TA&AT: Enhancing Task-Oriented Dialog with Turn-Level Auxiliary Tasks and Action-Tree Based Scheduled Sampling

Longxiang Liu[1,2], **Xiuxing Li**[1,2], **Yang Feng**[1,2,*]

[1]Key Laboratory of Intelligent Information Processing,
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
[2]University of Chinese Academy of Sciences, Beijing, China
{liulongxiang21s, lixiuxing, fengyang}@ict.ac.cn

Abstract

Task-oriented dialog systems have witnessed substantial progress due to conversational pre-training techniques. Yet, two significant challenges persist. First, most systems primarily utilize the latest turn's state label for the generator. This practice overlooks the comprehensive value of state labels in boosting the model's understanding for future generations. Second, an overreliance on generated policy often leads to error accumulation, resulting in suboptimal responses when adhering to incorrect actions. To combat these challenges, we propose turn-level multi-task objectives for the encoder. With the guidance of essential information from labeled intermediate states, we establish a more robust representation for both understanding and generation. For the decoder, we introduce an action tree-based scheduled sampling technique. Specifically, we model the hierarchical policy as trees and utilize the similarity between trees to sample negative policy based on scheduled sampling, hoping the model to generate invariant responses under perturbations. This method simulates potential pitfalls by sampling similar negative policy, bridging the gap between task-oriented dialog training and inference. Among methods without continual pre-training, our approach achieved state-of-the-art (SOTA) performance on the MultiWOZ dataset series and was also competitive with pre-trained SOTA methods.

1 Introduction

The goal of task-oriented dialog (TOD) is to better accomplish a user-specific task through multi-turn dialog. As shown in Figure 1, a typical TOD system consists of four modules: (1) natural language understanding (NLU) to determine the user intent. (2) dialog state tracking (DST) to extract the user constraints which will be used to query the database (DB). (3) policy (POL) to plan for the system's next action sequence. (4) natural language generation (NLG) to generate a fluent and informative response. In recent works, NLU is usually not handled specifically, but put into DST module (Takanobu et al. 2020). End-to-end task-oriented dialog, which is also the focus of our work, integrates submodules into one model for joint training.

Performance of end-to-end TOD systems has improved dramatically in recent years thanks to powerful pre-trained language models, especially dialog pre-training. However, two issues still exist. **Firstly**, there are some datasets with intermediate state annotations, which most works simply use to supervise the generator. While we believe that the annotations are not fully utilized and their essential value for understanding is overlooked. **Secondly**, the sequence-to-sequence (Seq2Seq) training approach leads to error accumulation, especially generating unsatisfying responses that are attached to incorrect actions.

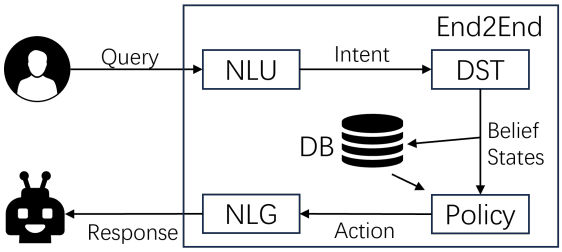


Figure 1: Illustration of task-oriented dialog system.

To solve the first problem, we utilize the labels of intermediate states to supervise the hidden states output by encoder, hoping better representations provide useful clues for the subsequent generation. Inspired by MTTOD (Lee 2021), which utilizes the belief state annotations to construct a context-span labeling auxiliary task, we leverage more annotations to construct more **auxiliary tasks** (e.g., slot type, slot change, action type, and response keywords prediction). Besides, inspired by DialoFlow (Li et al. 2021), we optimize the **turn-level** representation instead of token-level since it reflects higher-level information such as conversational goal or potential influence before generation of next response.

To solve the second problem, we attempt to use **scheduled sampling** technique (Bengio et al. 2015) to reduce the inconsistency between training and inference. However, in TOD system, simply using the token-level scheduled sampling does not actually simulate the errors at inference. Given a specific token, the likelihood of generating a subsequent token is highly deterministic due to the strong conditional relationships between tokens. This results in a sharp token-level conditional probability distribution, making it challenging for a single negative token to be sampled. In-

*Corresponding author.

stead, there is more uncertainty among action sequences. Therefore, we propose a method that can directly sample a negative action sequence similar to the ground truth action at the training time, called **action-tree based scheduled sampling**. Specifically, inspired by SPACE (He et al. 2022a), we model the action sequence as a tree, calculate the similarity according to the edit distance between action trees, and then use similarity as the sampling distribution of negative action sequences. We optimize the likelihood of reference response under the perturbation of action sequence.

We have conducted comprehensive experiments on MultiWOZ 2.0/2.1/2.2. Experiments show that our method **TA&AT** substantially improves TOD system and achieves new state-of-the-art results among methods that **do not adopt continual pre-training**, pushing the end-to-end combined score on MultiWOZ 2.0/2.1/2.2 to **109.27/108.03/103.59**. Ablation study also verifies the effectiveness of our proposed method.

In summary, our main contributions are three-fold:

- We explore how to make the most of intermediate annotations in TOD system, through turn-level auxiliary tasks.
- To the best of our knowledge, this is the first attempt to introduce sequence-level scheduled sampling into TOD.
- Extensive experiments show our method achieves state-of-the-art performance on MultiWOZ datasets.

2 Related Work

End-to-end task-oriented dialog aims at jointly training submodules and building a text-in, text-out integrated system. (Wen et al. 2016) first proposed a trainable neural network-based framework for end-to-end TOD, using CNN (Kalchbrenner, Grefenstette, and Blunsom 2014) and LSTM (Hochreiter and Schmidhuber 1997) in different modules. (Lei et al. 2018; Zhang et al. 2020; Zhang, Ou, and Yu 2020) proposed their methods mainly based on CopyNet (Gu et al. 2016) in seq2seq training and elaborate design of decoder.

Due to the blooming of pre-trained language models (PLMs), recent approaches employ PLM as their backbone such as GPT (Radford et al. 2018), T5 (Raffel et al. 2020) and UniLM (Dong et al. 2019). (Kulhánek et al. 2021; Peng et al. 2021; Yang, Li, and Quan 2021; Hosseini-Asl et al. 2020) applied GPT-2 model for different modules, training in turn-level or session-level. Since there are not only generation tasks but also language understanding tasks in TOD, encoder-decoder framework fits better. There are many works that use T5 as a base model and promote end-to-end performance from their own perspectives. Among them, (Su et al. 2021; Lee 2021; Bang, Lee, and Koo 2023) utilize multi-task learning, (Sun et al. 2023) leverages contrastive learning to model the relationship between dialog context and belief/action state representations. There are also works based on parameter-shared encoder-decoder UniLM, (He et al. 2022b,a) continually pre-train their proposed semi-supervised or self-supervised learning tasks on UniLM and then adapt to downstream tasks through finetuning, which achieves current state-of-the-art.

To mitigate error accumulation in end-to-end TOD, (Zhang, Ou, and Yu 2020) takes different valid dialog policies into consideration to learn a balanced action distribution, guiding the dialog model to generate diverse responses. (Sun et al. 2022) introduced a back and denoising reconstruction approach and (He et al. 2022b) employed consistency regularization to refine the learned representation. Different from above works, we attempted to apply scheduled sampling, which is proposed in sequence generation task (Bengio et al. 2015) and improved in neural machine translation (Zhang et al. 2019).

3 Model Framework

In this section, we will introduce our model framework. As described in Section 1, end-to-end task-oriented dialog generation is usually modeled as a cascading generation problem. In each turn, the system receives the user's input, which will be concatenated with the context information in the memory block. Then the belief states should be generated, which is a hierarchical semantic state reflecting the constraints of user requests. The belief states are used to query the database, whose matching results will be used together with the context information to determine a policy. Policy is a hierarchical action sequence, guiding the process of response generation. In general, belief states contain *(domain, slot, value)* and policy contains *(domain, action, slot)*, both of which are three-level structures.

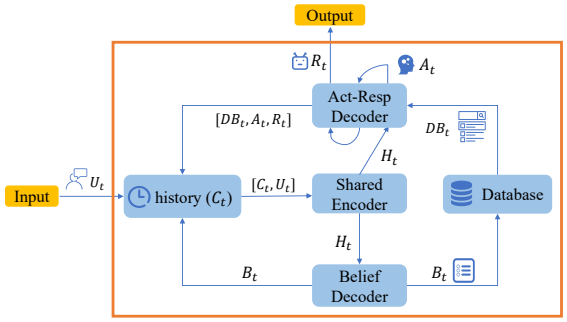
There are several choices for the base model framework, like decoder-only GPT (Yang, Li, and Quan 2021; Peng et al. 2021), encoder-decoder T5 (Su et al. 2021; Bang, Lee, and Koo 2023), UniLM-based models (He et al. 2022b,a), encoder-2decoders based models (Lee 2021; Cholakov and Kolev 2022). Considering that the belief generation depends more on understanding and summarization ability, while the policy and response generation relies more on generative ability to maintain contextual coherence. We believe that they belong to different semantic subspace, and in our experiments UniLM requires time-consuming pre-training to show good performance, which is also verified in (He et al. 2022b). We finally adopt the framework proposed in (Lee 2021), containing one shared encoder and two different decoders, as shown in Figure 2.

3.1 Definitions

Here we introduce the symbols involved according to the input stream. In the t-th turn of a dialog, U_t represents the user input utterance. B_t is the belief state, which in the Figure 2 is {*restaurant:*{*pricerange:expensive, area:centre, food:Chinese*}}. DB_t represents the database result, reflecting the matching number of entities satisfying the belief states. A_t represents the action sequence, which in the Figure 2 is {*restaurant:*{*inform:[address,name], offerbook:[]*}}. R_t represents the system response. The context information $I_t = (U_t, B_t, DB_t, A_t, R_t)$ will be gathered in the memory block. Note that inspired by (Yang, Li, and Quan 2021), we concatenate all the history information, where $C_t = \text{Concat}(I_0, ...I_{t-1})$.

3.2 Objectives

In the end-to-end task-oriented dialog framework, the context information in memory module and current user utter-



C_t: Can you tell me about any expensive restaurants in the centre? [restaurant] pricerange expensive area centre [db_3] [restaurant] [inform] price choice area [request] food We have [value_choice] [value_pricerange] restaurants in the [value_area], do you have a specific cuisine in mind?
U_t: Yes, I would prefer Chinese please.
B_t: [restaurant] pricerange expensive area centre food Chinese
DB_t: [db_3]
A_t: [restaurant] [inform] address name [offerbook]
R_t: I have the [value_name] located at [value_address]. Would you like to make reservations?

Figure 2: Illustration of our task-oriented dialog system framework. For simplicity, we show an example dialog in the scenario of a user ordering a restaurant, $t = 1$ (starts from 0). The memory module will keep track of the new generated belief states, db states, acts, and responses.

ance will be input to a shared transformer encoder to get the hidden states H_t. Then H_t is first input to the belief decoder to generate belief states. The generated belief states B_t will be used to query the database, returning DB_t. Finally, H_t and DB_t are input together to the Action-Response Decoder to autoregressively generate the action A_t and response R_t.

$$\begin{aligned} H_t &= \text{Encoder}([C_t, U_t]) \\ B_t &= \text{Decoder}_b(H_t) \\ A_t, R_t &= \text{Decoder}_{ar}(H_t, DB_t) \end{aligned} \tag{1}$$

Both decoders and the encoder are optimized with cross entropy loss supervised by the teacher-forcing ground truth belief states, action, and response.

$$\begin{aligned} \mathcal{L}_B &= -\log P(\hat{B}_t | H_t) \\ \mathcal{L}_{AR} &= -\log P(\hat{A}_t, \hat{R}_t | H_t, DB_t) \\ \mathcal{L} &= \mathcal{L}_B + \mathcal{L}_{AR} \end{aligned}$$

4 Methodology

In this section, we elaborate on our proposed method. In order to relieve the problem of **insufficient** utilization of labels we described in Section 1, we propose four turn-level auxiliary tasks to enhance the understanding ability of encoder, providing some inherent clues for subsequent generation; To alleviate the problem of **sequence-level error accumulation**, we propose action-tree based scheduled sampling, making response generation more robust. We will first describe the auxiliary tasks, then describe the action-tree based scheduled sampling approach, and discuss the training and inference process at last. Our proposed method is shown in Figure 3.

4.1 Turn-Level Auxiliary Tasks

There are many annotations other than ground truth responses, which can be used to strengthen the understanding of the encoder. Inspired by MTTOD (Lee 2021), which leverages the annotations of belief states to introduce a simple span prediction task for task-oriented dialog enhancement, we propose to leverage more types of annotations and introduce additional auxiliary tasks. Besides, as stated in DialoFlow (Li et al. 2021), the turn-level representations reflect higher-level information such as the conversational goal or potential influence before generation of next responses. Based on the above two points, we supervise turn-level representation learning by using high-level supervision signals from different types of annotations (e.g., belief states, actions, responses). We hope that these turn-level representations can better provide clues for subsequent generation. Four below auxiliary tasks are proposed, and each corresponding **true** label set in the example of Figure 2 is given behind. For simplicity, domain labels are ignored.

- Slot type: *[pricerange, area, food]*
- Slot transition: {*pricerange:keep, area:keep, food:new*}
- Action type: *[inform, offerbook]*
- Response keywords: *([value_name],[value_address])*

Now we describe these four tasks in detail.

Turn representation Both our model encoder and two decoders are initialized using T5's corresponding modules, which is not the focus of this paper, so the introduction of T5's model structure is ignored. According to Equation 1, H_t is the encoder output hidden states, we use the end position of each turn to select the turn-level representations from turn-0 to turn-$(t-1)$, denoted as T_t.

$$\begin{aligned} P_t^{end} &= [pos_0, pos_1, ..., pos_{t-1}] \in \mathbb{N}^t \\ T_t &= \text{IndexSelect}(H_t, P_t^{end}) \in \mathbb{R}^{d \times t} \end{aligned}$$

Slot Type Prediction Determining which slots are mentioned in a user's utterance can help to generate belief states since it narrows the scope of slot-value pairs, and such discriminative task is better suited to the capabilities of the encoder. There are some turns associated with multiple types of slots mentioned. Following GALAXY (He et al. 2022b), we model the slot type prediction task as a multi-label classification problem. In Equation 2, we denote $ST = (st_1, st_2, ..., st_N)$, where N is the total number of slot types. A multi-dimensional Bernoulli distribution is used for modeling the slot types. The turn representation T will be passed through a multi-dimensional binary classifiers to get the prediction score of each slot type.

$$\begin{aligned} p(ST|T) &= \prod_i^N p(st_i|T) \\ p(st_i|T) &= \text{sigmoid}(W_{st}T) \in \mathbb{R}^N \\ \mathcal{L}_{st} &= -\sum_{i=1}^{N} \{y_i \log p(st_i|T) + (1-y_i) \log(1 - p(st_i|T))\} \end{aligned} \tag{2}$$

where W_{st} is trainable parameter matrix of linear slot type head and $y_i \in \{0, 1\}$ is the label of whether st_i appears in current turn.

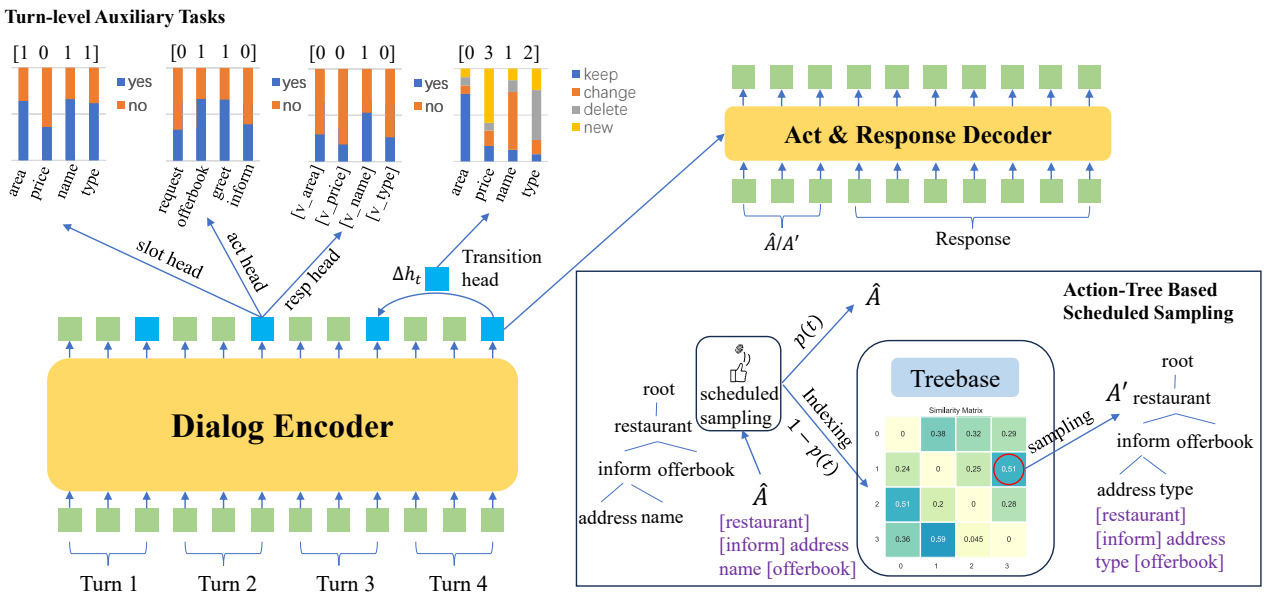


Figure 3: Overall framework of our proposed methods. The left part shows the process of extracting turn-level representations and passing them to four multi-dimensional Bernoulli/Categorical classification heads. The right part shows the process of action-tree based scheduled sampling, where the ground truth action $\hat{A}$ will be replaced with the probability of $1 - p(t)$, a replacing action sample A' is then sampled according to the normalized similarity score. The calculation of similarity score is based on the action-tree Editing Distance, which will be discussed detailedly in Section 4.2.

Slot Change Prediction SOM-DST (Kim et al. 2019) has mentioned that state operation prediction allows state tracking model to efficiently generate the values of only a minimal subset of the slots. In our situation, predicting the slot change also provides important clues for belief state generation. We define slot change into four categories: {keep, change, delete, new}, which is similar to the operations in database system. Given two consecutive turns' belief states, the slot change between them is easy to get. Here a multi-dimensional categorical distribution is adopted for modeling the slot change, as shown in Equation 3, we denote $SC = (sc_1, sc_2, ..., sc_N)$. $\Delta T = T_t - T_{t-1}$, which reflects the difference between adjacent turns' representations, is fed to the trainable transition head W_{sc}. Note that for simplicity, the subscript of T_t is omitted when there is no ambiguity.

$$\begin{aligned} \Delta T_t &= T_t - T_{t-1} \\ p(SC|\Delta T) &= \prod_{i}^{|SC|} p(sc_i^{y_i}|\Delta T) \\ p(sc_i|\Delta T) &= \text{Softmax}(W_{sc}\Delta T) \in \mathbb{R}^4 \\ \mathcal{L}_{sc} &= -\sum_{i=1}^{N} \log p(sc_i^{y_i}|\Delta T) \end{aligned} \tag{3}$$

where $y_i \in \{0, 1, 2, 3\}$ is the label of i-th slot change.

Action Prediction As described in GALAXY (He et al. 2022b), identifying the actions (e.g. request, offerbook, etc.) can facilitate learning better representations for policy optimization to improve the overall end-to-end performance. Here we adopt the same way in GALAXY using multi-dimensional Bernoulli distribution to model action prediction. The difference is that we predict all turns' actions while GALAXY only predicts current turn's.

$$\begin{aligned} p(A|T) &= \prod_{i}^{N} p(a_i|T) \\ p(a_i|T) &= \text{sigmoid}(W_a T) \in \mathbb{R}^N \\ \mathcal{L}_a &= -\sum_{i=1}^{N} \{y_i \log p(a_i|T) + (1 - y_i) \log(1 - p(a_i|T))\} \end{aligned} \tag{4}$$

where W_a is trainable parameter matrix of linear action head and $y_i \in \{0, 1\}$ is the label of whether a_i is taken in current action.

Response Keywords Prediction In most situations in task-oriented dialog, the system should inform some essential values in the response according to what the user requests, which also relates to the evaluation metric **Success** in Section 5.1. Predicting such keywords can make the model focus on essential information to be generated, such as *[value_name], [value_area]*, etc. In delexicalized responses (Zhang, Ou, and Yu 2020), words like *[value_xxx]* are in a finite set. Here we model the bag-of-words predictions as a multi-dimensinal Bernoulli distribution, as shown in Equation 5.

$$\begin{aligned} p(K|T) &= \prod_{i}^{N} p(k_i|T) \\ p(k_i|T) &= \text{sigmoid}(W_k T) \in \mathbb{R}^N \\ \mathcal{L}_k &= -\sum_{i=1}^{N} \{y_i \log p(k_i|T) + (1 - y_i) \log(1 - p(k_i|T))\} \end{aligned} \tag{5}$$

where N is the vocabulary size of keywords, W_k is trainable parameter matrix of linear response head and $y_i \in$

$\{0, 1\}$ is the label of whether keyword k_i is appeared in the response.

To be summarized, total loss for the turn-level auxiliary tasks is

$$\mathcal{L}_{TA} = \mathcal{L}_{st} + \mathcal{L}_{sc} + \mathcal{L}_a + \mathcal{L}_k$$

4.2 Action-Tree Based Scheduled Sampling

In our experiments, we found that the more we train, the more faithful the generated responses are to the generated action sequence. However, this can lead to unsatisfying responses, especially when the generated actions are not reasonable enough. This phenomenon is called error accumulation, which is caused by exposure bias. Since the seq2seq training process is teacher-forced, it is inconsistent with the inference phase (Zhang et al. 2019). Scheduled sampling (Bengio et al. 2015) is a straightforward way to mitigate this problem, which randomly replaces target-side input tokens with model predictions following a curriculum learning strategy.

However, directly adopting token-level scheduled sampling is not effective for our task, because the main error at inference exists in the action sequence bringing inaccurate response sequence, so sequence-level replacement is needed. To this end, we propose an action-tree based scheduled sampling method. Next we describe the method in detail.

Action Tree Inspired by SPACE (He et al. 2022a), we calculate the similarity score among action sequences and save the similarity matrix. When calculating the similarity score, we first convert the action sequence to a hierarchical action tree, containing the tertiary structure (domain,action,slot) from top to bottom, as shown in the right part of Figure 3. Then we derive the Tree Editing Distance (Zhang and Shasha 1989), which is the weighted number of edit operations (insert, delete, and modify) to transform one tree to another. Note that we use **ordered tree**, which is different from SPACE, since in our experiments we found the relative position of actions will affect the response generation. Besides, reordering to an unordered tree may result in some not existing action sequences. Denoting the semantic trees of i-th action and j-th action are T_i and T_j. Tree Editing Distance is calculated, and then similarity score $s_{i,j}$ between i-th action and j-th action is calculated by Equation 6.

$$\begin{aligned} s_{i,j} &= \frac{\max\{|T_i|, |T_j|\} - d_{i,j}}{\max\{|T_i|, |T_j|\}} \\ d_{i,j} &= \text{TreeEditingDistance}(T_i, T_j) \end{aligned} \tag{6}$$

Scheduled Sampling As shown in the bottom right of Figure 3, in the training process, before one ground truth action $\hat{A}_t$ is input to the act-response decoder, it will be retained with probability $p(t)$, which is calculated by Equation 7 (Zhang et al. 2019).

$$p = \frac{\mu}{\mu + \exp(t/\mu)} \tag{7}$$

where μ is a hyper-parameter. And the function is strictly monotone decreasing. Otherwise, the ground truth action will be used to index the similarity matrix, assuming the indexed column is i. We denote the similarity matrix as M, then the sampling distribution is

$$p_j^* = \frac{M[i,j]}{\sum_{j=1, j\neq i}^{N} M[i,j]}$$

note that here we guarantee that the same i-th action will not be sampled.

Loss As shown in Equation 8, when the perturbed action is adopted as input, the action loss should not be optimized but the response loss should still be optimized to improve the robustness of response generation in the presence of noisy actions. In such situation, model should learn to depend more on the context when generating the response.

$$\begin{aligned} \mathcal{L}_A &= -\log P(A_t | H_t, DB_t) \\ \mathcal{L}_R &= -\log P(R_t | H_t, DB_t, A_t) \\ \mathcal{L}_{AT} &= \begin{cases} \mathcal{L}_A + \mathcal{L}_R, & A_t = \hat{A}_t \\ \mathcal{L}_R, & A_t = A'_t \end{cases} \end{aligned} \tag{8}$$

4.3 Traning and Inference

The final loss in our training process is described by Equation 9.

$$\mathcal{L} = \mathcal{L}_{TA} + \mathcal{L}_B + \mathcal{L}_{AT} \tag{9}$$

Note that since the belief decoder is not our focus in this work, we did not discuss this module and omit it in Figure 3, but the loss $\mathcal{L}_B$ always exists.

In the inference phase, we only utilize the shared encoder and two decoders, and neither the classification head nor scheduled sampling is required, making the overall inference cost completely unchanged with respect to our backbone.

5 Experiments

In this section, we will introduce experimental data, metrics, compared baselines, and our results in different tasks. Our code is released in our github repository[1].

5.1 Datasets and Evaluation Metrics

Datasets We evaluate end-to-end dialog system performance of our proposed methods on public task-oriented dialog benchmark MultiWOZ (Budzianowski et al. 2018). We evaluate our method on MultiWOZ 2.0, 2.1 and 2.2. Following the data split in (Lee 2021), the number of train/-validation/test set is 8438/1000/1000. And to reduce diversity of the surface form, we replace some specific slot values with *[value_xxx]* to construct the delexicalized response, allowing the model to learn value-independent parameters (Zhang, Ou, and Yu 2020).

Metrics We follow the automatic evaluation metrics to evaluate the response quality for task-oriented dialog system on MultiWOZ datasets. **Inform rate** measures whether a dialog system has provided an accurate entity; **Success rate** measures whether a dialog system has answered all requested information; **BLEU** is computed with references, measuring the fluency of the generated response. **Combined score =** $(\text{Inform} + \text{Success}) \times 0.5 + \text{BLEU}$, reflects the overall quality of the dialog system, which is our main metric.

[1] https://github.com/ictnlp/TA-AT

Model	MultiWOZ 2.0				MultiWOZ 2.1				MultiWOZ 2.2			
	Inform	Success	BLEU	Comb	Inform	Success	BLEU	Comb	Inform	Success	BLEU	Comb
w.o. continual pre-training												
SimpleTOD	84.40	70.10	15.01	92.26	85.00	70.50	15.23	92.98	-	-	-	-
DoTS	86.59	74.14	15.06	95.43	86.65	74.18	15.90	96.32	80.40	68.70	16.80	91.40
SOLOIST	85.50	72.90	16.54	95.74	-	-	-	-	82.30	72.40	13.60	90.9
MinTL	84.88	74.91	17.89	97.79	-	-	-	-	73.70	65.40	19.40	89.00
UBAR	**95.40**	80.70	17.00	105.05	**95.70**	81.80	16.50	105.25	83.40	70.30	17.60	94.40
GALAXY	93.10	81.00	18.44	105.49	93.50	81.70	18.32	105.92	85.40	75.70	19.64	100.20
BORT	93.80	**85.80**	18.50	108.30	-	-	-	-	85.50	77.40	17.90	99.40
Mars	-	-	-	-	-	-	-	-	**89.20**	**80.30**	19.00	103.40
MTTOD	90.99	82.58	20.25	107.04	90.99	82.08	19.68	106.22	85.90	76.50	19.00	100.20
TA&AT	93.60	83.60	**20.67**	**109.27**	92.50	**84.00**	**19.78**	**108.03**	86.40	80.10	**20.34**	**103.59**
w. continual pre-training												
PPTOD*	89.20	79.40	18.62	102.92	87.09	79.08	19.17	102.26	83.10	72.70	18.20	96.10
GALAXY*	94.40	85.30	20.50	110.35	95.30	86.20	20.01	110.76	-	-	-	-
SPACE*	95.30	88.00	19.30	110.95	95.60	86.10	19.91	110.76	-	-	-	-

Table 1: E2E performances on MultiWOZ 2.0/2.1/2.2. TA&AT is our method, short for Turn-level Auxiliary tasks and Action-Tree based scheduled sampling. All results are from original papers or public MultiWOZ leaderboard. '*' means using continual training on extra datasets.

Model	10% data				20% data				50% data			
	Inform	Success	BLEU	Comb	Inform	Success	BLEU	Comb	Inform	Success	BLEU	Comb
MinTL	55.5	44.9	15.6	65.8	64.3	54.9	16.2	75.8	70.3	62.2	18.0	84.3
PPTOD	68.3	53.7	15.7	76.7	72.7	59.2	16.3	82.3	74.8	62.4	17.0	85.6
UBAR	50.3	34.2	13.5	55.8	65.5	48.7	14.5	71.6	77.6	63.3	16.3	86.8
MTTOD	66.9	55.2	13.8	74.9	75.0	63.3	14.3	83.5	78.5	67.5	15.2	88.2
Mars	69.4	55.3	15.6	78.0	76.7	62.9	**17.2**	87.0	82.2	71.2	**18.6**	95.3
TA&AT	**71.5**	**58.4**	**16.2**	**81.1**	**79.2**	**68.2**	16.8	**90.5**	**83.5**	**73.8**	18.1	**96.8**

Table 2: E2E results of low-resource experiments. 10% (800 dialogs), 20% (1600 dialogs), 50% (4000 dialogs) of training data is used to train our model. All of the results are cited from Mars (Sun et al. 2023).

5.2 Settings

Following (Lee 2021), we use a pre-trained T5-base model (Raffel et al. 2020) to initialize our shared encoder and two decoders. We implement our methods based on the HuggingFace Transformers library (Wolf et al. 2020). We train our model for 10 epochs on a single 40G NVIDIA A100. Our model is trained for approximately 10 hours. In low resource setting, our model is trained for 20 epochs. The initial learning rate is set to 5e-4, batch size is set to 8 and the proportion of warmup steps is set to 0.1. We adopt an optimizer as AdamW (Loshchilov and Hutter 2017) with linear learning rate decay. We select the best model based on the performance on the validation set. For the hyperparameter μ, we choose most suitable one from $\{10,15,20\}$ for different datasets. To remove randomness, we fix our random seed to 42 in our experiments. A simple greedy search algorithm is used when decoding belief states, action, and responses.

5.3 Baselines

For a fair comparison, we confine our analysis to those methods that utilize PLMs. And the methods using PLMs typically fall under two distinct settings:

- **Without Continual**: Directly fine-tuning the PLM for specific downstream tasks, such as end-to-end modeling.
- **With Continual**: Beginning with continual pre-training on extra datasets and then transitioning to fine-tuning.

We will compare our method with those in the **Without Continual** setting to underline the strengths of our approach. Additionally, comparisons with the methods in **With Continual** setting will be conducted to clearly illustrate the extent of the gap between our method and them.

We compared serveral strong baselines, including SimpleTOD (Hosseini-Asl et al. 2020), DoTS (Jeon and Lee 2021), SOLOIST (Peng et al. 2021), MinTL (Lin et al. 2020), PPTOD (Su et al. 2021), UBAR (Yang, Li, and Quan 2021), GALAXY (He et al. 2022b), MTTOD (Lee 2021), BORT (Sun et al. 2022), Mars (Sun et al. 2023) and SPACE (He et al. 2022a).

5.4 Main Results

As shown in Table 1, our method TA&AT achieves new state-of-the-art combined scores on all the datasets in **w.o. Continual** setting. Even compared with the SOTA SPACE model, performance of our method is comparable, indicating that our proposed methods are competitive for end-to-end task-oriented dialog modeling. Note that based on MTTOD, our method can improve its performance on MultiWOZ 2.0 by 2.23 points (from 107.04 to 109.27), MultiWOZ 2.1 by 1.81 points (from 106.22 to 108.03), MultiWOZ 2.2 by 3.39 points (from 100.2 to 103.59). Note that our model acheives best **BLEU** in each dataset while keeping other metrics at a high-level, verifying the effectiveness of our method in im-

Model	Inform	Success	BLEU	Comb
TA&AT	93.60	83.60	**20.67**	109.27
- $\mathcal{L}_{st}$	93.10	84.50	19.79	108.59 (-0.68)
- $\mathcal{L}_{sc}$	92.60	**84.50**	20.28	108.83 (-0.44)
- $\mathcal{L}_{a}$	93.50	84.40	20.05	109.00 (-0.27)
- $\mathcal{L}_{k}$	**93.70**	84.40	20.29	**109.34** (+0.07)
w.o. AT	93.40	83.50	19.72	108.17 (-1.10)
w.o. TA	92.90	83.30	19.76	107.86 (-1.41)
MTTOD	90.99	82.58	20.25	107.04 (-2.23)

Table 3: Ablation study on E2E results of MultiWOZ 2 'w.o. AT' means normal teacher-forcing without any acti tree based scheduled sampling.

proving the generation quality.

5.5 Low-Resource Evaluation

In order to explore whether our method is equally effective in low-resource scenarios, following the setting of Mars, we tested the performance of the model with 10%, 20%, and 50% number of training sessions, respectively. As shown in Table 2 our method achieves the best in most of data ratio, demonstrating its robustness.

6 Analysis

In this section, we first analyze the effectiveness of each auxiliary task and scheduled sampling. Then we discuss some observations from the learning curve in our training process.

6.1 Ablation Study

As shown in Table 3, most auxiliary tasks are effective especially those slot-related ones. Interestingly, we found a phenomenon that removing the response keywords prediction task results in a higher combined score. It seems that this task does not work. We attribute this to the fact that different losses have different learning periods, that is, the learning of $\mathcal{L}_k$ may still be underfitted when the $\mathcal{L}_{sc/st}$ are already overfit (see Section 6.2), finding an optimal balanced point or determining the best ratio may be a future direction. Besides, such a small increase (+0.07) can also be due to randomness.

In a word, both AT and TA are effective, and the latter is more important, because because it provides a better encoder representation, influencing both understanding and generation. Compared to most similar baseline MTTOD, our method outperforms it by 2.23, verifying the effectiveness of TA&AT.

6.2 Learning Curve

The variation of F1-score corresponding to different tasks during training is shown in Figure 4. It can be found that at the very early stage of training, the F1 value is almost unchanged, indicating that the generation loss is primal at this time. In the process of learning the initial generation ability, the hidden state space changes greatly, causing the classifier difficult to train, and it will be relatively easier to predict random/all-1/all-0. As the training progresses, the generation loss decreases and the proportion of auxiliary loss increases, at which time the auxiliary tasks can be optimized.

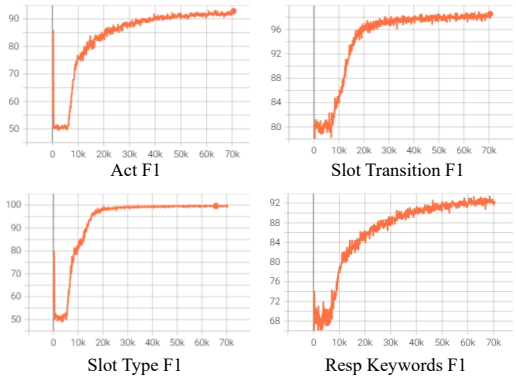


Figure 4: Learning curve for different tasks in training. X-axis represents the number of training steps and Y-axis represents macro F1-score.

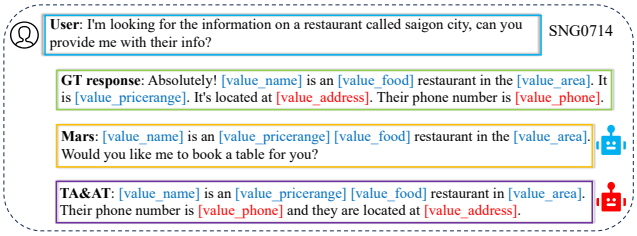


Figure 5: Case Study: Delexicalized responses generated by Mars and TA&AT on MultiWOZ 2.0 test data. 'GT' is short for ground truth.

In addition, the tasks related to belief state converge quickly and can reach F1-score above 96, while the tasks related to policy converge slowly and can only reach F1-score around 92. It can be seen that the latter is more difficult than the former, because it requires more planning ability besides understanding.

6.3 Case Study

As shown in Figure 5, our method can generate more keywords than Mars when the user needs some information, covering all the information contained in the ground truth response and containing no redundant information.

7 Conclusion

In this study, we explore the techniques for optimizing task-oriented dialog via turn-level auxiliary tasks and action-tree based scheduled sampling. To address the insufficient utilization of labels and sequence-level error accumulation issues that existing models struggle with, we primarily introduce turn-level multi-task objectives for the encoder module. Furthermore, we introduce an action-tree based scheduled sampling technique for the decoder module. Our approach has depicted superior performance on the MultiWOZ dataset series compared to methods without continual pre-training and remains competitive even when benchmarked against methods that adopt pre-training.

Acknowledgments

We thank all the anonymous reviewers for their insightful and valuable comments. This work was supported by National Key R&D Program of China (NO. 2018AAA0102502) and Independent Research Project of Medical Engineering Laboratory of Chinese PLA General Hospital (2022SYSZZKY23).

References

Bang, N.; Lee, J.; and Koo, M.-W. 2023. Task-Optimized Adapters for an End-to-End Task-Oriented Dialogue System. *arXiv preprint arXiv:2305.02468*.

Bengio, S.; Vinyals, O.; Jaitly, N.; and Shazeer, N. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28.

Budzianowski, P.; Wen, T.-H.; Tseng, B.-H.; Casanueva, I.; Ultes, S.; Ramadan, O.; and Gašić, M. 2018. Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Cholakov, R.; and Kolev, T. 2022. Efficient Task-Oriented Dialogue Systems with Response Selection as an Auxiliary Task. *arXiv preprint arXiv:2208.07097*.

Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; and Hon, H.-W. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*, 32.

Gu, J.; Lu, Z.; Li, H.; and Li, V. O. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

He, W.; Dai, Y.; Yang, M.; Sun, J.; Huang, F.; Si, L.; and Li, Y. 2022a. Unified dialog model pre-training for task-oriented dialog understanding and generation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 187–200.

He, W.; Dai, Y.; Zheng, Y.; Wu, Y.; Cao, Z.; Liu, D.; Jiang, P.; Yang, M.; Huang, F.; Si, L.; et al. 2022b. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 10749–10757.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Hosseini-Asl, E.; McCann, B.; Wu, C.-S.; Yavuz, S.; and Socher, R. 2020. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems*, 33: 20179–20191.

Jeon, H.; and Lee, G. G. 2021. Domain state tracking for a simplified dialogue system. *arXiv preprint arXiv:2103.06648*.

Kalchbrenner, N.; Grefenstette, E.; and Blunsom, P. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Kim, S.; Yang, S.; Kim, G.; and Lee, S.-W. 2019. Efficient dialogue state tracking by selectively overwriting memory. *arXiv preprint arXiv:1911.03906*.

Kulhánek, J.; Hudeček, V.; Nekvinda, T.; and Dušek, O. 2021. AuGPT: Auxiliary tasks and data augmentation for end-to-end dialogue with pre-trained language models. *arXiv preprint arXiv:2102.05126*.

Lee, Y. 2021. Improving end-to-end task-oriented dialog system with a simple auxiliary task. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 1296–1303.

Lei, W.; Jin, X.; Kan, M.-Y.; Ren, Z.; He, X.; and Yin, D. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1437–1447.

Li, Z.; Zhang, J.; Fei, Z.; Feng, Y.; and Zhou, J. 2021. Conversations are not flat: Modeling the dynamic information flow across dialogue utterances. *arXiv preprint arXiv:2106.02227*.

Lin, Z.; Madotto, A.; Winata, G. I.; and Fung, P. 2020. Mintl: Minimalist transfer learning for task-oriented dialogue systems. *arXiv preprint arXiv:2009.12005*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Peng, B.; Li, C.; Li, J.; Shayandeh, S.; Liden, L.; and Gao, J. 2021. Soloist: Building task bots at scale with transfer learning and machine teaching. *Transactions of the Association for Computational Linguistics*, 9: 807–824.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.

Su, Y.; Shu, L.; Mansimov, E.; Gupta, A.; Cai, D.; Lai, Y.-A.; and Zhang, Y. 2021. Multi-task pre-training for plug-and-play task-oriented dialogue system. *arXiv preprint arXiv:2109.14739*.

Sun, H.; Bao, J.; Wu, Y.; and He, X. 2022. BORT: Back and denoising reconstruction for end-to-end task-oriented dialog. *arXiv preprint arXiv:2205.02471*.

Sun, H.; Bao, J.; Wu, Y.; and He, X. 2023. Mars: Modeling Context & State Representations with Contrastive Learning for End-to-End Task-Oriented Dialog. In *Findings of the Association for Computational Linguistics: ACL 2023*, 11139–11160.

Takanobu, R.; Zhu, Q.; Li, J.; Peng, B.; Gao, J.; and Huang, M. 2020. Is your goal-oriented dialog model performing really well? empirical analysis of system-wise evaluation. *arXiv preprint arXiv:2005.07362*.

Wen, T.-H.; Vandyke, D.; Mrksic, N.; Gasic, M.; Rojas-Barahona, L. M.; Su, P.-H.; Ultes, S.; and Young, S. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 38–45.

Yang, Y.; Li, Y.; and Quan, X. 2021. UBAR: Towards fully end-to-end task-oriented dialog system with GPT-2. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 14230–14238.

Zhang, K.; and Shasha, D. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6): 1245–1262.

Zhang, W.; Feng, Y.; Meng, F.; You, D.; and Liu, Q. 2019. Bridging the gap between training and inference for neural machine translation. *arXiv preprint arXiv:1906.02448*.

Zhang, Y.; Ou, Z.; Wang, H.; and Feng, J. 2020. A probabilistic end-to-end task-oriented dialog model with latent belief states towards semi-supervised learning. *arXiv preprint arXiv:2009.08115*.

Zhang, Y.; Ou, Z.; and Yu, Z. 2020. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 9604–9611.