Modularized Pre-training for End-to-end Task-oriented Dialogue
Libo Qin, Xiao Xu, Lehan Wang, Yue Zhang, Wanxiang Che

Abstract—Pre-training for end-to-end task-oriented dialogue systems (EToDs) is a challenging task due to its unique knowledge base query (accuracy) need and lack of sufficient training data (fluency). In this paper, we try to mitigate the above challenges by introducing a modularized pre-training framework for EToDs, which achieves to effectively improve both accuracy and fluency of EToDs through a pre-training paradigm. The core insight is a modular design by decomposing EToDs into a generation (fluency) module and a knowledge-retriever (accuracy) module, which allows us to optimize each module by pre-training these two sub-modules with different well-designed pre-training tasks, respectively. In addition, such a modularized paradigm enables us to make full use of large amounts of KB-free dialogue corpus for the pre-training generation module, which can alleviate the insufficient training problem. Furthermore, we introduce a new consistency-guided data augmentation (CGDA) strategy to cope with the data scarcity problem to better pre-train the knowledge-retriever module. Finally, we fine-tune the pre-trained generation module and knowledge-retriever module jointly. Experimental results on three datasets show that our model achieve superior performance in terms of both fluency and accuracy. To our knowledge, this is the first work to explore modularized pre-training methods for EToDs.

Index Terms—Task-oriented Dialogue System, Modularized Pre-training, consistency-guided data augmentation.

I. INTRODUCTION

Task-oriented dialogue systems (ToDs) can complete user goals such as hotel bookings and restaurant reservations, which gains increasing attention. Traditional ToDs consists of modularly connected components for natural language understanding (NLU) [1], dialogue state tracking (DST) [2], dialogue policy (DP) [3] and natural language generation (NLG) [4] module. In recent years, end-to-end task-oriented dialogue systems (EToDs) has emerged in the literature, which use a unified sequence-to-sequence model to generate a response given a dialogue history and knowledge base (KB) [5]. For example, given the dialogue history “Send me to the nearest gas station, I want to fuel my car.” in the first turn and the corresponding knowledge base in Figure 1, EToDs can directly produce the system response “The nearest gas station is Valero at 200 Alester Ave, 7 miles away setting directions now”, which does not need any intermediate supervision.

Pre-trained language models, such as GPT-2 [6], have shown empirical success on open-domain dialogue direction [7], [8]. In addition, great progress has also been witnessed in and task-oriented dialogue direction, including dialogue state tracking [9], natural language generation [10]. However, it is relatively under-explored for EToDs due to the following challenges: First, pre-training for current EToDs requires large amounts of KB-grounded dialogues that is extremely hard to collect, which is insufficient for capturing task-oriented domain characteristic, resulting in low fluency; Second, more importantly, task-oriented dialogue systems require a KB retrieval module, which does not seamlessly integrate with the general pre-training stage, leading to low accuracy. As shown in Figure 1, the dialogue system is required to query a corresponding KB to retrieve the entity like “Valero” for the driver’s query about gas station.

Motivated by this, we propose a new pre-training paradigm to solve the aforementioned challenges. To be more specific, we propose a modularized pre-training framework that decomposes the model into a generation (fluency) module and a knowledge-retriever (accuracy) module. With the help of modularized pre-training paradigm, generation module and knowledge retriever module are decoupled, which brings us at least two advantages: (1) it allows each sub-module to benefit from further pre-training to improve both fluency and accuracy, which makes it easier integrate them into a unified EToDs architecture; (2) it enables the model to make full use of a large amount of KB-free dialogues corpus, which is able to alleviate the insufficient data problem.

To improve fluency, as shown in Figure 2(a), we design two pre-training tasks for the generation module to mitigate data distribution shift issue between the pre-training stage and fine-tuning stage: (1) Language Modeling Task (LMT): inspired
by Wu et al. [12], we apply LMT on collected task-oriented dialogue corpus to bridge the domain gap between the general domain and the EToDs domain; (2) Entity Prediction Task (EPT): EPT is used for predicting whether each generated word is a knowledge entity during the decoding stage that aims to enhance the model with knowledge awareness, which is essential for task-oriented dialogue domain. It is worth noting that both LMT and EPT pre-training only need KB-free dialogues, which is much easier to collect.

To improve accuracy, we first freeze the generation module and further pre-train the knowledge-retriever module independently. Specifically, we propose a Knowledge Retrieval Task (KRT) that is used for selecting which knowledge entities in KB will be retrieved by the knowledge-retriever module, which improves the ability to retrieve correct knowledge entities from KB for dialogue generation, as shown in Figure 2(b). The pre-trained knowledge-retriever module can be easily integrated with the generation module due to our modularization framework. Nevertheless, unlike the generation module pre-training, the knowledge-retriever module has to rely on a considerable amount of KB-grounded dialogue for pre-training, which is much difficult to acquire than KB-free dialogues collection. To tackle the data scarcity problem, we further propose a consistency-guided data augmentation strategy, which enlarges pseudo KB-grounded dialogues corpus from existing data by replacing specific knowledge entities without breaking the dialogue context consistency and KB consistency (see §III-B).

As shown in Figure 2(c), after separately pre-training the generation module and knowledge-retriever module, we fine-tune the pre-trained system on three public EToDs datasets including SMD [11], CamRest [13] and an extension of MultiWOZ 2.1 [14]. Experimental results demonstrate the effectiveness of our proposed modularized pre-training framework by obtaining promising performance on both accuracy and fluency in EToDs.

The contribution of this work can be summarized as: (1) We propose a modularized pre-training framework for EToDs by decoupling the generation module and the knowledge-retriever module, allowing the model to optimize the two sub-modules to improve accuracy and fluency, respectively. In addition, such modularized paradigm can enable model to utilize KB-free dialogue; (2) We devise a new consistency-guided data augmentation strategy (CGDA) to alleviate the data insufficiency problem for pre-training knowledge-retriever module while keeping the dialogue context consistency and KB consistency; (3) Results on three public datasets show that our framework achieves superior performance. Extensive analysis verifies that the well-designed pre-training tasks improve both fluency and accuracy in EToDs.

To facilitate the further research, our code are publicly available at https://github.com/LooperXX/MPEToDs.

II. BACKGROUND

In this section, we provide a brief introduction to end-to-end task-oriented dialogue systems (EToDs).

a) Dialogue History: Given a user $u$ and system $s$, we follow Qin et al. [15] to represent the n-turned dialogue utterances as $(u_1, s_1), (u_2, s_2), ..., (u_n, s_n)$. In addition, at the i-th turn of the dialogue, we flatten dialogue context $(u_1, u_2, ..., u_i)$ and denote $X = (x_1, x_2, ..., x_m)$ as the dialogue history where $m$ stands for the number of words in the dialogue history.

b) Knowledge Base: As described in Qin et al. [15], the corresponding knowledge base (KB) is a relational-database-like KB B, which contains $|R|$ rows and $|C|$ columns. The value of entity in the i-th row and the j-th column is represented as $v_{i,j}$.

c) End-to-end Task-oriented Dialogue Systems: Following Eric et al. [11] and Qin et al. [14], the end-to-end task-oriented dialogue generation task can be defined as predicting the most likely response $Y$ given the dialogue history $X = (x_1, x_2, ..., x_m)$ and KB $B$. Formally, this process is defined as:

$$P(Y | X, B) = \prod_{t=1}^{n} p(y_t | y_1, ..., y_{t-1}, X, B),$$

(1)

where $y_t$ denotes an output word; $m$ and $n$ are the length of dialog history and response, respectively.

III. MODULARIZED PRE-TRAINING FRAMEWORK

This section illustrates the overview of Modularized Pre-training Framework. Specifically, it consists of three main stages:

1. first pre-train generation module (§ III-A);
2. then freeze the pre-trained generation module and pre-train knowledge-retriever module (§ III-B);
In our framework, we employ DialoGPT as the backbone since it has incorporated open-domain dialogue features.

A. Generation Module Pre-Training

a) Data collection: Following Wu et al. [12], we collect and combine nine human-human task-oriented dialogue corpora to further pre-train DialoGPT. In total, there are around 102K dialogues with 1.7M utterances. The statistics of data used in the generation module pre-training are shown in Table I.

b) Language Modeling Task (LMT): Though achieving promising performance on open-domain dialogue, DialoGPT leads to a large gap in domain distribution between open-domain dialogue (OOD) and EToDs. Therefore, we explore Language Modeling Task (LMT) to further pre-train DialoGPT, achieving to capture domain feature of EToDs.

The task of LM is to predict a distribution of the next word given the previous words. Formally, given the hidden outputs \( \{ h_1, \ldots, h_{m+n} \} \) of DialoGPT, the prediction probability for next word (\( t \) timestep) is computed as:

\[
y^L_t = \text{Softmax} \left( U h_t \right),
\]

where \( U \in \mathbb{R}^{V \times d} \) is the output embedding matrix; \( d \) denotes the dimension size; \( V \) denotes the vocabulary size.

c) Entity Prediction Task (EPT): It’s hard for DialoGPT to distinguish whether the generated word is a common word or an entity, since it’s just pre-trained on open-domain dialogue corpus. To facilitate DialoGPT awareness of the knowledge that is essential for EToDs, we propose the entity prediction task, which is used for judging whether the generated word is an entity or not. Formally, the task is defined as:

\[
y^E_t = \text{Sigmoid}(u^\top h_t + b),
\]

where \( y^E_t \) denotes the probability that the \( i \)-th generated word is an entity. For each word in the system response, we use 1 to indicate the word is a knowledge entity, and 0 otherwise.

We argue that if model can successfully predict the entity in the system response, the model is capable of capturing knowledge awareness.

B. Knowledge-retriever Module Pre-training

A Knowledge Retrieve Task (KRT) is introduced to pretrain the knowledge-retriever module to improve the KB-retriever ability. Specifically, KRT is used for selecting which knowledge entities in KB will be retrieved by the generation module, which enhances the ability to retrieve correct knowledge entities from KB for dialogue generation.

a) Knowledge Retrieve Task (KRT): Following Wu et al. [21] and Qin et al. [14], we adopt memory-network to store knowledge. Since knowledge entities can either come from the KB or the dialog history, we use an external knowledge memory to encode both the KB \( B \) and the dialogue history \( X \). The entities in external knowledge memory can be denoted as a triple format \((\text{subject}, \text{relation}, \text{object})\), which are represented as \( M = [B; X] = (m_1, \ldots, m_b, m_b+1, \ldots, m_b+T), \) where \( m_b \) stands for triplet in \( M \); \( b \) and \( T \) denotes the number of triplets in KB and dialog history, respectively. A bag-of-word representation is used to represent each entity (triple) in \( M \). For
Task Formulation: KRT aims to predict knowledge retriever filter $G = (g_1, \ldots, g_{b+1})$, which is used for filtering irrelevant knowledge information in the knowledge entity generation process. $G$ can be calculated with a multi-hop memory network.

Entity Detection process includes three steps: (1) Type Filter, (2) Entity-Select, and (3) Entity-Replace. We perform entity value replacement to keep the dialog context consistency.

The intuition behind KRT pre-training is that a better KB consistency can be better used to correctly retrieve external knowledge trainable parameters obtained from the pre-training stage can be better used to correctly retrieve knowledge in the KB for EToDs.

### Algorithm 1: Consistence-guided data augmentation (CGDA) in knowledge-retriever module pre-training

<table>
<thead>
<tr>
<th>Input:</th>
<th>Seed KB-grounded dialogues: $D_S$;</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E$: Entity set;</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{E}_{type}$: Candidate entity type set;</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{E}_{ent}$: Entity dictionary;</td>
<td></td>
</tr>
<tr>
<td>Output:</td>
<td>Augmented KB-grounded Dialogues: $D_K$.</td>
</tr>
<tr>
<td>$D_K = \emptyset$;</td>
<td></td>
</tr>
<tr>
<td>for $(X, B) \in D_S$ do</td>
<td></td>
</tr>
<tr>
<td>if $\exists e \in D$ then</td>
<td></td>
</tr>
<tr>
<td>$E \leftarrow$ Entity-Detection($X, \mathcal{E}_{ent}$);</td>
<td></td>
</tr>
<tr>
<td>$\hat{E} \leftarrow$ Entity-Type-Filter($E, \mathcal{E}_{type}$);</td>
<td></td>
</tr>
<tr>
<td>if $\hat{E} \neq \emptyset$ then</td>
<td></td>
</tr>
<tr>
<td>$(\hat{X}, \hat{B}) \leftarrow (X, B)$</td>
<td>$\triangleright$ Init Augmented Data</td>
</tr>
<tr>
<td>for $e \in \hat{E}$ do</td>
<td></td>
</tr>
<tr>
<td>$t \leftarrow$ Entity-Type($e, \mathcal{E}_{ent}$);</td>
<td></td>
</tr>
<tr>
<td>$\hat{e} \leftarrow$ Entity-Select($\mathcal{E}_{ent}[t], B$);</td>
<td></td>
</tr>
<tr>
<td>$(\hat{X}, \hat{B}) \leftarrow$ Entity-Replace($\hat{X}, \hat{B}, e, \hat{e}$);</td>
<td></td>
</tr>
<tr>
<td>$D_K \leftarrow D_K \cup {\hat{X}, \hat{B}}$;</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
</tbody>
</table>

where $p_i^k$ is logits at the $i$-th position, $c_i^k$ is embeddings in $i$-th memory position using $C^K$, $\omega^k$ denotes the weighted sum over $c_i^{k+1}$ and $q_i^{k+1}$ represents the updated query vector.

At the last $K$ hop, each pointer value can be obtained by:

$$ g_i = \text{Sigmoid}(\langle q_i^K \rangle^T c_i^K), $$

where $g_i$ represents the possibility of the $i$-th object word existing in the system response.

The details of the mentioned function are as follows:

- **Entity Detection**: We first detect all the entities and their entity types in the dialogue history. For example, the gas station and its entity type POI type are detected, which is shown in Figure 3(b).

- **Entity Type Filter**: We replace the selected entities. Here, we should not randomly select what entities to replace, since this may cause dialog context inconsistency. For example, if we replace the selected gas station with a Chinese restaurant, there is a contradiction between dialog context. This is because the replaced entity Chinese restaurant has a conflict with the driver’s utterance *I want to fuel my car*. To alleviate this issue, we maintain an entity type set $\mathcal{E}_{type}$ to ensure dialog context consistency.

- **Entity Value Replacement**: We perform entity value replacement. To keep KB consistency, we ensure that the entities used for replacement do not exist in the current KB. As shown in Equation 3(d), we replace 200 Alester Ave with 481 Amaranta Ave. However, KB consistency is violated when we replace 200 Alester Ave with 271 Springer Street, since it causes two different POIs to have the same Address 271 Springer Street in the augmented KB. This violates the common sense that the address attribute of each POI is unique in the world.

Algorithm 1 shows the pseudocode for the consistence-guided data augmentation process, where lines 4-7 denote the generation process that can use KB-free dialogues, only KB-grounded task-oriented dialogues can be used for pre-training to seed KB-grounded dialogues.

Unlike data augmentation in other areas, we require the augmented dialogues to follow two important general logical rules: (1) **dialog context consistency** and (2) **KB consistency**. Next, we describe the process of data augmentation and how to keep the consistency. As shown in Figure 3, the augmentation process includes three steps: (1) **Entity Detection**, (2) **Entity Type Filter** and (3) **Entity Value Replacement**.

**Entity Detection**. We first detect all the entities and their entity types in the dialogue history. For example, the gas station and its entity type POI type are detected, which is shown in Figure 3(b).

**Entity Type Filter**. We replace the selected entities. Here, we should not randomly select what entities to replace, since this may cause dialog context inconsistency. For example, if we replace the selected gas station with a Chinese restaurant, there is a contradiction between dialog context. This is because the replaced entity Chinese restaurant has a conflict with the driver’s utterance *I want to fuel my car*. To alleviate this issue, we maintain an entity type set $\mathcal{E}_{type}$ to ensure dialog context consistency.

1In our work, entity type set $\mathcal{E}_{type}$ are {party, room, agenda, location, address}, {address, area, id, location, phone, postcode}, {address, area, phone, postcode, ref, stars} for SMD, Camrest676, and Multi-WOZ 2.1, respectively.
The Entity-Replace function replaces the original entity $e$ in the augmented dialog history $\hat{X}$ and augmented knowledge base $\hat{B}$ with the selected entity $\hat{e}$.

Finally, the augmented pseudo dialogues are used for pre-training the knowledge-retriever module.

The data statistics of the augmentation data obtained by our consistency-guided data augmentation method are shown in Table II. Each dialogue obtained through data augmentation has replaced all replaceable entities based on our consistency-guided data augmentation method.

### C. EToDs Task Fine-tuning

Finally, after separately pre-training the generation module and knowledge-retriever module, we show how to fine-tune the pre-trained system on the EToDs tasks. Following Wu et al. [21] and Qin et al. [14], we use a sketch tag to control whether the model generates the common word or knowledge entity. Sketch tag denotes the possible slot types that start with whether the model generates the common word or knowledge KB, which can be calculated by:

$$h_t = \text{DialoGPT}(e_t, h_{t-1}),$$

where $h_t$ is the hidden representation at $t$ timestep.

$$o_t = \mathbf{U} h_t,$$

where $\mathbf{U}$ represents the trainable projection matrix and $p(y_t)$ denotes the probability of token $y_t$.

2) Knowledge-retriever Module: When a sketch tag is generated, $h_t$ is also used as query to retrieve knowledge from KB, which can be calculated by:

$$q^1_t = h_t,$$

where $q^1_t$ is the hidden representations at $t$ timestep.

$$p_i^k = \text{Softmax}((q^k_t)^\top c_i^k y_t).$$

Following Wu et al. [21] and Qin et al. [14], at the last hop, $p_t = (p^1_t, \ldots, p^K_{b+T})$ is treated as the probabilities of knowledge entities at $t$ timestep.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domains</th>
<th>Dialogues</th>
<th>Utterances</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMD</td>
<td>Navigate, Weather, Schedule</td>
<td>18,710</td>
<td>51,720</td>
</tr>
<tr>
<td>CamRest</td>
<td>Restaurant</td>
<td>4,040</td>
<td>16,620</td>
</tr>
<tr>
<td>MultiWOZ2.1</td>
<td>Restaurant, Attraction, Hotel</td>
<td>15,830</td>
<td>72,720</td>
</tr>
</tbody>
</table>

### TABLE II: Data Statistics for Task-Oriented Dialogue Augmentation Data in Knowledge-retriever Module Pre-Training.

D. Training Objective

1) Generation Module Pre-Training: LM and EP tasks are adopted to pre-train our generation module jointly and the loss is:

$$\mathcal{L}_G = \mathcal{L}_{LM} + \mathcal{L}_{EP},$$

where $\mathcal{L}_{LM} = -\sum_{i=1}^{m+n} \hat{y}^{LM}_i \log (y^{LM}_i)$, and $\mathcal{L}_{EP} = -\sum_{i=1}^{1} (\hat{y}^{EP}_i \log (y^{EP}_i) + (1-\hat{y}^{EP}_i) \cdot \log (1-y^{EP}_i))$.

2) Knowledge-retriever Module Pre-training: The loss of the knowledge retriever task is:

$$\mathcal{L}_{GP} = -\sum_{i=1}^{n} (\hat{g}_i \cdot \log g_i + (1-\hat{g}_i) \cdot \log (1-g_i)),$$

where $\hat{g}_i$ denotes gold label and $\hat{g}_i = 1$ if the $i$-th object word in the memory existing in the system response, 0 otherwise.

3) Fine-tuning Training Objective: Specifically, given the system response $Y$, we can get local memory pointer label sequence $\hat{L} = (\hat{l}_1, \ldots, \hat{l}_n)$ as follows:

$$\hat{l}_t = \left\{ \begin{array}{ll} \max(z) & \text{if } \exists z \text{ s.t. } y_t = \text{Object}(m_z) \\ b + T + 1 & \text{otherwise} \end{array} \right. ,$$

where Object($\cdot$) function is used for acquiring the object word from a triplet.

Based on the $\hat{L}$ and $\mathbf{P}_t = (p^1_t, \ldots, p^K_{b+T})$, we can calculate the standard cross-entropy loss $\mathcal{L}_{LP}$ as follows:

$$\mathcal{L}_{LP} = -\sum_{t=1}^{n} \log (\mathbf{P}_t(\hat{l}_t)).$$

The final training objective $\mathcal{L}$ used in the fine-tuning process is the weighted-sum of four losses:

$$\mathcal{L} = \gamma_{lm} \mathcal{L}_{LM} + \gamma_{ep} \mathcal{L}_{EP} + \gamma_{gp} \mathcal{L}_{GP} + \gamma_{lp} \mathcal{L}_{LP},$$

where $\gamma_{lm}$, $\gamma_{ep}$, $\gamma_{gp}$ and $\gamma_{lp}$ are hyperparameters.

### IV. EXPERIMENTS

We evaluate our framework by fine-tuning the pre-trained generation and knowledge-retriever module for the end-to-end task-oriented dialogue system.
TABLE IV: Main results. The bolded number indicates the best performance and “*” indicates the original paper does not report results in the same dataset. Results with * indicate that the improvement of our framework is statistically significant under t-test.

<table>
<thead>
<tr>
<th>Model</th>
<th>SMD BLEU</th>
<th>SMD F1</th>
<th>Weather BLEU</th>
<th>Weather F1</th>
<th>Calendar BLEU</th>
<th>Calendar F1</th>
<th>CamRest676 BLEU</th>
<th>CamRest676 F1</th>
<th>Multi-WOZ 2.1 BLEU</th>
<th>Multi-WOZ 2.1 F1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non pre-trained Models</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mem2Seq† [22]</td>
<td>12.6</td>
<td>33.4</td>
<td>20.0</td>
<td>32.8</td>
<td>49.3</td>
<td></td>
<td>16.6</td>
<td>42.4</td>
<td>5.8</td>
<td>14.4</td>
</tr>
<tr>
<td>GLMP† [21]</td>
<td>13.9</td>
<td>60.7</td>
<td>54.6</td>
<td>56.5</td>
<td>72.5</td>
<td></td>
<td>17.4</td>
<td>54.7</td>
<td>6.9</td>
<td>32.4</td>
</tr>
<tr>
<td>TTOS [23]</td>
<td>17.4</td>
<td>55.4</td>
<td>45.9</td>
<td>64.1</td>
<td>63.5</td>
<td></td>
<td>20.5</td>
<td>61.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DDMN† [24]</td>
<td>15.8</td>
<td>60.7</td>
<td>53.2</td>
<td>64.7</td>
<td>69.3</td>
<td></td>
<td>18.7</td>
<td>59.1</td>
<td>11.5</td>
<td>34.2</td>
</tr>
<tr>
<td>DF-Net† [14]</td>
<td>14.4</td>
<td>62.7</td>
<td>57.9</td>
<td>57.6</td>
<td>73.1</td>
<td></td>
<td>18.8</td>
<td>59.8</td>
<td>9.4</td>
<td>35.1</td>
</tr>
<tr>
<td>Pre-trained Models</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DialoGPT† [7]</td>
<td>14.6</td>
<td>47.0</td>
<td>32.2</td>
<td>55.6</td>
<td>52.8</td>
<td></td>
<td>12.3</td>
<td>44.4</td>
<td>7.0</td>
<td>15.9</td>
</tr>
<tr>
<td>DialoGPT+KB† [25]</td>
<td>16.5</td>
<td>59.2</td>
<td>51.2</td>
<td>53.1</td>
<td>73.3</td>
<td></td>
<td>12.7</td>
<td>51.6</td>
<td>8.4</td>
<td>29.0</td>
</tr>
<tr>
<td>- Out framework</td>
<td>18.8∗</td>
<td>63.8∗</td>
<td>59.1∗</td>
<td>58.4</td>
<td>75.0∗</td>
<td></td>
<td>22.0∗</td>
<td>63.9∗</td>
<td>13.6</td>
<td>36.3∗</td>
</tr>
</tbody>
</table>

VI. A. Experimental Settings

We conduct experiments on three datasets, including SMD [11], CamRest [13] and an extension of Multi-WOZ 2.1 [14]. We follow same partition as Qin et al. [14].

The batch size we use in our framework is 16 and the dropout ratio is 0.1. We use AdamW to optimize the parameters in our model and adopt the suggested hyper-parameters for optimization. We adopt DialoGPT-Medium [7] architecture. All hyperparameters are selected according to the performance of the validation set. All experiments are conducted at Tesla A100 and V100. The detailed hyperparameters used for all datasets are shown in Table III.

B. Baselines

We compare our model with the following strong baselines: (1) Mem2Seq [22]: the model adopts a memory network to encode knowledge entities; (2) GLMP [21]: the framework adopts the global-to-local pointer mechanism to query the KB; (3) TTOS [23]: the model proposes a teacher-student framework to improve the generation and KB query ability; (4) DDMN [24]: the model uses a dual memory network to select better knowledge; (5) DF-Net [14]: the model considers domain features to promote the multi-domain EToDs, which obtains the best results; (6) DialoGPT [7]: the model directly yields response given the dialog history with pre-trained model DialoGPT; (7) DialoGPT+KB: the model first linearizes a table into a sequence [25]. Then it concatenates the linearized KB and the dialog history as input and directly models the task-oriented dialog task as the language modeling task with DialoGPT or GPT2;

V. MAIN RESULTS

A. Automatic Evaluation

Following Wu et al. [21] and Qin et al. [14], we adopt the Micro-Entity F1 and BLEU [27] to evaluate the knowledge querying and fluent response generation ability, respectively.

The results are listed in Table IV. We have the following observations:

1. On BLEU, the DialoGPT-based pre-trained models outperform some strong models without pretraining (i.e., Mem2Seq, GLMP, DDMN and DF-Net) revealing that pre-training from large amounts of dialogues can improve fluency performance, which is consistent with the observation in Peng et al. [28];

2. On Micro-Entity F1, some non pre-trained models (e.g., DF-Net) outperform DialoGPT-based models (e.g., 62.7% vs. 59.2%). This indicates that simply fine-tuning the DialoGPT-based pre-trained models does not effectively integrate the KB query ability;

3. Our framework not only outperforms the non pre-trained models but also the DialoGPT-based pre-trained models in both metrics. Compared with the best non pre-trained model DF-Net, our framework obtains 4.4 point and 1.1% improvements on BLEU and F1 scores, respectively. Our model also outperforms the best pre-trained model DialoGPT + KB by 2.3 point and 4.6% on BLEU and F1, respectively. The same trend is witnessed on other two datasets, which verifies the effectiveness of the modularized pre-training framework that can integrate fluent generation capability and accurate KB query capability.

VI. ANALYSIS

We conduct in-depth analysis to better understand our framework. Specifically, we first explore the effect of generation module pre-training mechanism. Next, we study the effect of the knowledge-retriever module pre-training mechanism. Then, we further explore the impact of modularized pre-training strategy and training data. We also investigate the
A. Generation Module Pre-training Improves Fluency

To verify the effectiveness of the generation module pre-training, instead of further pre-training the DialoGPT model, we directly load the initial DialoGPT weights and then conduct the knowledge-retriever module pre-training. Finally, the pre-trained model is fine-tuned on the downstream task dataset.

The results are illustrated in Table V (w/o GP row). It can be seen that without generation module pre-training, the performance decreases especially on BLEU (-0.7 point on SMD, -1.7 point on Camrest676 and -0.6 point on MultiWOZ2.1), indicating that further pre-training on the task-oriented dialogue can encourage our model to learn the characteristic of EToDs domain to improve fluency.

B. Knowledge-retriever Module Pre-training Boosts Accuracy

To analyze the knowledge-retriever module pre-training stage, we remove the knowledge-retriever module pre-training stage and utilize the model pre-trained with the generation module to fine-tune EToDs.

Table V (w/o KRP row) presents the results. We observe performance drops, especially on F1 scores, which demonstrates the knowledge-retriever module pre-training can learn better representation to enhance the ability to query KB to improve accuracy.

C. Investigation of Modularized Pre-training

To analyze the effectiveness of the proposed modularized pre-training mechanism, we directly fine-tune our model on EToDs without the modularization pre-training strategy.

Table V shows the results. We observe without each pre-training module, the results drop a lot in terms of BLUE or F1 scores, which demonstrates the proposed modularized pre-training boosts both fluency and accuracy.

D. Impact of Training data

In addition, a natural question that arises is whether the collected pre-training data rather than the proposed modularized pre-training framework contributes to the final performance. To answer the question, we first pre-train the DialoGPT+KB model on the same KB-free task-oriented datasets. Then we fine-tune the model on the pseudo KB-grounded dialogues and SMD dataset.

As shown in Table VI, when these two models use the same data, our framework significantly outperforms the DialoGPT+KB model by 4.2 point and 4.6% on BLEU and F1. This further demonstrates that performance improvement comes from the proposed modularized pre-training rather than the pre-training data.

E. Impact of CGDA

To investigate the impact of CGDA, we conduct experiments with different amounts of pseudo dialogues (vary from 1 to 10 times) generated by CGDA.

From Figure 4, we can observe that more pseudo dialogues, the higher the F1 score, which indicates CGDA is essential for generating sufficient augmented training data to pre-train the knowledge-retriever module.

F. Qualitative Analysis

We calculate Entity F1 for each entity type to provide a qualitative analysis to help intuitively understand the effectiveness of the knowledge-retriever module pre-training.

![Figure 5: Qualitative Analysis.](image-url)
As shown in Figure 5, we select five entity types with the most significant performance improvement among all the entity types. In particular, the entity types “room”, “address” and “location” are in the candidate entity type set $E_{type}$, which further verifies that our consistency-guided data augmentation can effectively improve the accuracy performance of our model.

G. Exploration on Low-resource Settings

We explore whether our framework is effective in the low-resource settings, by randomly selecting dialogues in the training set varying from [50, 100, 500, 1000] to simulate the low-resource setting.

The results are shown in Figure 6. We find that our framework consistently outperforms DF-Net and DialoGPT+KB on all sizes, which verifies the robustness of our framework. Besides, our framework gives the largest improvement on extreme low-resource settings (50 dialogues), which makes it more scalable in real scenarios.

H. Human Evaluation

Following Qin et al. [14], we perform human evaluation by hiring human experts to judge the quality of the responses according to the correctness, fluency, and human-likeness on a scale from 1 to 5. We randomly selected 100 dialogue histories on the SMD test data and evaluated the generated responses on our model, DF-Net and DialoGPT + KB. The experts judge the response generated by the anonymous system based on the given dialogue history, knowledge base and golden response.

Results are illustrated in Table VII. We observe our framework beats other baselines on all metrics, which is consistent with automatic evaluation.

VII. RELATED WORK

Existing EToDs work can be classified into two main categories: (1) EToDs without any intermediate supervision and (2) EToDs without intermediate supervision. This section will describe the related work in detail.

Annotators evaluate the correctness of the predicted response based on the ground-truth response and entity labels.

A. End-to-end Task-oriented Dialog without any Intermediate Supervision

The first strand of end-to-end task-oriented dialogue systems (EToDs) directly integrate a unified sequence-to-sequence model for generating system response given the dialogue history and the corresponding knowledge base. This line of work can reduce the efforts of annotating manually designed pipeline modules and easily be adapted to a new domain, which has attracted more and more attention. Our work follows this line of literature. Recently, some work use attention-based models [11], [30] to query the knowledge entity. Eric et al. [11] performs attention over KB to generate entities. Lei et al. [30] considers track dialogue beliefs in the end-to-end task-oriented dialog. Another series of work [21]–[24], [31]–[34] consider the memory network [35] to encode knowledge base. Madotto et al. [22] first combines end-to-end memory network [35] to consider KB query. Gangi et al. [32] proposes a multi-level memory architecture for end-to-end task oriented dialogue system. Wu et al. [21] proposes a global-to-locally pointer mechanism to query KB. Qin et al. [14] further incorporate domain-aware features for EToDs, achieving promising performance. Different from the above work, we explore the pre-training paradigm for EToDs and focus on how to improve both fluency and accuracy with a modularized pre-training framework while their work mainly considers improving performance in fine-tuning stage. To our knowledge, we are the first to explore modularized pre-training methods for EToDs.

B. End-to-end Task-oriented Dialog with Intermediate Supervision

This second strand of end-to-end task-oriented dialogue systems investigates to model of all pipeline sub-tasks dialogue state tracking, action, and response generation as a single sequence in a unified way [36]–[38]. While appealing, this line of literature requires intermediate supervision for training the whole system. Specifically, Peng et al. [28] introduce a transformer-based auto-regressive model to parameterize classical modular task-oriented dialog systems, which requires belief state annotation for supervision. Yang et al. [37] propose a GPT-based end-to-end task-oriented dialogue to generate belief states, the system acts and responses simultaneously. Su et al. [39] propose a unified model that is capable of supporting both task-oriented dialogue understanding and response generation. To this end, they introduce a novel multi-task pre-training strategy to enable the model to learn from different heterogeneous dialog corpora (e.g., NLU and DST). He et al. [40] take the first step to explicitly learn dialog policy by introducing a dialogue act prediction task, which achieves promising performance on the end-to-end dialogue tasks. The above work requires intermediate supervision (e.g., belief states or dialogue act annotation), which is not within the scope of our discussion. Compared with their work, we focus on pre-training for EToDs that directly operate on dialog history and interact with KB without any intermediate supervision.
In this paper, we proposed a modularized pre-training framework for EToDs, with the generation module and knowledge-retriever module. With modularization, our framework can devise various pre-training tasks to enhance the two modules, achieving high fluency and accuracy. In addition, we proposed a simple yet effective consistency-guided data augmentation strategy to generate sufficient KB-grounded dialogues without breaking the dialogue context consistency and KB consistency. Experimental results on three datasets show that our framework achieves superior performance on both fluency and accuracy metrics. To the best of our knowledge, we are the first to explore the modularized pre-training approach for EToDs.

ACKNOWLEDGEMENTS

This work was supported by the National Key R&D Program of China via grant 2020AAA0106501 and the National Natural Science Foundation of China (NSFC) via grant 61976072 and 62176078.

REFERENCES


