0,\pi,\mathcal{E}}[\sum_{i=1}^{N} \sum_{t=1}^{T_{\max}} \frac{R(s_t^{(i)},a_t^{(i)})}{NT_{\max}}], \quad (2)$$

where h_i's are padded conversations sampled from interactions between π and $\mathcal{E}$. Since there is only one non-zero reward for every $T_{\max}$ steps, rewards in the augmented MDP are also sparse.

We remark that the augmented MDP has a unique stationary distribution $d^\pi(s, a)$. For the station-action pair (s_t, a_t) in a conversation h with padded pseudo states, we have

$$d^\pi(s_t, a_t) = \tfrac{1}{T_{\max}} \textstyle\sum_{\{(s_k,a_k)\}_{k=1}^{t-1}} [\mu_0(s_1)\pi(a_1|s_1) \\ P(s_2|a_1, s_1) \cdots P(s_t|a_{t-1}, s_{t-1})\pi(a_t|s_t)], \quad (3)$$

where $\{(s_k, a_k)\}_{k=1}^{t-1}$ are the state-action pairs in the same conversation as (s_t, a_t).

Moreover, the policy value of π under the augmented MDP is proportional to its counterpart under the original MDP without augmentation:

$$\rho_A(\pi) = \mathbb{E}_{(s,a)\sim d^\pi(s,a)}[R(s, a)] = \frac{\rho(\pi)}{T_{\max}}. \quad (4)$$

Due to space limit, we defer the details and proof to Appendix A.1.

Remark 1. Some OPE methods, e.g., LSTDQ (Lagoudakis and Parr, 2003), can handle fixed horizons, therefore only applying the fixed-horizon padding would suffice. DICE estimators (Nachum et al., 2019a), on the other hand, can only handle infinite horizons, therefore the infinite-horizon augmentation is necessary.

Remark 2. In practice, we do not actually need to concatenate infinitely many conversations for computing $\rho_A(\pi)$. As suggested by (4), $\rho_A(\pi)$ can be computed based on $d^\pi(s_t, a_t)$ defined in (3), which is the product of only finite terms.

3.2 Model-Free Behavior-Agnostic DICE Estimator

With the proposed augmentation, we obtain an infinite horizon MDP from which the policy value of the original MDP can be recovered. We then apply DICE (Nachum et al., 2019a; Yang et al., 2020) to estimate $\rho_A(\pi)$ based on pre-collected experience data $\mathcal{D} = \{(s, a, r, s')_i\}_{i=1}^N$ without interacting with $\mathcal{E}$ (i.e., a human), where $(s, a) \sim d^{\mathcal{D}}$ are samples from some unknown distribution $d^{\mathcal{D}}$. We slightly abuse the notations and use $(s, a, r, s') \sim d^{\mathcal{D}}$ as a shorthand for $(s, a) \sim d^{\mathcal{D}}, r = R(s, a), s' \sim P(\cdot|s, a)$, which simulates sampling form the dataset $\mathcal{D}$.

DICE is a model-free policy evaluation method (without explicitly modeling $\mathcal{E}$) and does not require knowledge of behavior policies for generating the experience data, which provides a more reliable estimation of $\rho_A(\pi)$ than other OPE methods. Specifically, DICE decomposes $\rho_A(\pi)$ into:

$$\rho_A(\pi) = \mathbb{E}_{(s,a,r)\sim d^{\mathcal{D}}}[\zeta(s, a)r], \quad (5)$$

where $\zeta(s, a) := d^\pi(s, a)/d^{\mathcal{D}}(s, a)$ is the *distribution correction ratio*. Then DICE estimates ζ by solving the following regularized minimax optimization problem:

$$\max_{\zeta\geq 0} \min_{\nu,\lambda} L_D(\zeta, \nu, \lambda) = \mathbb{E}_{(s,a,r,s')\sim d^{\mathcal{D}}, a'\sim\pi(s')}[\zeta(s, a) \\ \cdot (\nu(s', a') - \nu(s, a))] + \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}[\lambda(\zeta(s, a) \\ - 1)] - \alpha_\zeta \cdot \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}[f(\zeta(s, a))]. \quad (6)$$

where $\nu(s, a)$'s are auxiliary variables, f is a convex regularizer (e.g., $f(x) = x^2$), and α_ζ is a tuning parameter. Due to the space limit, we omit the details of deriving the DICE estimator. Please refer to Yang et al. (2020) for more details.

- **Post-Normalization**. Note that (6) handles the constraint $\mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}\zeta(s, a) = 1$ by Lagrange multipliers λ, which cannot guarantee that the constraint is *exactly* satisfied when solving (6) using alternating SGD-type algorithms (Dai et al., 2017; Chen et al., 2018). To address this issue, we propose a post-normalization step that explicitly enforces the constraint:

$$\rho_n(\pi) = \sum_{(s,a,r)\sim d^{\mathcal{D}}} \zeta(s, a)r \Big/ \sum_{(s,a)\sim d^{\mathcal{D}}} \zeta(s, a). \quad (7)$$

As we will see in our experiments in Section 4, the post-normalization step is crucial for DICE to attain good estimation accuracy in practice; without post-normalization, we observe potential divergence in terms of policy value estimation.

• **Why do we prefer DICE?** Deep Q-learning and its variants are another popular model-free and behavior-agnostic approach to off-policy evaluation. However, due to the sparse rewards in dialogs, fitting the state-action value function (i.e., the Q-function) in deep Q-learning is notoriously difficult (Mataric, 1994). We observe in Section 4 that deep Q-learning is computationally unstable.

In contrast, DICE only needs to estimate the density correction ratio ζ, which is decoupled from the rewards associated with the policy value as shown from (6). This significantly alleviates the computational challenge incurred by sparse rewards. Moreover, DICE also applies the post-normalization, additional regularization (i.e., $\mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}[f(\zeta(s,a))]$), and constraints on ζ (i.e., $\zeta \geq 0$ and $\mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}[\zeta(s,a)] = 1$), all of which further stabilize training. These features allow DICE achieve better estimation performance than deep Q-learning in dialog systems evaluation.

Recent progresses in OPE based on density ratio estimation are remarkable (Liu et al., 2018; Nachum et al., 2019a; Xie et al., 2019; Uehara et al., 2019), however, there exists a statistical limit in off-policy evaluation. Specifically, the Cramer-Rao lower bound of the MSE has been established in Jiang and Li (2016), which is proportional to the square of the density ratio. This implies that we can only obtain accurate estimation of policy value only if the ratio ζ is not too large. While the ratio-based minimax algorithms should have achieved the lower bound (Kallus and Uehara, 2019; Yin and Wang, 2020; Ren et al., 2021), even better estimation results can be obtained when behavior and target policies are more similar. We thus introduce an experience data collection protocol in Section 4.1 which satisfies the bounded ratio requirement and ensures the success of OPE methods.

3.3 Function Approximation with RoBERTa

Despite the apparent advantages of DICE estimators, directly training DICE from scratch will fall short due to the bounded ratio requirement being quickly broken in the large combinatorial state-action space in dialog.

We alleviate this issue by using reliable representations of pre-trained language models (Devlin et al., 2018). By virtue of the huge amounts of pre-training data and the massive model size, the pre-trained models can effectively capture rich semantic and syntactic information of natural language (rather than enumerating the original combinatorial language space).

In particular, we transfer the knowledge from RoBERTa (Liu et al., 2019) to dialog evaluation, and parameterize ζ and ν as follows: we keep the pre-trained RoBERTa encoder layer and replace the original mask language modeling head by a two-layer fully connected network with a scalar output. For simplicity, we denote the corresponding parametric forms of ζ and ν in (6) as RoBERTa-ζ and RoBERTa-ν, respectively. Note that we only need RoBERTa-ζ and RoBERTa-ν to share the same encoder. We then use RoBERTa-ζ and RoBERTa-ν as the initial solution to solve (6), which is also known as fine-tuning (Devlin et al., 2018).

With a properly designed mask, the self-attention mechanism in the bi-direction transformer architecture allows us to efficiently compute $\zeta(s,a)$ and $\nu(s,a)$ for all state-action pairs in the same dialog simultaneously. Due to the space limit, we defer the mask design details to Appendix A.2.

Algorithm 1 ENIGMA

```
Input: Experience conversations $\mathcal{D} = \{(h_i = \{e_0^{(i)}, a_1^{(i)}, e_1^{(i)}, ..., a_{T_i}^{(i)}\}, r^{(i)})\}_{i=1}^{N}$, Target Policy $\pi$, Padding Length $T_{\max}$, Regularization function $f$, DICE hyper-parameters $\alpha_\zeta$, $\alpha_R$
Output: Performance Estimation $\widehat{\rho}_n(\pi)$
Parameters: $\zeta = \{\text{RoBERTa-}\zeta, [\zeta_{\text{pad},t}]_{t=1}^{T_{\max}}\}$, $\nu = \{\text{RoBERTa-}\nu, [\nu_{\text{pad},t}]]_{t=1}^{T_{\max}}\}$, $\lambda$
while Sample $(h, r)$ in $\mathcal{D}$ do
    $\zeta_0 = \zeta_{\text{pad},T_{\max}}, \;\; \nu_0 = \nu'_0 = \nu_{\text{pad},T_{\max}}$
    for $t$ in $1, \cdots, T$ do
        $\zeta_t = \text{RoBERTa-}\zeta(e_0, a_1, ..., e_{t-1}, a_t)$
        $\nu_t = \text{RoBERTa-}\nu(e_0, a_1, ..., e_{t-1}, a_t)$
        $\widetilde{a}_t \sim \pi(e_0, a_1, ..., e_{t-1})$
        $\nu'_t = \text{RoBERTa-}\nu(e_0, a_1, ..., e_{t-1}, \widetilde{a}_t)$
    end for
    for $t$ in $T+1, \cdots, T_{\max}$ do
        $\zeta_t = \zeta_{\text{pad},t}, \;\; \nu_t = \nu'_t = \nu_{\text{pad},t}$
    end for
    $L_D(\zeta, \nu, \lambda) = \frac{1}{T_{\max}}[\sum_{t=0}^{T_{\max}-1}[\zeta_t(\nu'_{t+1} - \nu_t) + \lambda(\zeta_t - 1) - \alpha_\zeta f(\zeta_t)]]$
    SGD update based on $\frac{\partial L_D}{\partial \nu}, \frac{\partial L_D}{\partial \lambda}, \frac{\partial -L_D}{\partial \zeta}$.
end while
for $(h_i, r^{(i)})$ in $\mathcal{D}$ do
    $\zeta_i = \text{RoBERTa-}\zeta(h_i)$
end for
Return $\widehat{\rho}_n(\pi) = \sum_i \zeta_i r^{(i)} / \sum_i \zeta_i$
```

3.4 Summary

We summarize ENIGMA in Algorithm 1. Due to the space limit, we only present ENIGMA using SGD with batch-size 1 here. We defer the details of ENIGMA with mini-batch SGD to Appendix A.3 (Algorithm 2).

4 Experiments

We empirically evaluate ENIGMA on two dialog datasets: AirDialog (Wei et al., 2018) for goal-oriented tasks and ConvAI2 (Dinan et al., 2020) for open-domain chit-chat respectively. See details of experimental setup in Appendix B. [2]

4.1 Policy Training Data and Experience Data

As mentioned in Section 3.2, there exists an information theoretic limit for all off-policy evaluation methods: no method can perform well when the state-action density ratio between the target and behavior policy is too large. To avoid such a circumstance, we need to ensure that the experience data collected by a behavior policy do not deviate too much from data induced by the target policy. Unfortunately, both datasets used in our experiments do not satisfy such a requirement. AirDialog, for example, consists of dialog between humans, which are near-perfect golden samples as human agents almost always successfully book tickets for customers. Dialog system agents, on the other hand, have many failure modes (i.e., the target policy/agent does not book the correct ticket for a human customer). Hence, directly using human dialog as the behavior data to evaluate dialog agents is subject to limitations.

In order to properly evaluate an imperfect target policy in the presence of the information theoretic limit, we refer to Lowe et al. (2017); Ghandeharioun et al. (2019), and collect experience data using behavior policies similar to the target policy. To avoid confusion, we call data collected by the behavior policy "experience data" and data used to train an agent "policy training data". More details are elaborated below for each dataset.

It is worth noting that existing work on dialog systems evaluation also enforces similar requirements. For example, Lowe et al. (2017) show higher Pearson correlation coefficient (0.37) between automatic metrics and human ratings when behavior policies contain the target policy. When the target policy is excluded from behavior policies, however, the correlation is only 0.13, even lower than the meaningless correlation between dialog lengths and human ratings (0.27). Another example is Ghandeharioun et al. (2019), where the studied agents are similar to each other in their hierarchical architectures, hyperparameters, and training data.

4.2 Goal-Oriented Systems

We first test ENIGMA for evaluating goal-oriented dialog systems on a flight ticket booking task.

• **Policy Training Data**. We use the ***AirDialog*** dataset[3] for policy training (Wei et al., 2018). It contains 402,038 pieces of dialog from human sellers and human customers collaborating on buying flight tickets. We use different proportions of the dataset and different hyperparameters to train 24 seller agents using behavioral cloning (See Appendix C for details) [4].

• **Experience Data**. We invite 20 people to evaluate the 24 seller agents. Specifically, each of the 20 human customers interacts with a seller agent 5 times to generate 100 pieces of dialog, and gives each piece an evaluation score between 0 and 1. The final score an agent receives is the average of the 100 scores. We consider three types of scores: flight score, status score, and overall reward used in Wei et al. (2018).

We evaluate ENIGMA, BLEU/PPL (Papineni et al., 2002) and Self-Play Evaluation (SPE) based on the correlation between estimated reward and true reward. The results are summarized in Table 1. ENIGMA uses the experience data of the other 23 agents to evaluate each agent (i.e., leave-one-bot-out). Note that SPE (Wei et al., 2018) needs to train a customer agent in addition to the seller agent being evaluated. For a fair comparison, we train the SPE customer agent on both experience data and policy training data (See Appendix C for details). Our empirical observations are as follows:

• **ENIGMA vs. BLEU/PPL**. ENIGMA significantly outperforms BLEU/PPL. As mentioned earlier, BLEU and PPL are well-known metrics for evaluating language quality. For goal-oriented systems whose goal is to complete a specific task, however, BLEU and PPL scores show little correlation

[2] We release our source code for ENIGMA algorithm here: https://github.com/google-research/google-research/tree/master/dialogue_ope/airdialogue_ope and dataset here https://github.com/HMJiangGatech/dialogue_ope_data.

[3] https://github.com/google/airdialogue

[4] We also demonstrate that ENIGMA can be applied to rule based agent in Appendix D.2.

Setting	Method	Pearson Correlation				Spearman's Rank Correlation			
		Flight Score	Status Score	Reward	Average	Flight Score	Status Score	Reward	Average
All Agents	BLEU	0.1450	-0.1907	-0.0709	-0.0389	0.0370	-0.1453	-0.1472	-0.0852
	PPL	-0.1598	0.1325	0.0195	-0.0026	-0.1817	0.0090	-0.0039	-0.0649
	SPE	0.6450	0.7926	0.7482	0.7286	0.3539	0.8004	0.7400	0.6314
	ENIGMA	**0.9255**	**0.9854**	**0.9672**	**0.9593**	**0.8948**	**0.9839**	**0.9435**	**0.9407**
Selected Agents	BLEU	-0.0621	-0.1442	0.2944	0.0294	-0.1273	-0.2208	0.1793	-0.1758
	PPL	-0.0197	-0.1775	0.0460	-0.0504	-0.1146	-0.4652	-0.0404	-0.2067
	SPE	0.0970	0.5203	0.4777	0.3650	0.1368	0.5304	0.4943	0.3872
	ENIGMA	**0.8640**	**0.9031**	**0.8952**	**0.8874**	**0.8496**	**0.9414**	**0.8782**	**0.8686**

Table 1: The correlation between two metrics. Each column is a task completion score obtained by interacting human customers ("Selected Agents" denotes only evaluating agents with reasonably good performance).

with task completion scores.

• **ENIGMA vs. SPE**. ENIGMA significantly outperforms SPE. To better understand their performance, we also present the regression plots between estimated and true rewards in Figure 1. Both ENIGMA and SPE can easily identify agents with extremely poor rewards. However, for certain good agents whose flight score, status score, and overall reward are better than 0.5, 0.7, and 0.65 respectively, SPE performs worse than ENIGMA by a much larger margin (especially for flight score). Additional regression plots are shown in Appendix D.1.

• **Ablation Study**. We select 2 out of the 24 agents to illustrate the importance of each component in ENIGMA.

⋆ ***DICE vs. LSTDQ***. Figure 2(a) and Figure 2(b) show the estimated values of LSTDQ (only fitting the Q-function) and DICE respectively: estimates of LSTDQ are stuck at 0 whereas estimates of DICE approach the true rewards (dotted lines) as training progresses. Figure 3 additionally shows that the training objectives of LSTDQ oscillates as DICE stably converges.

⋆ ***Post-normalization***. Figure 2(c) shows the performance of ENIGMA without post-normalization: The algorithm fails to estimate the true rewards.

⋆ ***Pretrained Encoder***. Figure 2(d) shows the performance of ENIGMA without the pretrained encoder: The estimated values can approach the true rewards, but are less stable and less accurate than the counterpart with the pretrained encoder.

4.3 Open-Domain Chit-chat Systems

We now test ENIGMA for evaluating open-domain chit-chat dialog systems.

• **Policy Training Data**. We use 29 pre-trained agents[5] provided by See et al. (2019). These agents are trained using behavioral cloning on the ***ConvAI2*** dataset[6] (Zhang et al., 2018; Dinan et al., 2020). The dataset contains 8,939 pieces of dialog, where participants are instructed to chat naturally using given personas.

• **Experience Data**. We use the experience dataset provided by See et al. (2019). The dataset contains 3,316 agent-human evaluation logs and 10 different language quality metrics for each log.

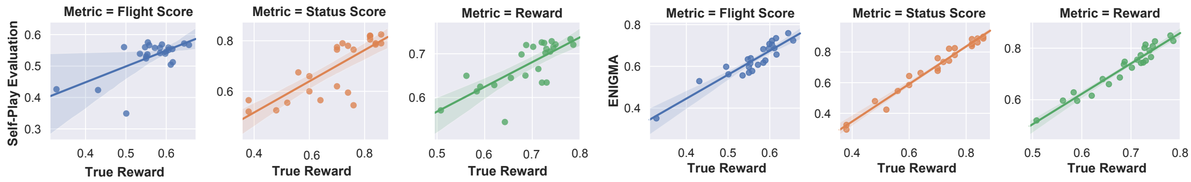
We follow the setups from Section 4.2 to evaluate ENIGMA, SPE, BLEU, BLEURT (Sellam et al., 2020), BERTscore (Zhang et al., 2019), and 8 Hand-Crafted Dialog Features (HCDFs) based on Pearson and Spearman's rank correlations between the estimated rewards and the true rewards. Here the true rewards are human evaluation scores under 10 different language quality metrics. More details of HCDFs and language quality metrics can be found in See et al. (2019). The average, minimum and maximum of the 10 correlations (under different language quality metrics) of each method are summarized in Table 2. Moreover, we consider using 8 HCDFs to fit the true rewards using linear regression, and the results are included in Table 2.

Note that since we are considering a chit-chat dialog system, SPE does not train an additional agent but asks two identical target agents to chat with each other. However, SPE needs to train an additional model to predict the reward of each dialog. Specifically, we fine-tune the pre-trained RoBERTa encoder with an output layer over the experience data (an additional sigmoid function is applied to ensure an output between 0 and 1). For automatic evaluation of each agent using ENIGMA, we use the experience data of the other 28 agents (i.e., leave-one-bot-out).

• **ENIGMA vs. Baselines**. ENIGMA significantly outperforms SPE, BLEU, BLEURT, BERTscore and HCDFs in both Pearson and Spearman's rank

[5] https://github.com/facebookresearch/ParlAI/tree/master/projects/controllable_dialogue

[6] https://parl.ai/projects/convai2/



(a) SPE vs. Human Evaluation

(b) ENIGMA vs. Human Evaluation

Figure 1: Regression Plots. The x-axis is the average reward obtained by chatting with human. The y-axis is the reward estimated by SPE / ENIGMA. Different colors denote different types of rewards (flight, status, and overall score). The solid line is obtained by linear regression and the shaded region indicates 95% confidence interval.

Method	Experience Data	Pearson Correlation			Spearman's Rank Correlation		
		Average	Min	Max	Average	Min	Max
Best of 8 HCDFs	Human-Human	0.6045	0.4468	0.9352	0.3384	0.1724	0.7526
8 HCDFs + Regression	Human-Human	0.5387	-0.0348	0.7519	0.4740	0.2784	0.7880
BLEU	Human-Human	0.4127	0.0671	0.6785	0.2965	0.0482	0.7236
BLEURT	Human-Human	0.4513	0.1557	0.6572	0.4389	0.0055	0.6864
BERTscore F-1	Human-Human	0.5365	0.0609	0.8385	0.5293	0.2852	0.7044
SPE	Human-Model	0.5907	0.0962	0.8820	0.4350	0.1363	0.6405
SPE	Human-Model (Challenging)	0.3559	-0.1679	0.6900	0.1429	-0.0777	0.3216
ENIGMA	Human-Model	**0.9666**	**0.9415**	**0.9792**	**0.9167**	**0.8717**	**0.9485**
ENIGMA	Human-Model (50% data)	0.9126	0.8506	0.9585	0.7790	0.6651	0.8647
ENIGMA	Human-Model (10% data)	0.7327	0.4544	0.9266	0.5214	0.3651	0.6492
ENIGMA	Human-Model (Challenging)	0.6505	0.5394	0.7762	0.5190	0.3168	0.6672

Table 2: The correlation between automatic metrics and language score obtained by interacting with human. We only present the average/min/max correlations to all 10 different language metrics in this table. For detailed numbers, please refer to Appendix D.3, Figure 23.

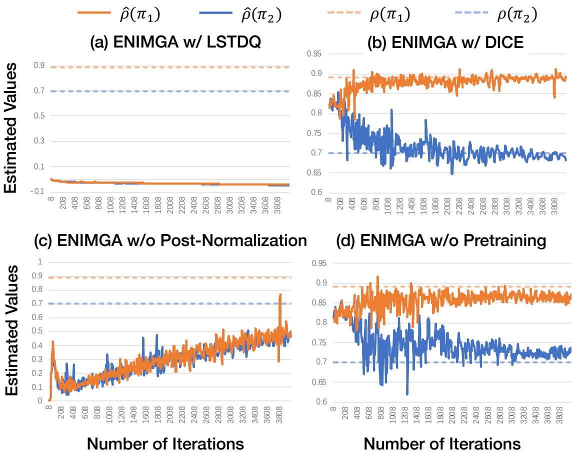


Figure 2: Value estimation using different methods for two target agents (π_1 and π_2) vs. # of iterations. Dotted lines denote the true rewards.

correlations. Moreover, we compare the correlations between estimated rewards and human evaluation scores under each language quality metric. Due to space limit, we only show the plots of ENIGMA and SPE under 3 out of 10 language quality metrics in Figure 4. Additional plots and detailed results can be found in Appendix D.3. We see that ENIGMA outperforms SPE and HCDFs under all language equality metrics.

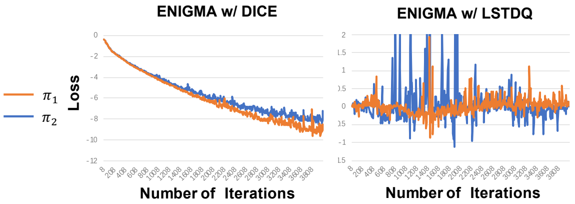
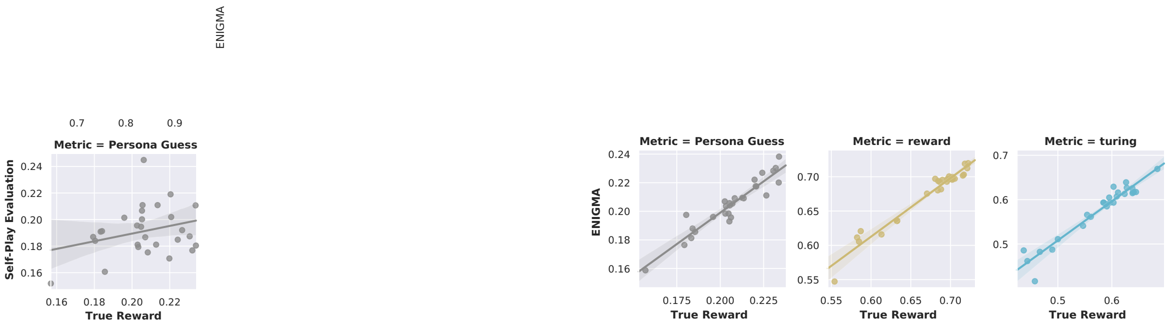


Figure 3: Training Objectives vs. Number of Iterations for two target agents.

- **Sample Efficiency of ENIGMA**. To demonstrate that ENIGMA is sample efficient, we test ENIGMA on randomly sub-sampled (10% and 50%) experience data. We found that even using only 10% of the experience data, ENIGMA still outperforms SPE and HCDFs.

- **Evaluation under Challenging Experience Data**. To make the evaluation more challenging, we further test ENIGMA by excluding the experience data obtained by the behavior policies similar to the target policy (see more details in Appendix D.3). We see that even with such challenging experience data, ENIGMA still outperforms SPE with trained on full data and HCDFs under



(a) SPE vs. Human Evaluation

(b) ENIGMA vs. Human Evaluation

Figure 4: Regression Plots. Only three metrics are presented. Please refer to Appendix D.3 for all plots.

almost all language quality metrics.

5 Discussions and Conclusion

Existing research on automatic evaluation of dialog systems can be categorized into static vs. dynamic evaluation. Most of existing research falls into static evaluation focusing on language quality of single-turn response or on task-completion given fixed dialog, while few literature emphasizes dynamic properties of an interactive environment and tries to considers the sequential interaction between a human and an agent, and thus it is more challenging.

We note that in both static and dynamic evaluations, the algorithms rely on the assumption of sufficient data coverage (explicitly or implicitly) to ensure reliable evaluation. For example, in static evaluation, BLEU score requires all reasonably good responses to be exactly covered by the experience data. More recently, Lowe et al. (2017) show that their method only works when the behavior policies include the target policy. Dynamic evaluation also assumes the sufficient coverage. We emphasize that it is the information-theoretic limit of all OPE methods (Jiang and Li, 2016), which requires the experience data to cover sufficient target policy behaviors to ensure accurate estimation. Therefore, we suggest the broader research community to release human-model interaction evaluation data to further promote research in automatic dialog systems evaluation.

In this paper, we develop a model-free dynamic evaluation framework, ENIGMA, which adopts the current state-of-the-art OPE method in reinforcement learning. Different from existing single-turn language quality metrics and model-based reinforcement learning methods, ENIGMA naturally takes into consideration the interactive and dynamic nature of conversations, while avoiding the difficulty of modeling complex human conversational behaviors. Our thorough experimental results demonstrate that ENIGMA significantly outperforms existing methods in terms of correlation with human evaluation scores. One potential future direction is to extend ENIGMA from off-policy evaluation to off-policy improvement, which aims to learn a dialog system based on experience data (Nachum et al., 2019b; Kallus and Uehara, 2020).

Broader Impact

This paper proposes ENIGMA, a model-free dynamic evaluation framework for dialog systems. We demonstrate that the ENIGMA framework can be used for both goal-oriented systems and chitchat systems. For AirDialog dataset, we collect experience data (human-model conversations), which does not contain any personal or sensitive information (see Figure 9, Appendix C). In all other experiments, we use publicly available data. We build our algorithms using public code bases and do not find any ethical concerns.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, Jennifer C Lai, and Robert L Mercer. 1992. An estimate of an upper bound for the entropy of english. *Computational Linguistics*, 18(1):31–40.

Zhehui Chen, Xingguo Li, Lin F Yang, Jarvis Haupt, and Tuo Zhao. 2018. On landscape of lagrangian functions and stochastic search for constrained nonconvex optimization. *arXiv preprint arXiv:1806.05151*.

Bo Dai, Albert Shaw, Niao He, Lihong Li, and Le Song.

2017. Boosting the actor with dual critic. *arXiv preprint arXiv:1712.10282*.

Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2020. Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, pages 1–56.

David DeVault, Anton Leuski, and Kenji Sagae. 2011. Toward learning and evaluation of dialogue policies with text examples. In *Proceedings of the SIGDIAL 2011 Conference*, pages 39–48.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. 2020. The second conversational intelligence challenge (convai2). In *The NeurIPS'18 Competition*, pages 187–208. Springer.

Yaqi Duan and Mengdi Wang. 2020. Minimax-optimal off-policy evaluation with linear function approximation. *arXiv preprint arXiv:2002.09516*.

Nouha Dziri, Ehsan Kamalloo, Kory W Mathewson, and Osmar Zaiane. 2019. Evaluating coherence in dialogue systems using entailment. *arXiv preprint arXiv:1904.03371*.

Sarah E Finch and Jinho D Choi. 2020. Towards unified dialogue system evaluation: A comprehensive analysis of current evaluation protocols. *arXiv preprint arXiv:2006.06110*.

Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. 2014. Bootstrapping dialog systems with word embeddings. In *Nips, modern machine learning and natural language processing workshop*, volume 2.

Michel Galley, Chris Brockett, Alessandro Sordoni, Yangfeng Ji, Michael Auli, Chris Quirk, Margaret Mitchell, Jianfeng Gao, and Bill Dolan. 2015. deltableu: A discriminative metric for generation tasks with intrinsically diverse targets. *arXiv preprint arXiv:1506.06863*.

Sudeep Gandhe and David Traum. 2016. A semi-automated evaluation metric for dialogue model coherence. In *Situated Dialog in Speech-Based Human-Computer Interaction*, pages 217–225. Springer.

Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. Dialogue response ranking training with large-scale human feedback data. *arXiv preprint arXiv:2009.06978*.

Asma Ghandeharioun, Judy Hanwen Shen, Natasha Jaques, Craig Ferguson, Noah Jones, Agata Lapedriza, and Rosalind Picard. 2019. Approximating interactive human evaluation with self-play for open-domain dialog systems. In *Advances in Neural Information Processing Systems*, pages 13658–13669.

Sarik Ghazarian, Johnny Tian-Zheng Wei, Aram Galstyan, and Nanyun Peng. 2019. Better automatic evaluation of open-domain dialogue systems with contextualized embeddings. *arXiv preprint arXiv:1904.10635*.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Ryuichiro Higashinaka, Toyomi Meguro, Kenji Imamura, Hiroaki Sugiyama, Toshiro Makino, and Yoshihiro Matsuo. 2014. Evaluating coherence in open domain conversational systems. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Daniel G Horvitz and Donovan J Thompson. 1952. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685.

Lishan Huang, Zheng Ye, Jinghui Qin, Liang Lin, and Xiaodan Liang. 2020. Grade: Automatic graph-enhanced coherence metric for evaluating open-domain dialogue systems. *arXiv preprint arXiv:2010.03994*.

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. 2019. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*.

Nan Jiang and Lihong Li. 2016. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661. PMLR.

Nathan Kallus and Masatoshi Uehara. 2019. Efficiently breaking the curse of horizon in off-policy evaluation with double reinforcement learning. *arXiv preprint arXiv:1909.05850*.

Nathan Kallus and Masatoshi Uehara. 2020. Statistically efficient off-policy policy gradients. In *International Conference on Machine Learning*, pages 5089–5100. PMLR.

Michail G Lagoudakis and Ronald Parr. 2003. Least-squares policy iteration. *Journal of machine learning research*, 4(Dec):1107–1149.

Tian Lan, Xian-Ling Mao, Wei Wei, Xiaoyan Gao, and Heyan Huang. 2020. Pone: A novel automatic evaluation metric for open-domain generative dialogue systems. *arXiv preprint arXiv:2004.02399*.

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*.

Junlong Li, Zhuosheng Zhang, Hai Zhao, Xi Zhou, and Xiang Zhou. 2020. Task-specific objectives of pre-trained language models for dialogue adaptation. *arXiv preprint arXiv:2009.04984*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

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*.

Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. 2018. Breaking the curse of horizon: Infinite-horizon off-policy estimation. In *Advances in Neural Information Processing Systems*, pages 5356–5366.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1116–1126.

Maja J Mataric. 1994. Reward functions for accelerated learning. In *Machine learning proceedings 1994*, pages 181–189. Elsevier.

Shikib Mehri and Maxine Eskenazi. 2020. Unsupervised evaluation of interactive dialog with dialogpt. *arXiv preprint arXiv:2006.12719*.

A. H. Miller, W. Feng, A. Fisch, J. Lu, D. Batra, A. Bordes, D. Parikh, and J. Weston. 2017. Parlai: A dialog research software platform. *arXiv preprint arXiv:1705.06476*.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *proceedings of ACL-08: HLT*, pages 236–244.

Sebastian Möller, Roman Englert, Klaus Engelbrecht, Verena Hafner, Anthony Jameson, Antti Oulasvirta, Alexander Raake, and Norbert Reithinger. 2006. Memo: towards automatic usability evaluation of spoken dialogue services by user error simulations. In *Ninth International Conference on Spoken Language Processing*.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. 2019a. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Advances in Neural Information Processing Systems*, pages 2318–2328.

Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. 2019b. Algaedice: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*.

Bo Pang, Erik Nijkamp, Wenjuan Han, Linqi Zhou, Yixian Liu, and Kewei Tu. 2020. Towards holistic and automatic evaluation of open-domain dialogue generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3619–3629, Online. Association for Computational Linguistics.

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.

Doina Precup. 2000. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, page 80.

Martin L Puterman. 1995. Markov decision processes: Discrete stochastic dynamic programming. *Journal of the Operational Research Society*, 46(6):792–792.

Tongzheng Ren, Jialian Li, Bo Dai, Simon S Du, and Sujay Sanghavi. 2021. Nearly horizon-free offline reinforcement learning. *arXiv preprint arXiv:2103.14077*.

Vasile Rus and Mihai Lintean. 2012. An optimal assessment of natural language student input using word-to-word similarity metrics. In *International Conference on Intelligent Tutoring Systems*, pages 675–676. Springer.

Ananya B Sai, Akash Kumar Mohankumar, Siddhartha Arora, and Mitesh M Khapra. 2020. Improving dialog evaluation with a multi-reference adversarial dataset and large scale pretraining. *Transactions of the Association for Computational Linguistics*, 8:810–827.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? how controllable attributes affect human judgments. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

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.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2019. Machine translation evaluation with bert regressor. *arXiv preprint arXiv:1907.12679*.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*.

Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2017. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. *arXiv preprint arXiv:1701.03079*.

AM Turing. 1950. Computing machinery and intelligence. *Mind*, 59(236):433–460.

Masatoshi Uehara, Jiawei Huang, and Nan Jiang. 2019. Minimax weight and q-function learning for off-policy evaluation. *arXiv*, pages arXiv–1910.

Cameron Voloshin, Hoang M Le, Nan Jiang, and Yisong Yue. 2019. Empirical study of off-policy policy evaluation for reinforcement learning. *arXiv preprint arXiv:1911.06854*.

Jie Wang, Rui Gao, and Hongyuan Zha. 2020a. Reliable off-policy evaluation for reinforcement learning. *arXiv preprint arXiv:2011.04102*.

Ruosong Wang, Dean P Foster, and Sham M Kakade. 2020b. What are the statistical limits of offline rl with linear function approximation? *arXiv preprint arXiv:2010.11895*.

Wei Wei, Quoc Le, Andrew Dai, and Jia Li. 2018. Airdialogue: An environment for goal-oriented dialogue research. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3844–3854.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413, Metz, France. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yang Xiang, Yaoyun Zhang, Xiaoqiang Zhou, Xiaolong Wang, and Yang Qin. 2014. Problematic situation analysis and automatic recognition for chinese online conversational system. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 43–51.

Tengyang Xie, Yifei Ma, and Yu-Xiang Wang. 2019. Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. In *Advances in Neural Information Processing Systems*, pages 9668–9678.

Mengjiao Yang, Ofir Nachum, Bo Dai, Lihong Li, and Dale Schuurmans. 2020. Off-policy evaluation via the regularized lagrangian. *arXiv preprint arXiv:2007.03438*.

Ming Yin and Yu-Xiang Wang. 2020. Asymptotically efficient off-policy evaluation for tabular reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3948–3958. PMLR.

Zhou Yu, Ziyu Xu, Alan W Black, and Alexander Rudnicky. 2016. Strategy and policy learning for non-task-oriented conversational systems. In *Proceedings of the 17th annual meeting of the special interest group on discourse and dialogue*, pages 404–412.

Tsuta Yuma, Naoki Yoshinaga, and Masashi Toyoda. 2020. uBLEU: Uncertainty-aware automatic evaluation method for open-domain dialogue systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 199–206, Online. Association for Computational Linguistics.

Hainan Zhang, Yanyan Lan, Liang Pang, Hongshen Chen, Zhuoye Ding, and Dawei Yin. 2020a. Modeling topical relevance for multi-turn dialogue generation. *arXiv preprint arXiv:2009.12735*.

Ruiyi Zhang, Bo Dai, Lihong Li, and Dale Schuurmans. 2020b. Gendice: Generalized offline estimation of stationary values. *arXiv preprint arXiv:2002.09072*.

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*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Tianyu Zhao, Divesh Lala, and Tatsuya Kawahara. 2020. Designing precise and robust dialogue response evaluators. *arXiv preprint arXiv:2004.04908*.

A

A.1

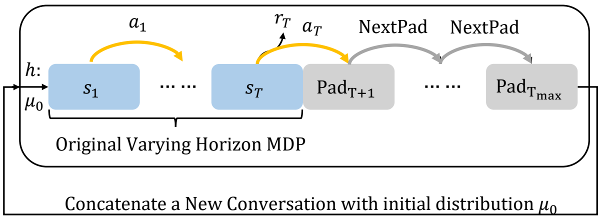

a_1
r_T
a_T
NextPad
NextPad
h:
μ_0
s_1
s_T
Pad_{T+1}
$\mathrm{Pad}_{T_{\max}}$
Original Varying Horizon MDP
Concatenate a New Conversation with initial distribution μ_0

Figure 5: Illustration of Augmented MDP with Infinite Horizon.

Theorem 1. The augmented MDP with infinite horizon satisfies the following properties:

- It has a unique stationary state-action visitation distribution $d^\pi(s, a)$;
- For the station-action pair (s_t, a_t) in a conversation h with padded pseudo states, we have

$$d^\pi(s_t, a_t) = \frac{1}{T_{\max}} \sum_{\{(s_k,a_k)\}_{k=1}^{t-1}} [\mu_0(s_1)\pi(a_1|s_1)P(s_2|a_1, s_1)\cdots P(s_t|a_{t-1}, s_{t-1})\pi(a_t|s_t)], \tag{8}$$

  where $\{(s_k, a_k)\}_{k=1}^{t-1}$ are the state-action pairs in the same conversation as (s_t, a_t);
- The policy value can be computed by sampling from $d^\pi(s, a)$, and we have

$$\rho_A(\pi) = \mathbb{E}_{(s,a)\sim d^\pi(s,a)}[R(s, a)] = \rho(\pi)/T_{\max}. \tag{9}$$

Proof. **First**, we prove that the augmented MDP has a unique stationary state-action visitation distribution shown in (8).

As the augmented MDP is periodic with period $T_{\max}$, the uniqueness and stationary distribution can not be immediately obtained by ergodicity of the MDP (the first two points of the Theorem).

To obtain the stationary state-action visitation distribution, we essentially need to solve the following equations:

$$d^\pi(s, a) = \sum_{(s',a')} d^\pi(s', a')P(s|s', a')\pi(a|s), \text{ for all } (s, a) \tag{10}$$

with $d^\pi(s, a)$ is a probability measure on the state-action space, i.e., $\sum_{(s,a)} d^\pi(s, a) = 1$.

We first group the state-action pairs by their dialog turns t. More specifically, we define $\mathcal{S}_t := \{s_t : s_t \text{ contains } t \text{ dialog turns}\}$, $\mathcal{A}_t := \{a_t : a_t \text{ is the response at the } t\text{−th dialog turn}\}$ and $\mathcal{Q}_t = \mathcal{S}_t \times \mathcal{A}_t$. We have the state space is the direct sum of state groups $\mathcal{S}_0 \oplus \mathcal{S}_1 \cdots \oplus \mathcal{S}_{T_{\max}} = \mathcal{S}$ and the action space is the union of all action groups $\bigcup_{t=1}^{T_{\max}} \mathcal{A}_t = \mathcal{A}$. We further have $\mathcal{Q}_0 \oplus \mathcal{Q}_1 \cdots \oplus \mathcal{Q}_{T_{\max}} = \mathcal{S} \times \mathcal{A} = \mathcal{Q}$. Notice that t is the number of dialog turns in original MDP, not the time step for the augmented MDP.

We then consider the quantity $S_t = \sum_{(s_t,a_t)\in\mathcal{Q}_t} d^\pi(s_t, a_t)$, which sum over the LHS of the Eq.(10) for each group of (s_t, a_t). We now expand the corresponding sum of the RHS of (10):

$$\begin{aligned}
S_t &= \sum_{(s_t,a_t)\in\mathcal{Q}_t} d^\pi(s_t, a_t) \\
&= \sum_{(s_t,a_t)\in\mathcal{Q}_t} \sum_{(s_{t-1},a_{t-1})\in\mathcal{Q}_{t-1}} d^\pi(s_{t-1}, a_{t-1})P(s_t|s_{t-1}, a_{t-1})\pi(a_t|s_t) \\
&= \sum_{(s_{t-1},a_{t-1})\in\mathcal{Q}_{t-1}} \sum_{(s_t,a_t)\in\mathcal{Q}_t} d^\pi(s_{t-1}, a_{t-1})P(s_t|s_{t-1}, a_{t-1})\pi(a_t|s_t) \\
&= \sum_{(s_{t-1},a_{t-1})\in\mathcal{Q}_{t-1}} d^\pi(s_{t-1}, a_{t-1}) \\
&= S_{t-1} \ (t > 1).
\end{aligned}$$

We have $S_1 = S_2 = \cdots = S_{T_{\max}} = S$. As $d^\pi(s, a)$ is a probability measure, we have the following unique solution for S_t's

$$S = \frac{1}{T_{\max}} \sum_{t=1}^{T_{\max}} S_t = \frac{1}{T_{\max}} \sum_{t=1}^{T_{\max}} \sum_{(s_t,a_t)\in\mathcal{Q}_t} d^\pi(s_t, a_t) = \frac{1}{T_{\max}} \sum_{(s,a)\in\mathcal{Q}} d^\pi(s, a) = \frac{1}{T_{\max}}.$$

As there is only one possibility for the last state-action pairs in all possible conversation that $s_{T_{\max}} = \mathrm{Pad}_{T_{\max}}, a_{T_{\max}} = \mathrm{NextPad}$, we have $d^\pi(\mathrm{Pad}_{T_{\max}}, \mathrm{NextPad}) = S_{T_{\max}} = \frac{1}{T_{\max}}$. We now consider (s_1, a_1), which is the first state-action pair of a conversation. We have

$$\begin{aligned} d^\pi(s_1, a_1) &= \sum_{(s',a')} d^\pi(s', a')P(s_1|s', a')\pi(a_1|s_1) \\ &= d^\pi(\mathrm{Pad}_{T_{\max}}, \mathrm{NextPad})P(s_1|\mathrm{Pad}_{T_{\max}}, \mathrm{NextPad})\pi(a_1|s_1) = \frac{1}{T_{\max}}\mu_0(s_1)\pi(a_1|s_1), \end{aligned} \tag{11}$$

which is the unique solution. For any $(s_t, a_t) \in \mathcal{Q}_t (t > 1)$, the previous state-action pairs in the same conversation must be in $\mathcal{Q}_{t-1}$. We have

$$\begin{aligned} d^\pi(s_t, a_t) &= \sum_{(s',a')} d^\pi(s', a')P(s_t|s', a')\pi(a_t|s_t) \\ &= \sum_{(s_{t-1},a_{t-1})\in\mathcal{Q}_{t-1}} d^\pi(s_{t-1}, a_{t-1})P(s_t|s_{t-1}, a_{t-1})\pi(a_t|s_t). \end{aligned} \tag{12}$$

Based on (11) and (12), we can obtain the unique solution for (10):

$$\begin{aligned} &d^\pi(s_t, a_t) \\ &= \sum_{(s_{t-1},a_{t-1})\in\mathcal{Q}_{t-1}} d^\pi(s_{t-1}, a_{t-1})P(s_t|s_{t-1}, a_{t-1})\pi(a_{t-1}|s_{t-1}) \\ &= \sum_{(s_{t-1},a_{t-1})\in\mathcal{Q}_{t-1}} \sum_{(s_{t-2},a_{t-2})\in\mathcal{Q}_{t-2}} d^\pi(s_{t-2}, a_{t-2})P(s_{t-1}|s_{t-2}, a_{t-2})\pi(a_{t-1}|s_{t-1})P(s_t|s_{t-1}, a_{t-1})\pi(a_t|s_t) \\ &\cdots \\ &= \frac{1}{T_{\max}} \sum_{\{(s_k,a_k)\}_{k=1}^{t-1}} [\mu_0(s_1)\pi(a_1|s_1)P(s_2|a_1, s_1) \cdots P(s_t|a_{t-1}, s_{t-1})\pi(a_t|s_t)], \end{aligned}$$

where we omit the constraint of $\mathcal{Q}_t$ as the transition kernel P naturally satisfies the constraints.

Till now, we have shown that the augmented MDP has a unique stationary state-action visitation distribution shown in (8) (the first two points of the Theorem).

Next, we show that the policy value of the policy π under the augmented MDP is proportional to its counterpart under the original MDP without the augmentation (the third point of the Theorem).

Recall that the expected reward of original MDP (1) is defined as

$$\begin{aligned} \rho(\pi) &= \mathbb{E}_{h\sim\mu_0,\pi,\mathcal{E}}[R(s_T, a_T)] = \sum_{T=1}^{T_{\max}} \sum_h Pr(h, h \text{ has } T \text{ turns})R(s_T, a_T) \\ &= \sum_{T=1}^{T_{\max}} \sum_{\{(s_k,a_k)\}_{k=1}^{T-1}} \mu_0(s_1)\pi(a_1|s_1)P(s_2|a_1, s_1) \cdots P(s_T|a_{T-1}, s_{T-1})\pi(a_T|s_T) \\ &\times \mathbb{1}(a_T \text{ End Conversation})R(s_T, a_T), \end{aligned}$$

where T is the number of turns in the original dialog before padding. Recall that, the MDP only obtain non-zero reward when the dialog ends, (i.e., when a End Conversation). On the other hand, Due to the existence of unique stationary distribution, the policy value of π for the augmented MDP (2) can written as:

$$\begin{aligned}\rho_A(\pi) &= \mathbb{E}_{(s,a)\sim d^\pi(s,a)}[R(s,a)] = \mathbb{E}_{(s,a)\sim d^\pi(s,a)}[\mathbb{1}(a \text{ End Conversation})R(s,a)] \\ &= \frac{1}{T_{\max}} \sum_{t=1}^{T_{\max}} \sum_{\{(s_k,a_k)\}_{k=1}^{t}} \mu_0(s_1)\pi(a_1|s_1)P(s_2|a_1,s_1)\cdots P(s_t|a_{t-1},s_{t-1})\pi(a_t|s_t) \\ &\times \mathbb{1}(a_t \text{ End Conversation})R(s_t,a_t) \\ &= \frac{1}{T_{\max}}\rho(\pi).\end{aligned}$$

□

Can we directly apply infinite-horizon augmentation without padding? The answer is *NO*. Here we use an example to illustrate the difference between ρ_{A} and ρ and why we need to pad every dialog to have the same length for using OPE:

Example 1. Suppose you have two experience dialogs $a_0 \to \cdots \to a_{t_1}$ and $b_0 \to \cdots \to b_{t_2}$ with rewards 0 and 1 respectively. For the target policy, dialogs has per-episode density 0.2 and 0.8 respectively. The true value of such policy is $0 \times 0.2 + 1 \times 0.8 = 0.8$. The corresponding per-state density of $[a_0, \cdots, a_{t_1}]$ is $\frac{0.2}{0.2\times t_1 + 0.8 \times t_2}$ and the one for $[b_0, \cdots, b_{t_1}]$ is $\frac{0.8}{0.2\times t_1+0.8\times t_2}$. The value in the new augmented MDP is $\frac{0.2*0+0.8*1}{0.2\times t_1+0.8\times t_2}$, which depends on the dialog turns and can not be directly turned into policy value in the original MDP.

A.2 Function Approximation with Pre-Trained Language Models

We can compute all state-action pairs for the same dialog in a parallel way as shown in Figure 6. The input to the RoBERTa encoder consists of three parts, word tokens, position ids, and token types.

Notation: an experience dialog $h = \{e_0, a_1, e_1, ..., a_T\}$, and the corresponding response generated by the target policy π, $\{a'_t = \pi(s_t)\}_{t=1}^{T}$.

Word Tokens. The input token is the concatenation of responses $\{e_0, a_1, a'_1, e_1, ..., e_{T-1}, a_T, a'_T\}$.

Position Ids. The position ids is separately calculated for each response. For e_i, the position ids is from $l_{2i} = \sum_{j<i} \text{len}(e_j) + \sum_{j\le i} \text{len}(a_j)$ to $l_{2i+1} = l_{2i} + \text{len}(e_i)$, where $\text{len}(\cdot)$ denotes the number of tokens of a given response. For a_i, the position ids is from l_{2i-1} to l_{2i}. For a'_i, the position ids is from l_{2i-1} to $l'_{2i} = l_{2i-1} + \text{len}(a'_i)$.

Token types. For e_i's, the token types are 0 which denotes human responses. For a_i's and a'_i's, the token types are 1 which denotes agent responses.

Attention Masking. We need to modify the attention masks to prevent tokens from attending future responses. Specifically, the attention masks make sure:

1. The tokens in each response can be mutually attended;
2. e_i attends to $\{e_0, a_1, e_1, a_2, ..., e_{i-1}, a_i\}$;
3. a_i attends to $\{e_0, a_1, e_1, a_2, ..., a_{i-1}, e_{i-1}\}$;
4. a'_i attends to $\{e_0, a_1, e_1, a_2, ..., a_{i-1}, e_{i-1}\}$;

A.3 ENIGMA with regularized DICE

Step 2: Solve Min-Max optimization with function approximator

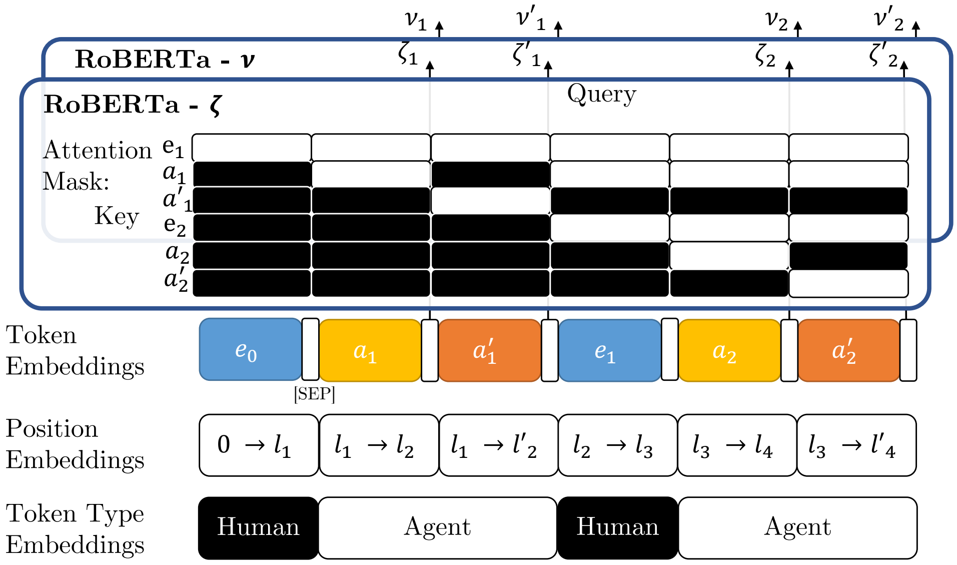


Figure 6: RoBERTa-ζ and RoBERTa-ν

Algorithm 2 Dialog OPE using regularized DICE

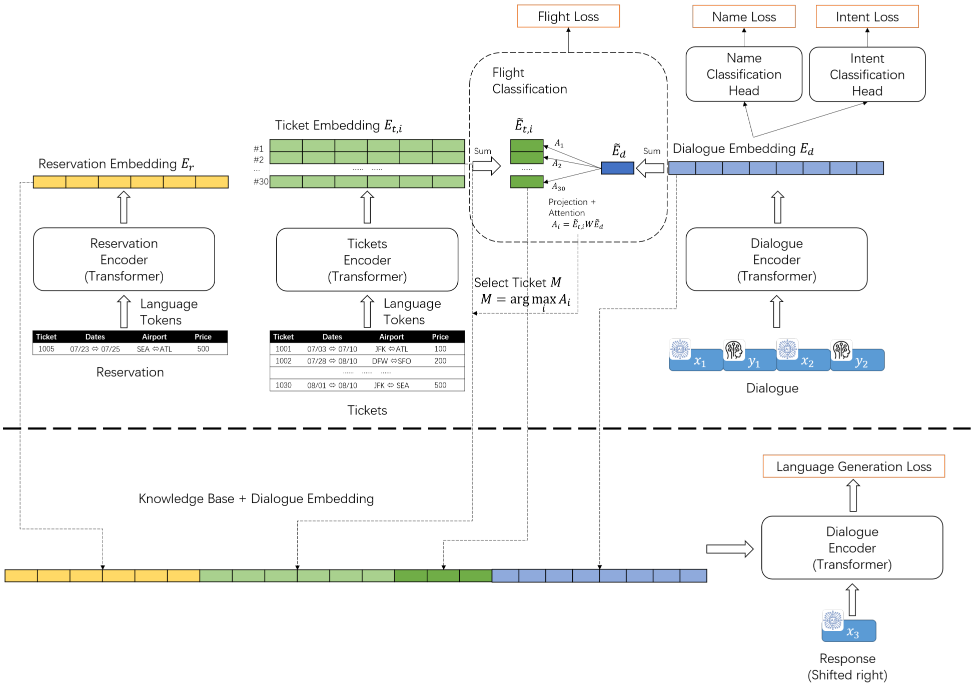
```
Input: Experience Dialog with rewards $\mathcal{D} = \{(h_i = \{e^{(i)}_0, a^{(i)}_1, e^{(i)}_1, ..., a^{(i)}_{T_i}\}, r^{(i)})\}_{i=1}^{N}$, Target Policy $\pi$,
    Padding Length $T_{\max}$, Regularization function $f$, DICE hyper-parameters $\alpha_\zeta$, $\alpha_R$
Output: Performance Estimation $\widehat{\rho}_n(\pi)$
Parameters: $\zeta = \{$ RoBERTa-$\zeta$, $[\zeta_{\text{pad},t}]_{1 \le t \le T_{\max}}\}$,$\nu = \{$ RoBERTa-$\nu$, $[\nu_{\text{pad},t}]_{1 \le t \le T_{\max}}\}$, $\lambda$
Generate OPE Data via Pseudo State Padding
for $(h_i, r^{(i)})$ in $\mathcal{D}$ do
    for $t$ in $1, \cdots, T_i$ do
        $\widetilde{a}^{(i)}_t \sim \pi(\{e^{(i)}_0, a^{(i)}_1, e^{(i)}_1, ..., e^{(i)}_{t-1}\})$ // Sample Action From Target Policy
    end for
end for
$\widetilde{\mathcal{D}} = \{(\widetilde{h}_i = e^{(i)}_0, a^{(i)}_1, e^{(i)}_1, ..., a^{(i)}_{T_i}\}, r^{(i)})\}_{i=1}^{N}$
Estimate $\zeta$ by Regularized DICE
while Not Converged do
    Sample Mini-Batch $\mathcal{B} \subset \widetilde{\mathcal{D}}$
    for $(\widetilde{h}_i, r^{(i)})$ in $\mathcal{B}$ do
        $\zeta^{(i)}_0 = \zeta_{\text{pad},T_{\max}}$, $\nu^{(i)}_0 = \nu'^{(i)}_0 = \nu_{\text{pad},T_{\max}}$ // infinite-horizon concatenation
        for $t$ in $1, \cdots, T_i$ do
            $\zeta^{(i)}_t =$ RoBERTa-$\zeta(e^{(i)}_0, a^{(i)}_1, ..., a^{(i)}_{t-1}, e^{(i)}_{t-1}, a^{(i)}_t)$
            $\nu^{(i)}_t =$ RoBERTa-$\nu(e^{(i)}_0, a^{(i)}_1, ..., a^{(i)}_{t-1}, e^{(i)}_{t-1}, a^{(i)}_t)$
            $\nu'^{(i)}_t =$ RoBERTa-$\nu(e^{(i)}_0, a^{(i)}_1, ..., a^{(i)}_{t-1}, e^{(i)}_{t-1}, \widetilde{a}^{(i)}_t)$
        end for
        for $t$ in $T_i + 1, \cdots, T_{\max}$ do
            $\zeta^{(i)}_t = \zeta_{\text{pad},t}$, $\nu^{(i)}_t = \nu'^{(i)}_t = \nu_{\text{pad},t}$ // pseudo state padding
        end for
        $\ell_i = \frac{1}{T_{\max}}[\sum_{t=0}^{T_{\max}-1}[\zeta_t(\nu'^{(i)}_{t+1} - \nu^{(i)}_t) + \lambda(\zeta^{(i)}_t - 1) - \alpha_\zeta f(\zeta^{(i)}_t)]]$
    end for
    $L_D(\zeta, \nu, \lambda) = \frac{1}{|\mathcal{B}|}\sum_{i \in \mathcal{B}} \ell_i$
    SGD update based on $\frac{\partial L_D}{\partial \nu}, \frac{\partial L_D}{\partial \lambda}, \frac{\partial - L_D}{\partial \zeta}$ (Gradient Reversal).
end while
Estimate Average Reward with Post-Normalization
for $(h_i, r^{(i)})$ in $\mathcal{D}$ do
    $\zeta_i =$ RoBERTa-$\zeta(e^{(i)}_0, a^{(i)}_1, ..., a^{(i)}_{T_i-1}, e^{(i)}_{T_i-1}, a^{(i)}_{T_i})$
end for
Return $\widehat{\rho}_n(\pi) = \sum_i \zeta_i r^{(i)} / \sum_i \zeta_i$
```

B Experiment Set-Up

In the following experiments, we share the RoBERTa encoder for RoBERTa-ζ and RoBERTa-ν. On the top of RoBERTa-ζ and RoBERTa-ν, it is a two-layer fully connected neural network equipped with GeLU activation (Hendrycks and Gimpel, 2016) and the same hidden dimension as RoBERTa. The RoBERTa encoder is initialized from RoBERTa-base checkpoint (Liu et al., 2019). We simply use reverse gradients for the mini-max updates. We set learning rate as 2×10^{-4} and use inverse square root learning rate decay. We impose the gradient norm clipping with the maximum norm $\|\cdot\|_2 \leq 10$. We use 100 times larger learning rate for optimizing λ, 2 times larger learning rate for RoBERTa-ν. In (6), we set $\alpha_\zeta = 1$, $f(x) = x^2$ as suggested in Yang et al. (2020). We maintain $\zeta \geq 0$ by adding a square activation at the end of RoBERTa-ζ. The source code is built based on Transformers (Wolf et al., 2019), AirDialog (Wei et al., 2018), and ParlAI (Miller et al., 2017). All experiments are conducted on a machine with 8× V100 GPUs on Google Cloud.

C Transformer-Based Agents for AirDialog

Seller Agent Transformer Architecture There are four components for the encoder: ticket encoder, reservation encoder, dialog encoder, and task-specific heads (intent classification head and name classification head). All tickets and reservation are converted to natural languages. Noticing that, we always append a pseudo ticket in the ticket database representing "no ticket found" situation. The architecture is illustrated in Figure 7.



Ticket	Dates	Airport	Price
1005	07/23 ⇔ 07/25	SEA ⇔ATL	500

Ticket	Dates	Airport	Price
1001	07/03 ⇔ 07/10	JFK ⇔ATL	100
1002	07/28 ⇔ 08/10	DFW ⇔SFO	200
......	......	......	
1030	08/01 ⇔ 08/10	JFK ⇔ SEA	500

Figure 7: Transformer-based Seller Agent

Customer Agent Transformer Architecture There are two components for the encoder: intent encoder, reservation encoder. All intents are converted to natural languages. The architecture is illustrated in Figure 8.

Training Objective Besides the language generation loss $\mathcal{L}_l$, the training objective for seller consists of three parts: name loss, flight loss, intent loss:

$$\min_\theta \mathcal{L}_s(\theta) = \mathcal{L}_l(\theta) + \lambda_n \mathcal{L}_{\text{name}}(\theta) + \lambda_f \mathcal{L}_{\text{flight}}(\theta) + \lambda_i \mathcal{L}_{\text{intent}}(\theta) \tag{13}$$

The customer agent is trained with normal language generation loss.

Benchmark We compare the proposed model with the current AirDialog RNN baseline (Wei et al., 2018). As can be seen, the agent used in this paper are significantly stronger than the baseline agent used in Wei et al. (2018).

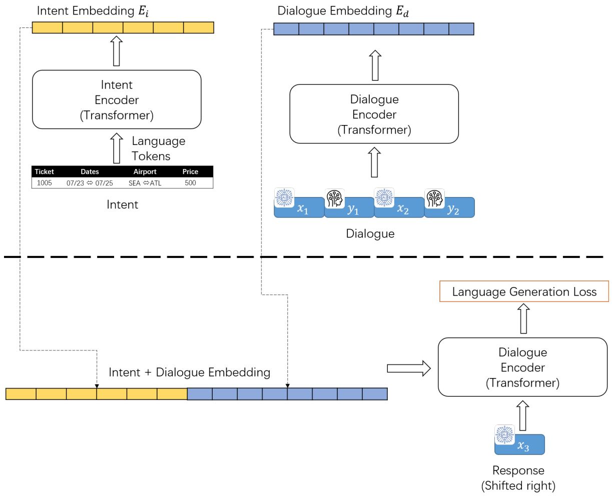


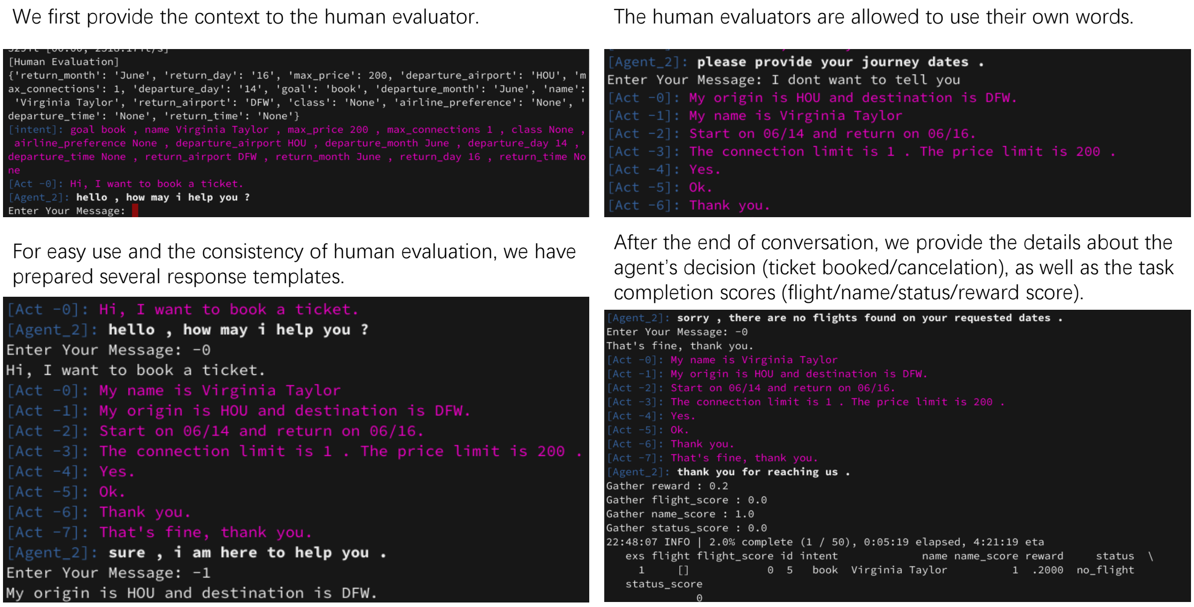
Figure 8: Transformer-based Customer Agent

Model	BLEU (C)	BLEU (S)	PPL (C)	PPL (S)	Reward	Name	Flight	Status
RNN	22.92	32.95	-	-	0.23	0.41	0.13	0.29
Ours	31.78	31.70	1.671	1.843	0.702	1.00	0.547	0.761

Table 3: Benchmark of the proposed transformer based agent. 'C' means customer, 'S' means seller. Reward, Name, Flight, status are the task-specific scores obtained from self-play evaluation.

Hyper-Parameters For training 24 seller agents used in Section 4, we varies the size of training data (number of training dialogs) from $5K$ to the full size and varies λ_i and λ_f from 0.0001 to 1. For training the customer agent used in self-play evaluation, we use the full training data and tune the hyperparameters based on the BLEU score evaluated using the validation set.

Human Evaluation The human evaluation is collected from 20 different Ph.D. students majored in Math/Stats/CS/IEOR. We provide detailed guidelines to the human evaluator that they have to speak to the agents with similar tone. Figure 9 presents the screen shots of the human evaluation software.



We first provide the context to the human evaluator.

```
329it [00:00, 2318.17it/s]
[Human Evaluation]
{'return_month': 'June', 'return_day': '16', 'max_price': 200, 'departure_airport': 'HOU', 'm
ax_connections': 1, 'departure_day': '14', 'goal': 'book', 'departure_month': 'June', 'name':
 'Virginia Taylor', 'return_airport': 'DFW', 'class': 'None', 'airline_preference': 'None', '
departure_time': 'None', 'return_time': 'None'}
[intent]: goal book , name Virginia Taylor , max_price 200 , max_connections 1 , class None ,
 airline_preference None , departure_airport HOU , departure_month June , departure_day 14 ,
departure_time None , return_airport DFW , return_month June , return_day 16 , return_time No
ne
[Act -0]: Hi, I want to book a ticket.
[Agent_2]: hello , how may i help you ?
Enter Your Message:
```

The human evaluators are allowed to use their own words.

```
[Agent_2]: please provide your journey dates .
Enter Your Message: I dont want to tell you
[Act -0]: My origin is HOU and destination is DFW.
[Act -1]: My name is Virginia Taylor
[Act -2]: Start on 06/14 and return on 06/16.
[Act -3]: The connection limit is 1 . The price limit is 200 .
[Act -4]: Yes.
[Act -5]: Ok.
[Act -6]: Thank you.
```

For easy use and the consistency of human evaluation, we have prepared several response templates.

```
[Act -0]: Hi, I want to book a ticket.
[Agent_2]: hello , how may i help you ?
Enter Your Message: -0
Hi, I want to book a ticket.
[Act -0]: My name is Virginia Taylor
[Act -1]: My origin is HOU and destination is DFW.
[Act -2]: Start on 06/14 and return on 06/16.
[Act -3]: The connection limit is 1 . The price limit is 200 .
[Act -4]: Yes.
[Act -5]: Ok.
[Act -6]: Thank you.
[Act -7]: That's fine, thank you.
[Agent_2]: sure , i am here to help you .
Enter Your Message: -1
My origin is HOU and destination is DFW.
```

After the end of conversation, we provide the details about the agent's decision (ticket booked/cancelation), as well as the task completion scores (flight/name/status/reward score).

```
[Agent_2]: sorry , there are no flights found on your requested dates .
Enter Your Message: -0
That's fine, thank you.
[Act -0]: My name is Virginia Taylor
[Act -1]: My origin is HOU and destination is DFW.
[Act -2]: Start on 06/14 and return on 06/16.
[Act -3]: The connection limit is 1 . The price limit is 200 .
[Act -4]: Yes.
[Act -5]: Ok.
[Act -6]: Thank you.
[Act -7]: That's fine, thank you.
[Agent_2]: thank you for reaching us .
Gather reward : 0.2
Gather flight_score : 0.0
Gather name_score : 1.0
Gather status_score : 0.0
22:48:07 INFO | 2.0% complete (1 / 50), 0:05:19 elapsed, 4:21:19 eta
   exs flight flight_score id intent             name name_score reward     status  \
     1     []            0  5   book  Virginia Taylor          1  .2000  no_flight
   status_score
              0
```

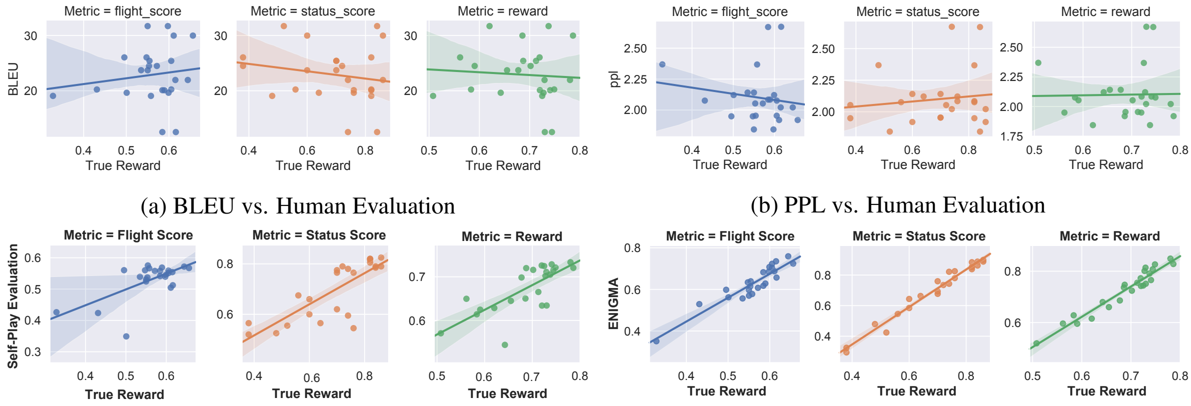
Figure 9: Screen Shots of Human Evaluation Software

D Additional Experiment

D.1 AirDialog

Regression Plot

We present the regression plot for the full setting in Figure 10 and for the selected agent in Figure 11.



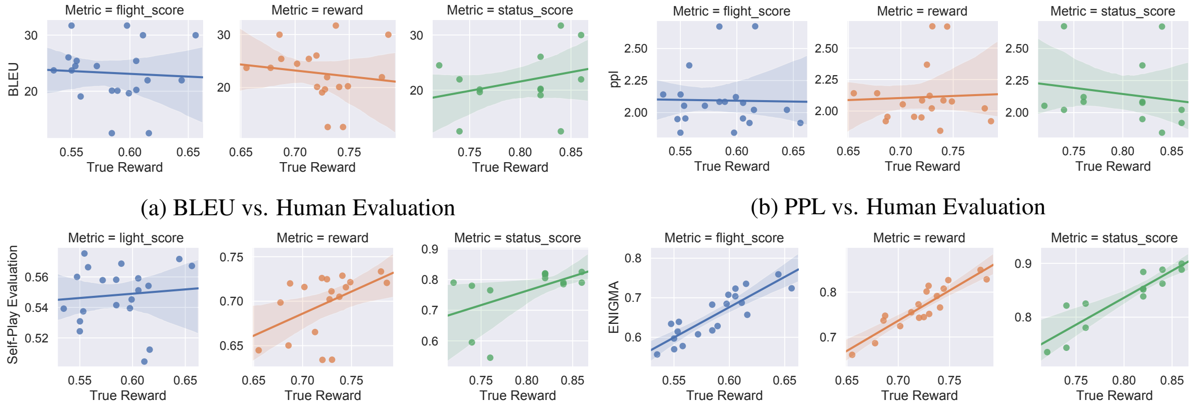
(a) BLEU vs. Human Evaluation

(b) PPL vs. Human Evaluation

(c) SPE vs. Human Evaluation

(d) ENIGMA vs. Human Evaluation

Figure 10: Regression Plot. The x-axis is the average reward obtained by chatting with human. The y-axis is BLEU/PPL/the reward estimated by ENIGMA. Different colors denotes different type of rewards (flight score, status score, and overall reward). The solid line is obtained by linear regression and the shaded region indicates 95% confidence interval. (see more in *seaborn* packages).



(a) BLEU vs. Human Evaluation

(b) PPL vs. Human Evaluation

(c) SPE vs. Human Evaluation

(d) ENIGMA vs. Human Evaluation

Figure 11: Regression Plot for "selected agent" Setting. The x-axis is the average reward obtained by chatting with human. The y-axis is BLEU/PPL/the reward estimated by ENIGMA/Self-Play Evaluation (SPE). Different colors denotes different type of rewards (flight score, status score, and overall reward). The solid line is obtained by linear regression and the shaded region indicates 95% confidence interval. (see more in *seaborn* packages).

Training Curves

We show the training curves of the ENIGMA in Figure 12. Here four models are presented, the best model (ranked 100%), model ranked as 50%, model ranked as 25% and the worst model (ranked 0%). As can been seen the estimated reward estimation converges steadily to it's true values.

Ablation Study

Here we provide large figures (Figure 13 and Figure 14) for the ablation study mentioned in Section 4.

D.2 Additional Results for Rule-Based Agents of AirDialog

• *Rule-Rule (R-R)*: Both customer and seller agents are rule based. We fix the customer rule-based model and construct and evaluate 6 seller agents. The strongest agent can perfectly interpret the intent of

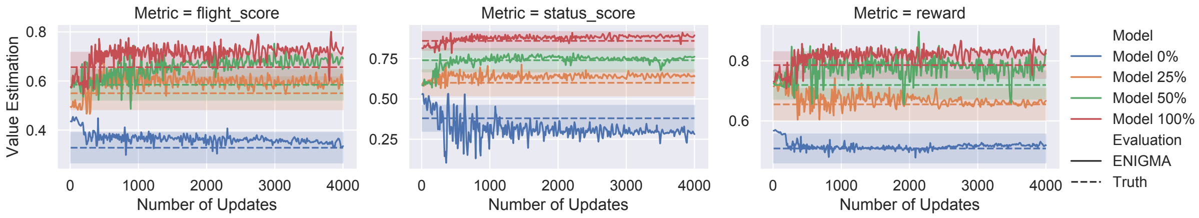


Figure 12: Learning curve for AirDialog. The x-axis is the number of mini-max updates, while y-axis is the estimated values. The straight line is the true reward, while the shaded region denotes the 90% confidence interval. The true reward and the confidence interval is obtained via different evaluation chats between the agents and the environment (model/human). Different colors denotes different agents.

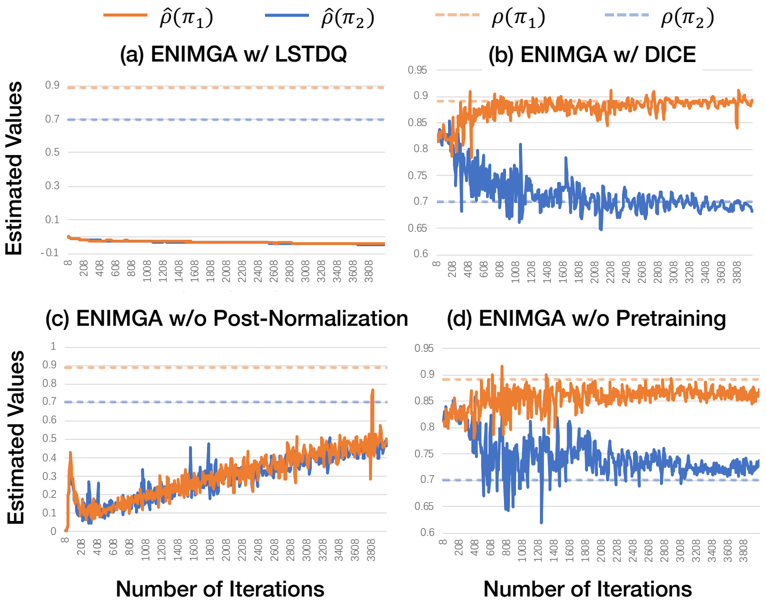


Figure 13: Reward estimation of two target agents (π_1 and π_2) vs. # of iterations. Dotted lines represents true rewards.

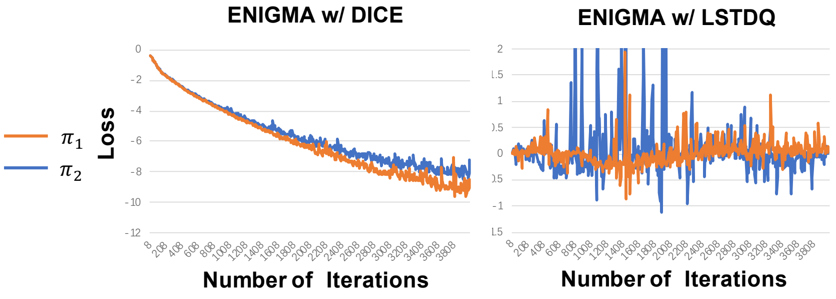


Figure 14: Loss value of two target agents during mini-max optimization.

rule-based customers. While the weaker agents interprets the intent with different levels of noise. The learning curve is presented in Figure 15.

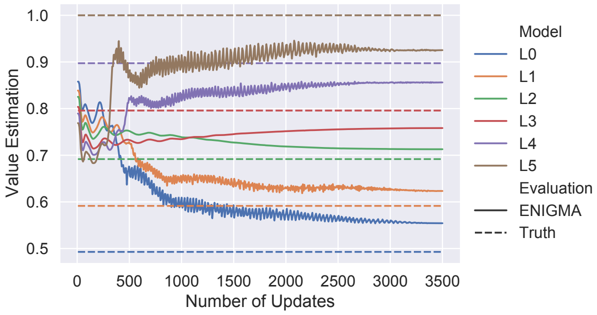


Figure 15: Learning Curve under Rule-Rule setting

Setting	**Method**	**Pearson Correlation**			**Spearman's Rank Correlation**		
		Flight Score	**Status Score**	**Reward**	**Flight Score**	**Status Score**	**Reward**
R-R	BLEU	0.1981	-0.0067	0.0980	0.1525	0.0009	0.0924
	ppl	-0.1584	-0.0610	-0.1209	-0.2475	-0.1060	-0.1178
	ENIGMA	**0.9687**	**0.9947**	**0.9874**	**0.8800**	**0.9872**	**0.9574**

Table 4: The correlation between two metrics. Each column is a task completion score obtained by interacting with the environments under R-R setting. Each row is an automatic metric.

D.3 ConvAI2

Training Curves.

Similar to the AirDialog dataset, we also show the training curves for the agents ranked at 100%, 50%, 25%, 0% in Figure 16. ENIGMA also converges steadily to the true values within a resonable error.

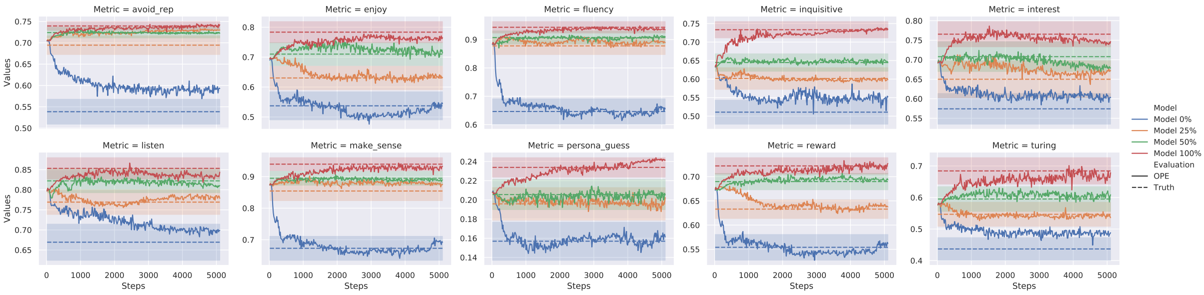


Figure 16: Learning curve for ConvAI2. The x-axis is the number of mini-max updates, while y-axis is estimated values. The straight line is the true reward, while the shaded area denotes the 95% confidence interval. The true reward and the confidence interval is obtained via different evaluation chats between the agents and human. Different colors denotes different agents.

Regression Plot.

We present the regression plot for the all 10 metrics in setting in Figure 17. The corresponding corresponding correlation is presented in Table 5. For comparison, we present the regression plot for self-play evaluation in Figure 18.

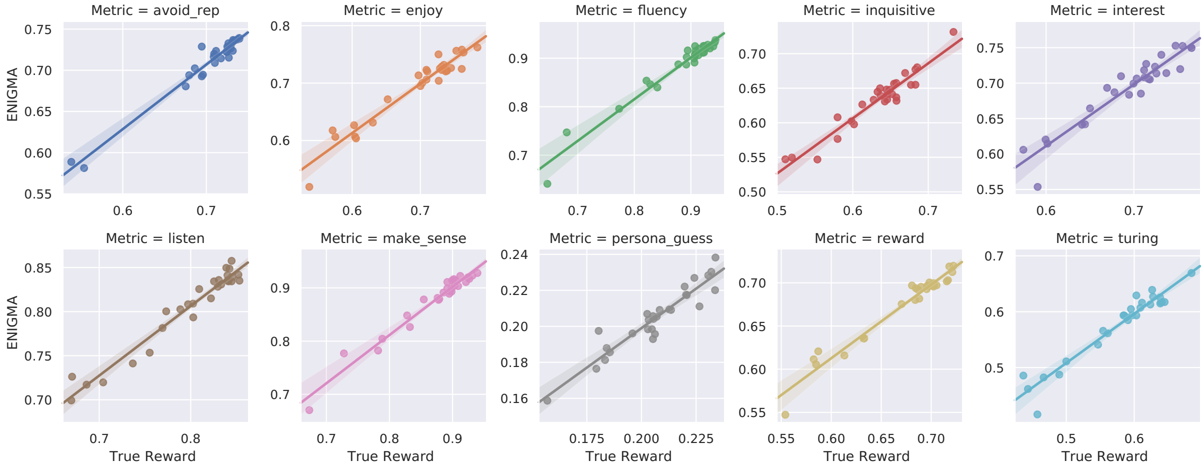


Figure 17: ENIGMA vs. Human Evaluation for ConvAI2. The x-axis is the average reward obtained by chatting with human. The y-axis it the reward estimated by ENIGMA. Different colors represent different language quality metrics. The solid line is obtained by simple linear regression.

Experience Data. To analysis how many human-model evaluation dialogs are needed, we analysis ENIGMA error under different sizes of the experience data. For ConvAI2, we compare the error for using 100% data, 50% data and 10% data. As shown in Table 5 and Figure 19, when we use half of the data, the error is similar to the one using full data. If we only use 10% data, ENIGMA becomes very inaccurate. OPE under low resource setting remains very challenging.

In Figure 19, we study the estimation error under different sizes of the experience data. As can be seen, when using 50% data, the reward value estimation is very similar to the one of using full data. When using only 10% data, the error is larger and ENIGMA has lower correlation with the true reward.

A More Challenging Setting.

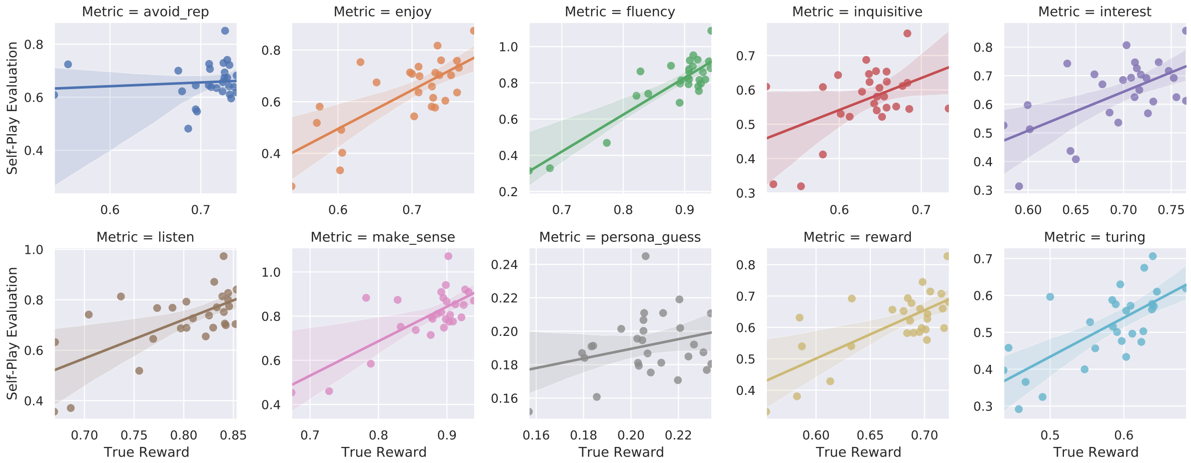


Figure 18: Self-Play Evaluation vs. Human Evaluation for ConvAI2. The x-axis is the average reward obtained by chatting with human. The y-axis it the reward estimated by self-play evaluation Different colors represent different language quality metrics. The solid line is obtained by simple linear regression.

Pearson Correlation					
Setting	**Avoid Rep.**	**Enjoy**	**Fluency**	**Inquisitive**	**Interest**
Full Data	0.9792	0.9661	0.9767	0.9584	0.9488
50% Data	0.9573	0.9046	0.9550	0.9237	0.8644
10% Data	0.9266	0.6595	0.8910	0.8286	0.5052
Selected Data	0.6944	0.6759	0.7762	0.5605	0.5820
Setting	**Listen**	**Make Sense**	**Persona**	**Reward**	**Turing**
Full Data	0.9754	0.9788	0.9415	0.9773	0.9637
50% Data	0.8971	0.9585	0.8770	0.9374	0.8506
10% Data	0.7455	0.8100	0.4544	0.8240	0.6825
Selected Data	0.5520	0.7402	0.6879	0.6968	0.5394

Spearman's rank correlation					
Setting	**Avoid Rep.**	**Enjoy**	**Fluency**	**Inquisitive**	**Interest**
Full Data	0.8905	0.9070	0.9178	0.8717	0.9210
50% Data	0.7558	0.7980	0.6651	0.8482	0.7727
10% Data	0.4128	0.6147	0.6492	0.6335	0.4713
Selected Data	0.5138	0.5561	0.4522	0.3168	0.6027
Setting	**Listen**	**Make Sense**	**Persona**	**Reward**	**Turing**
Full Data	0.9240	0.9448	0.9205	0.9485	0.9213
50% Data	0.7784	0.8647	0.8293	0.7750	0.7026
10% Data	0.3914	0.5096	0.3651	0.5774	0.5893
Selected Data	0.4585	0.6126	0.5844	0.6672	0.4259

Table 5: The correlation between different metrics and ENIGMA estimation. Each column is each average language quality score obtained by chatting with human. Different rows represent different experience data ENIGMA used.

Considering that some target agents are similar to the behavior policies with only slight difference in the way of decoding, they might yield very the similar dialog when the human acts in the same way. Specifically, in the data collection process, the target model might yield the responses that are very similar to the ones of the behavior policy for all turns in the dialog: $\mathrm{EditDistance}(a_t, a'_t) \leq 15 \;\; \forall 0 \leq t \leq T$. For a more realistic setting, we consider removing these highly overlapped dialogs after the data collection

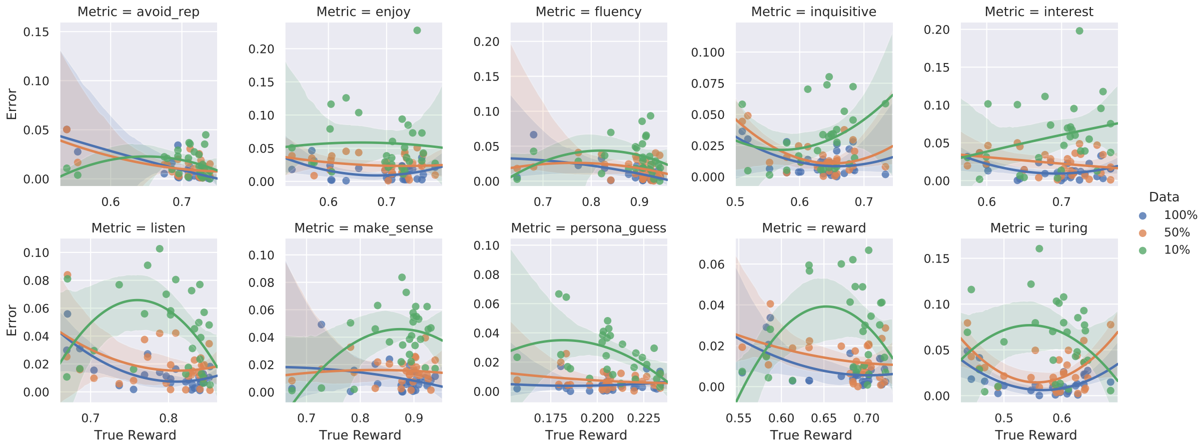


Figure 19: Error Analysis on Convai2 under different data size. The x-axis is the true average reward. The y-axis is the ENIGMA error. The solid line is the fitted quadratic function. Blue, orange, green colors represent 100%, 50%, 10% datasets respectively.

process. This setting is very challenging that the target policy behavior is less covered by the experience data and ENIGMA can only hopefully generalize via pre-trained RoBERTa. The results are shown in Figure 20 and Table 5. As can be seen, this setting remains challenging as the Pearson correlation is between 0.5 and 0.8. For comparison, we present the regression plot for self-play evaluation using this challenging subset of the experience data in Figure 21.

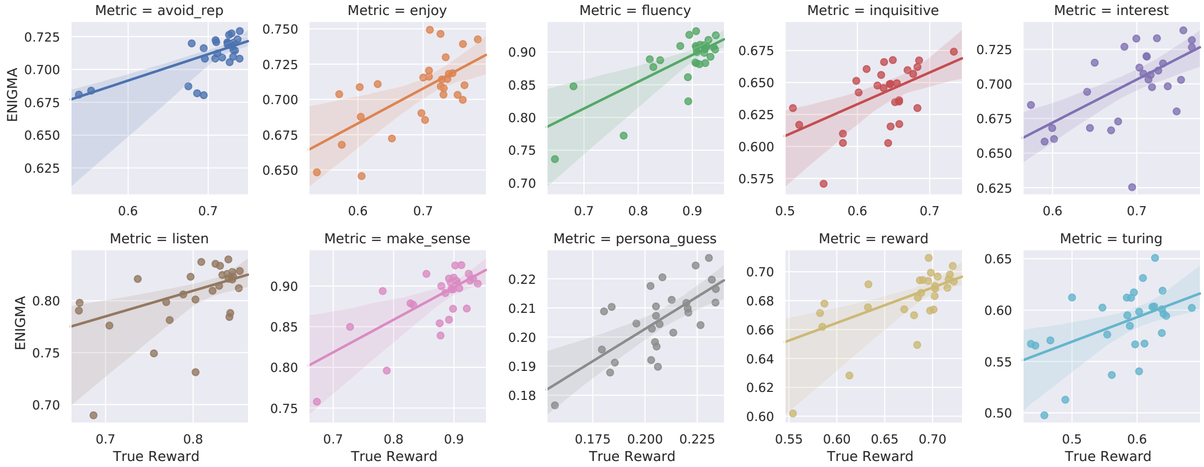


Figure 20: ENIGMA vs. Human Evaluation for ConvAI2 under the challenging setting. The x-axis is the average reward obtained by chatting with human. The y-axis it the reward estimated by ENIGMA. Different colors represent different language quality metrics. The solid line is obtained by simple linear regression.

We remark that such experiments can also be done for AirDialog. However, due to the limitation that most agents are just learning template responses due to the goal-oriented nature, removing overlapped dialogs results in an extremely incomplete experience dataset. For example, most "cancelation" dialogs will be removed since they are very simple and basically the same for different agents. As a result ENIGMA can not make a reasonable estimation due to the highly incomplete experience data.

Figure 22 compares the error of ENIGMA between using the normal experience data and the selected challenging one. As can be seen, the error using the selected data is larger particularly for the agents with exceptionally low/high true reward. That indicates the problem of the lack of dialog coverage is exaggerated under the challenging setting, while the ENIGMA estimation remains accurate when there is sufficient dialog coverage.

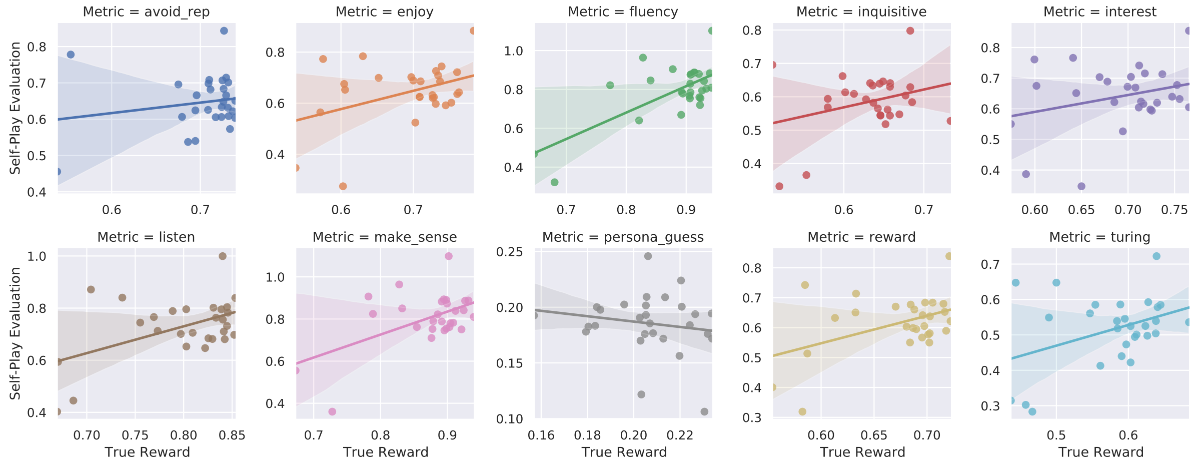


Figure 21: Self-Play Evaluation vs. Human Evaluation for ConvAI2 under the challenging setting. The x-axis is the average reward obtained by chatting with human. The y-axis it the reward estimated by self-play evaluation. Different colors represent different language quality metrics. The solid line is obtained by simple linear regression.

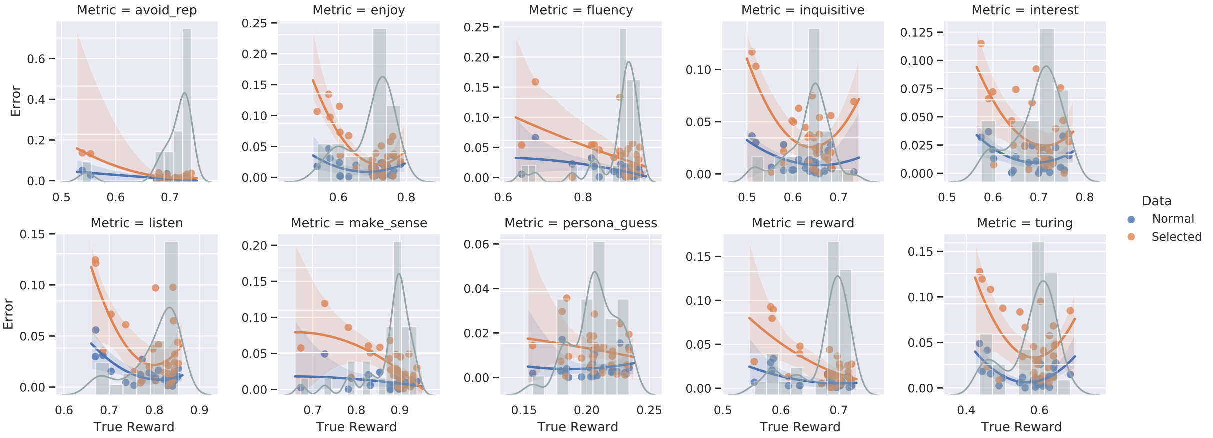
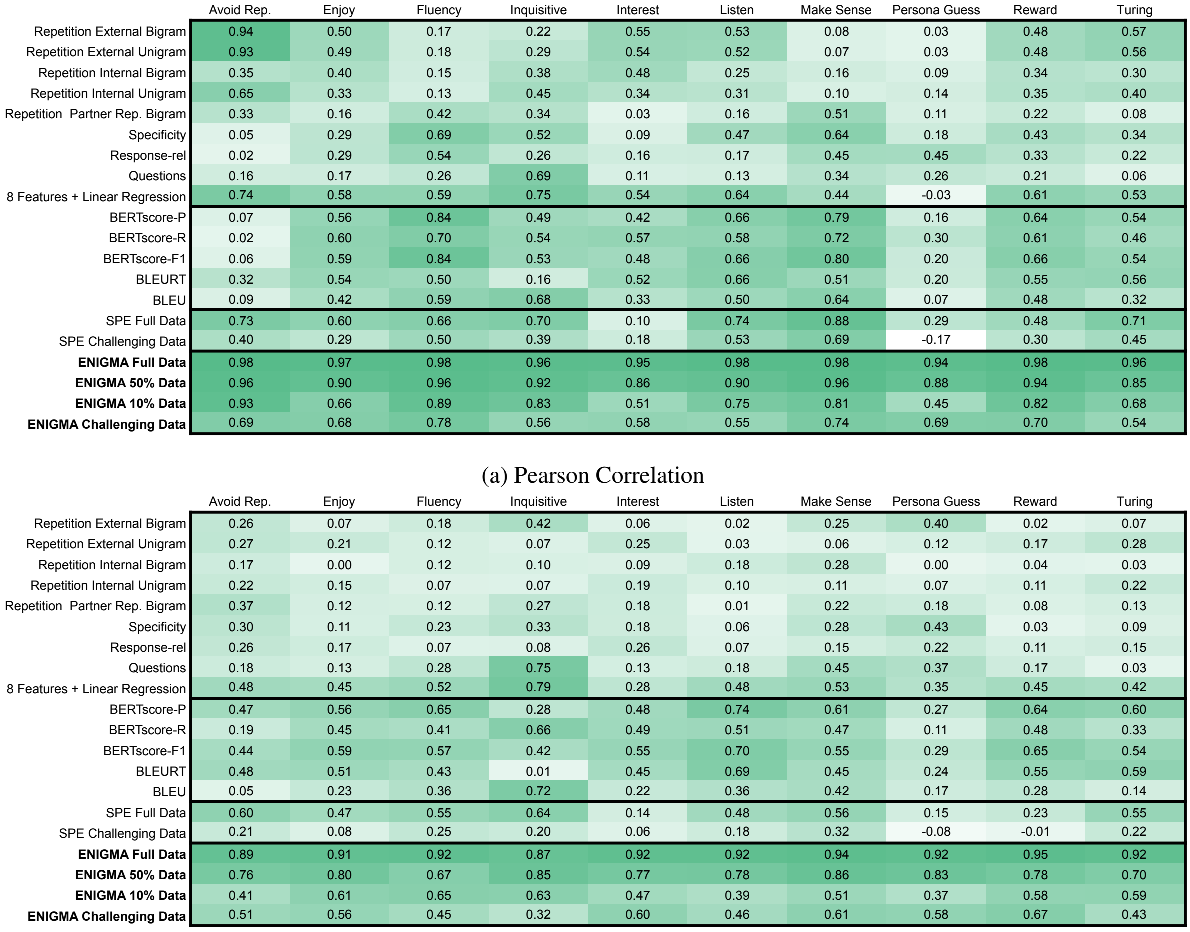


Figure 22: ENIGMA Error Comparison between using normal and selected challenging experience data on ConvAI2. The x-axis is the true average reward. The y-axis is the ENIGMA error. The solid line is the fitted quadratic function. The histogram is the empirical distribution of the rewards of all the experience data. Orange represents challenging dataset, and blue represents normal dataset.

Comparison to Automatic Hand-crafted Metrics.

We compare ENIGMA with other automatic hand-crafted metrics proposed in See et al. (2019). For a more intuitive comparison, we use heat map and box plot to visualize the correlations between different automatic evaluation metrics and different human evaluation metrics. As can be seen in Figure 23 and Figure 24, most hand-crafted metrics have relatively low correlation to human evaluation metrics. The only exception is the “question marks” automatic metrics for inquisitive human evaluation metric. Some hand-crafted metrics have high Pearson correlation to some human evaluation metrics, while the corresponding Spearman’s rank correlation is low. The reason is that they can easily identify some extremely good/bad agents while they are less effective for identifying agents with similar performance.

Comparison to BLEU, BLEURT, and BERTscore. We compare ENIGMA with other automatic single-turn language quality metrics in Figure 23: BLEU, BLEURT (Sellam et al., 2020), and BERTscore (Zhang et al., 2019). As can be seen, these metrics only have high correlation to certain human evaluation metrics and low correlation to other metrics. Note that, we do not compare the perplexity as the agents rely on complicated decoding methods (See et al., 2019) and perplexity does not take decoding into consideration.



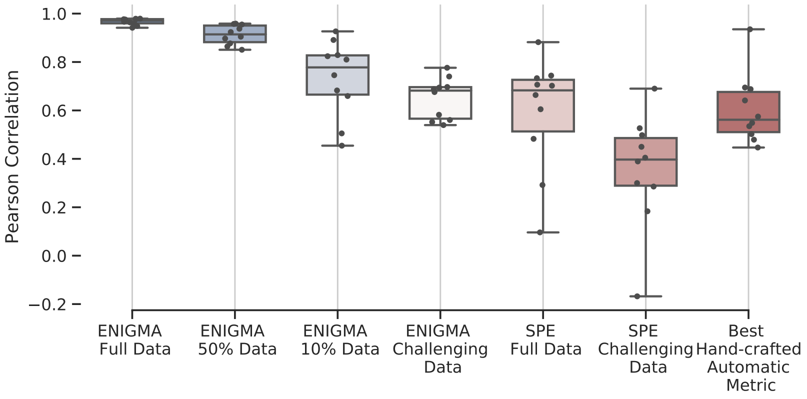
	Avoid Rep.	Enjoy	Fluency	Inquisitive	Interest	Listen	Make Sense	Persona Guess	Reward	Turing
Repetition External Bigram	0.94	0.50	0.17	0.22	0.55	0.53	0.08	0.03	0.48	0.57
Repetition External Unigram	0.93	0.49	0.18	0.29	0.54	0.52	0.07	0.03	0.48	0.56
Repetition Internal Bigram	0.35	0.40	0.15	0.38	0.48	0.25	0.16	0.09	0.34	0.30
Repetition Internal Unigram	0.65	0.33	0.13	0.45	0.34	0.31	0.10	0.14	0.35	0.40
Repetition Partner Rep. Bigram	0.33	0.16	0.42	0.34	0.03	0.16	0.51	0.11	0.22	0.08
Specificity	0.05	0.29	0.69	0.52	0.09	0.47	0.64	0.18	0.43	0.34
Response-rel	0.02	0.29	0.54	0.26	0.16	0.17	0.45	0.45	0.33	0.22
Questions	0.16	0.17	0.26	0.69	0.11	0.13	0.34	0.26	0.21	0.06
8 Features + Linear Regression	0.74	0.58	0.59	0.75	0.54	0.64	0.44	-0.03	0.61	0.53
BERTscore-P	0.07	0.56	0.84	0.49	0.42	0.66	0.79	0.16	0.64	0.54
BERTscore-R	0.02	0.60	0.70	0.54	0.57	0.58	0.72	0.30	0.61	0.46
BERTscore-F1	0.06	0.59	0.84	0.53	0.48	0.66	0.80	0.20	0.66	0.54
BLEURT	0.32	0.54	0.50	0.16	0.52	0.66	0.51	0.20	0.55	0.56
BLEU	0.09	0.42	0.59	0.68	0.33	0.50	0.64	0.07	0.48	0.32
SPE Full Data	0.73	0.60	0.66	0.70	0.10	0.74	0.88	0.29	0.48	0.71
SPE Challenging Data	0.40	0.29	0.50	0.39	0.18	0.53	0.69	-0.17	0.30	0.45
ENIGMA Full Data	0.98	0.97	0.98	0.96	0.95	0.98	0.98	0.94	0.98	0.96
ENIGMA 50% Data	0.96	0.90	0.96	0.92	0.86	0.90	0.96	0.88	0.94	0.85
ENIGMA 10% Data	0.93	0.66	0.89	0.83	0.51	0.75	0.81	0.45	0.82	0.68
ENIGMA Challenging Data	0.69	0.68	0.78	0.56	0.58	0.55	0.74	0.69	0.70	0.54

(a) Pearson Correlation

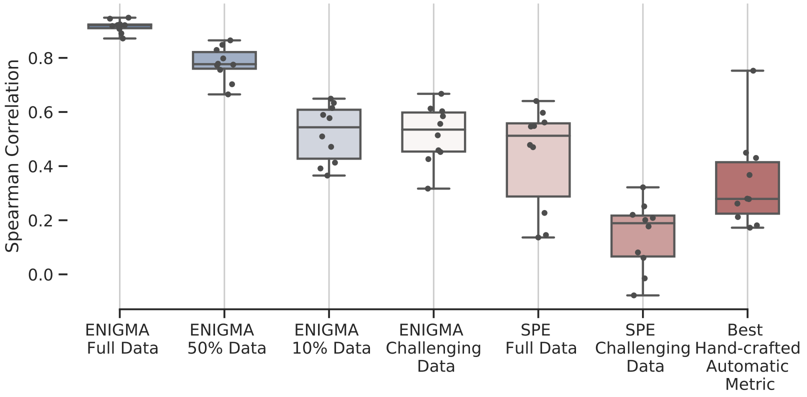
	Avoid Rep.	Enjoy	Fluency	Inquisitive	Interest	Listen	Make Sense	Persona Guess	Reward	Turing
Repetition External Bigram	0.26	0.07	0.18	0.42	0.06	0.02	0.25	0.40	0.02	0.07
Repetition External Unigram	0.27	0.21	0.12	0.07	0.25	0.03	0.06	0.12	0.17	0.28
Repetition Internal Bigram	0.17	0.00	0.12	0.10	0.09	0.18	0.28	0.00	0.04	0.03
Repetition Internal Unigram	0.22	0.15	0.07	0.07	0.19	0.10	0.11	0.07	0.11	0.22
Repetition Partner Rep. Bigram	0.37	0.12	0.12	0.27	0.18	0.01	0.22	0.18	0.08	0.13
Specificity	0.30	0.11	0.23	0.33	0.18	0.06	0.28	0.43	0.03	0.09
Response-rel	0.26	0.17	0.07	0.08	0.26	0.07	0.15	0.22	0.11	0.15
Questions	0.18	0.13	0.28	0.75	0.13	0.18	0.45	0.37	0.17	0.03
8 Features + Linear Regression	0.48	0.45	0.52	0.79	0.28	0.48	0.53	0.35	0.45	0.42
BERTscore-P	0.47	0.56	0.65	0.28	0.48	0.74	0.61	0.27	0.64	0.60
BERTscore-R	0.19	0.45	0.41	0.66	0.49	0.51	0.47	0.11	0.48	0.33
BERTscore-F1	0.44	0.59	0.57	0.42	0.55	0.70	0.55	0.29	0.65	0.54
BLEURT	0.48	0.51	0.43	0.01	0.45	0.69	0.45	0.24	0.55	0.59
BLEU	0.05	0.23	0.36	0.72	0.22	0.36	0.42	0.17	0.28	0.14
SPE Full Data	0.60	0.47	0.55	0.64	0.14	0.48	0.56	0.15	0.23	0.55
SPE Challenging Data	0.21	0.08	0.25	0.20	0.06	0.18	0.32	-0.08	-0.01	0.22
ENIGMA Full Data	0.89	0.91	0.92	0.87	0.92	0.92	0.94	0.92	0.95	0.92
ENIGMA 50% Data	0.76	0.80	0.67	0.85	0.77	0.78	0.86	0.83	0.78	0.70
ENIGMA 10% Data	0.41	0.61	0.65	0.63	0.47	0.39	0.51	0.37	0.58	0.59
ENIGMA Challenging Data	0.51	0.56	0.45	0.32	0.60	0.46	0.61	0.58	0.67	0.43

(b) Spearman's Rank Correlation

Figure 23: Heat map for correlation between different automatic evaluation metrics and different human evaluation metrics. Different rows represent different automatic metrics. Different column represent different human evaluation metrics.



(a) Pearson Correlation

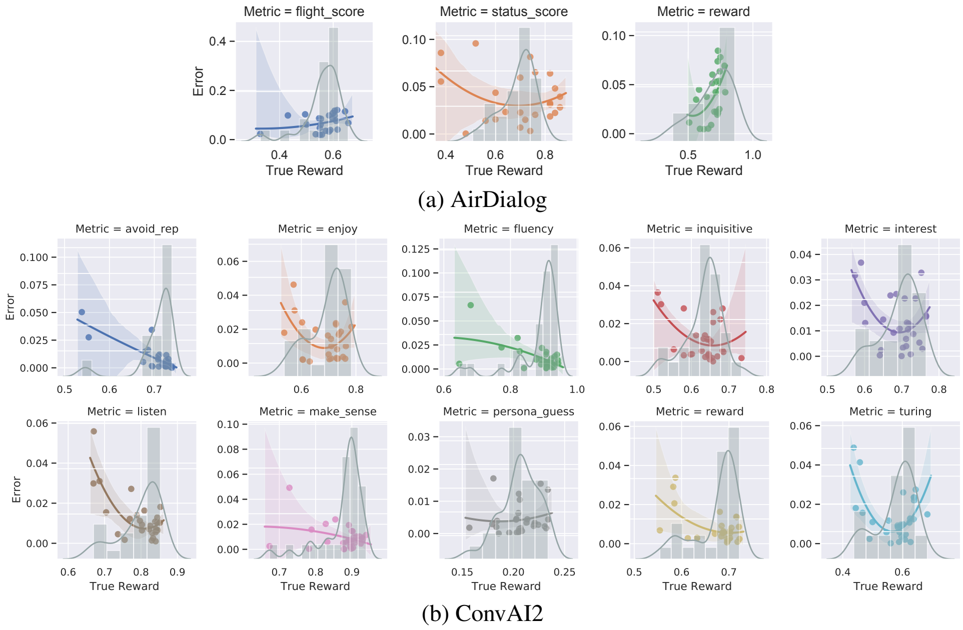


(b) Spearman's Rank Correlation

Figure 24: Box plot of performance. Each box corresponds to each method. There are 10 points for each box representing correlations to 10 different human evaluation metrics.

D.4 Error Analysis

We analyze the detailed errors to identify the error pattern for better understand the limit of ENIGMA. We calculate the absolute difference between the estimation and the true average reward. The results are summarized in Figure 25. A common pattern we see in ConvAI2 is that, when the true average reward is too high or too low, the ENIGMA becomes less accurate. One possible reason for that is the lack of samples of dialogs with the extreme rewards in the experience data. We empirically verify this conjecture by comparing the the error with the reward distribution in the experience data in Figure 25. For AirDialog, such pattern is not obvious. That is because the quality of the decision module is more important to the agent performance for this task completion scores. As a result, even performance of the target agent is much higher/lower than the experience data, as long as they share similar languages, ENIGMA can estimate the performance accurately.



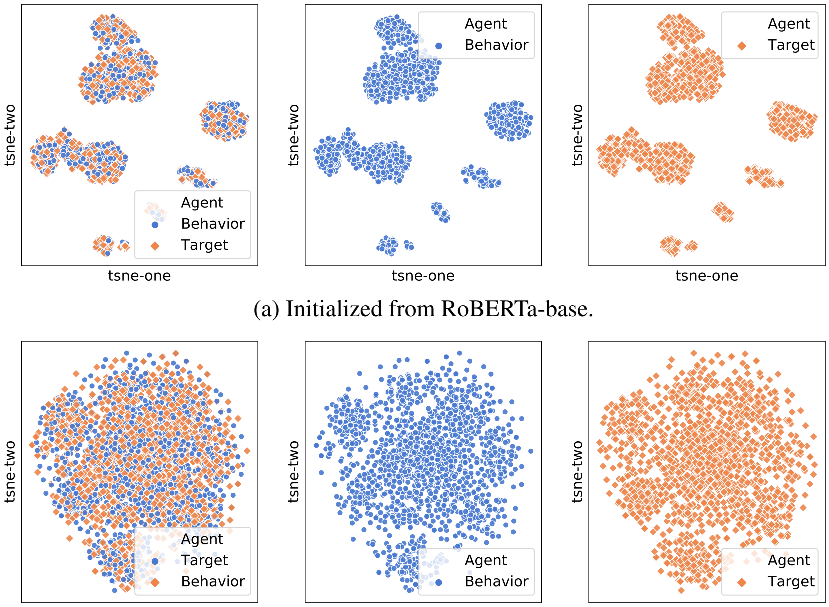
(a) AirDialog

(b) ConvAI2

Figure 25: Error Analysis on AirDialog and ConvAI2. The x-axis is the true reward. The y-axis is the Estimation error. The solid line is the fitted quadratic function. The histogram is the empirical distribution of the true rewards of all the experience data.

D.5 Embedding Visualization

In Figure 26, we present the t-SNE plots for the embedding of the state-action pairs from the behavior experience data and the target policy. The two sets of embeddings provided by the pre-trained language models are largely overlapped with rich semantic information. On the other hand, the embeddings provided by a randomly initialized model spread over the entire high-dimensional space.



(a) Initialized from RoBERTa-base.

(b) Random initialization.

Figure 26: t-SNE Plots for contextual embedding extracted from RoBERTa-ζ and RoBERTa-ν on AirDialog.

E Automatic Dialog Evaluation Comparison

Method	Criterion	Dynamic (RL)	Model Free	Experience Data	Behavior Policy Similar to Target Policy	Behavior Agnostic	Description / Examples
BLEU, Perplexity,METEOR,ROUGE (Papineni et al., 2002; Brown et al., 1992; Banerjee and Lavie, 2005; Lin, 2004; Galley et al., 2015)	Language Quality Score	No	N/A	Human-Human	Yes	N/A	The most widely use metrics: Given a **fixed** dialog history, they compute heuristic scores / statistics based on comparing **single turn** response given by the model and reference human responses. E.g., BLEU, perplexity
Mitchell and Lapata (2008); Rus and Lintean (2012); Forgues et al. (2014); Higashinaka et al. (2014); Xiang et al. (2014); Wieting et al. (2015); Gandhe and Traum (2016); Tao et al. (2017); Shimanaka et al. (2019); Zhang et al. (2019); Ghazarian et al. (2019); Li et al. (2020); Mehri and Eskenazi (2020); Gao et al. (2020); Lan et al. (2020); Pang et al. (2020); Zhang et al. (2020a); Yuma et al. (2020); Zhao et al. (2020); Sai et al. (2020)	Language Quality Score	No	N/A	Human-human experience data or specially designed data.	Yes (Implicitly) *Although they do not explicitly require such similarity, the single-turn responses of models trained from the same data are usually similar to human responses.*	N/A	Given a **fixed** dialog history, they compute some scores for **single-turn** response given by the model using **an evaluator**, e.g., pretrained word embeddings and pretrained language models. These method are the so-called "embedding-based metrics". The evaluator usually require training on a large-scale text dataset. They may or may not depends on reference human responses. E.g. RUBER (Tao et al., 2017).
Lowe et al. (2017); Huang et al. (2020); Sellam et al. (2020)	Language Quality Score	No	N/A	Human-Human and Human-Model	Yes	N/A	Mostly the same as above. In addition, the data for training the evaluator includes human-model experience data to improve performance. E.g., ADEM (Lowe et al., 2017).
Hemphill et al. (1990); Williams et al. (2013)	Task Completion Score	No	N/A	Human-Human	No	N/A	They compute task related score of task-specific actions (e.g., intent detection) given by the model for a **fixed complete** dialog. These can only be used to test classification / information retrieval module. E.g., Intent Detection Accuracy.
Wei et al. (2018)	Task Completion Score	Yes	No	Human-Human and/or Human-Model	Yes (Implicitly)	N/A	They compute task related score of task-specific actions (e.g., intent detection) given by the model for a dialog that is obtained by **interaction** with a **user simulator**. E.g., Self-Play Evaluation (Wei et al., 2018).
Ghandeharioun et al. (2019)	Language Quality Score	Yes	No	Human-Model	Yes	N/A	Basically the same as above. In addition to modeling human responses, they usually require **modeling human reward function**. E.g., Self-Play Evaluation (Ghandeharioun et al., 2019).
Inverse Proportional Score E.g., Horvitz and Thompson (1952); Wang et al. (2020a); Precup (2000) (not practical for dialog)	Both	Yes	Yes	Human-Model	Yes	No (not practical for dialog)	**Directly model the performance** under the **interaction** environment using experience collected from **known** probabilistic models. E.g., Inverse Proportional Score.
ENIGMA	Both	Yes	Yes	Human-Model	Yes	Yes	**Directly model the performance** under **interaction** environment using experience collected from **unknown** distribution. E.g., Q-Learning, ENIGMA.

Table 6: Comparison between current automatic evaluation approaches. Part of the table is collected from two comprehensive surveys (Finch and Choi, 2020; Deriu et al., 2020). **Red: Drawback**; **Green: Advantage**.

E.1 Static Methods

As can be seen, most previous methods only focus on evaluating *language quality* for **single-turn** response of a **fixed** context. These methods can not evaluate agents under interactive context. As a result, they can not be extended to *goal-oriented* dialogs.

For goal-oriented dialogs, the static evaluation methods are very limited. The static methods can only evaluate the model actions to a **fixed complete** dialog, e.g., intent detection.

Comparison to Meena Paper (Adiwardana et al., 2020): 1. They only show that PPL correlates with **one specific** metric: Sensibleness and Specificity Average. We consider a wide range of metrics for both task-completion scores and dialog quality scores (listed in Table 7). No evidence shows PPL correlates well with most metrics. 2. They draw the conclusion using **only 7** chatbots. This conclusion is not statistically reliable, i.e. for $R^2 = 0.93$ with 7 data points, the 95% confident interval is $0.64 \leq R^2 \leq 0.99$. On the other hand, we use **24/29** agents. With 24 data points, the 95% CI is $0.87 \leq R^2 \leq 0.96$, which is much more reliable.

E.2 Dynamic Methods

Previous dynamic methods under RL framework are based on self-play evluation, which requires learning the environment, i.e, human. As discussed in the main paper, learning a human model is significantly beyond the current technical limit.

ENIGMA overcome learning the environment by directly modeling the performance of agents.

E.3 Information Theoretic Limit

The common limitation of all existing methods is that they require similarity between the target policy and behavioral policies, so that the experience data can cover sufficient interaction patterns between the target policy and human.

For example, BLEU score requires the agent response being similar to the reference response. Another example is ADEM (Lowe et al., 2017), they include the target policy into the experience data collection to achieve decent performance (0.37 Pearson correlation to human ratings). If the target policy is excluded from the behavior policies, ADEM only achieves 0.13 Pearson correlation, which is even lower than the one between dialog length and human ratings 0.27.

For static single-turn evaluation for language quality, one might satisfy the requirement by just using human as the behavior policy and large-scale diverse experience data. That is because the single-turn responses of the target model have a very similar pattern to the human responses, as they are usually trained to mimic one-turn human response. However, high similarity of responses between the target model and human requires a very strong target model trained with large-scale data, which is not practical in most settings. Some existing work try to alleviate such requirement and increase the coverage of experience data by external knowledge graph (Huang et al., 2020) and synthetic samples (Sellam et al., 2020). We remark that although the static methods only require single-turn similarity between behavior and target policies, their empirical performance is unsatisfactory comparing with multi-turn interactive human evaluation (Ghandeharioun et al., 2019).

In multi-turn interactive evaluation, we can not just use human as the behavior policy especially for goal-oriented dialogs. That is because the multi-turn behavior of the target model is very different from the human behavior. Take Airdialog as an example, human agents can always book the correct tickets while the target model may fail for many times.

Such a limitation is the theoretical requirement of bounded state-action density ratio between target and behavior policies, which has been discussed in many off-policy evaluation literature (Wang et al., 2020b; Xie et al., 2019).

Due to such theoretical limitation, a large amount of **human-model** interactive evaluation data is needed to study automatic interactive evaluation. However, most evaluation logs are not publicly available, and research in this direction has largely lagged behind. To the best of our knowledge, ConvAI2 (See et al., 2019) is the only public comprehensive human-model interactive evaluation data. [7] Therefore, we recommend that the research community release human-model interaction evaluation data to promote dialog evaluation/learning research and benefit the entire community.

[7] Our human-model evaluation data on Airdialog will also be released.